

CEK TURNITIN

SKRIPSI MUHAMMAD AIDIL SI

 SATSET DIGITAL 1

Document Details

Submission ID

trn:oid:::3618:150983522

Submission Date

Sep 8, 2026, 3:25 PM GMT+7

Download Date

Sep 8, 2026, 3:32 PM GMT+7

File Name

SKRIPSI MUHAMMAD AIDIL SI.pdf

File Size

6.3 MB

195 Pages

42,271 Words

260,492 Characters

39% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Top Sources

- 6%  Internet sources
- 33%  Publications
- 11%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

6%	Internet sources
33%	Publications
11%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	Muhammad Aidil, Muhammad Jibril, Fitri Yunita. "Automation Testing pada Webs...	29%
2	Internet	repository.unisi.ac.id	1%
3	Student papers	Forum Komunikasi Perpustakaan Perguruan Tinggi Kristen Indonesia (FKPPTKI) o...	<1%
4	Student papers	Universitas Putera Batam on 2026-07-24	<1%
5	Internet	ejournal.itn.ac.id	<1%
6	Internet	repository.nurulfikri.ac.id	<1%
7	Student papers	Universitas Wijaya Kusuma Surabaya on 2026-07-21	<1%
8	Internet	lab.tik.pnj.ac.id	<1%
9	Student papers	UPN Veteran Yogyakarta on 2026-05-05	<1%
10	Student papers	Universitas Wijaya Kusuma Surabaya on 2026-07-21	<1%
11	Internet	digilib.unila.ac.id	<1%

12	Internet	transit-ftik.usm.ac.id	<1%
13	Student papers	Universitas Brawijaya on 2016-11-07	<1%
14	Student papers	Universitas Wijaya Kusuma Surabaya on 2026-07-21	<1%
15	Internet	lib.uin-malang.ac.id	<1%
16	Internet	repository.upi.edu	<1%
17	Internet	www.scribd.com	<1%
18	Student papers	Institut Teknologi Sumatera on 2026-08-18	<1%
19	Student papers	Universitas Dian Nuswantoro on 2016-11-03	<1%
20	Student papers	Sriwijaya University on 2023-05-12	<1%
21	Student papers	Universitas Pendidikan Ganesha on 2026-07-25	<1%
22	Student papers	Sriwijaya University on 2020-03-12	<1%
23	Internet	repository.polibatam.ac.id	<1%
24	Internet	journal.stiestekom.ac.id	<1%
25	Internet	ejurnal.lkpkaryaprima.id	<1%

26	Student papers	STKIP Sumatera Barat on 2026-08-24	<1%
27	Internet	repository.umsu.ac.id	<1%
28	Internet	kc.umn.ac.id	<1%
29	Student papers	Universitas Kristen Duta Wacana on 2026-05-08	<1%
30	Internet	manggalajournal.org	<1%
31	Internet	repository.polinela.ac.id	<1%
32	Internet	repository.upy.ac.id	<1%
33	Student papers	Universiti Teknologi MARA on 2014-06-28	<1%
34	Internet	repository.bunghatta.ac.id	<1%
35	Student papers	STKIP Sumatera Barat on 2026-08-27	<1%
36	Student papers	UIN Maulana Malik Ibrahim Malang on 2026-06-25	<1%
37	Internet	repository.uinsu.ac.id	<1%
38	Publication	Ahmad Nurfaqih, Eko Budihartono, Syefudin Syefudin. "RANCANG BANGUN SISTE...	<1%
39	Student papers	National School of Business Management NSBM, Sri Lanka on 2026-09-05	<1%

40	Student papers	Universitas Muria Kudus on 2019-09-09	<1%
41	Student papers	University of Wolverhampton on 2026-05-26	<1%
42	Student papers	Universitas Singaperbangsa Karawang on 2025-05-27	<1%
43	Student papers	Erciyes Āniversitesi on 2026-07-09	<1%
44	Internet	ejournal.gunadarma.ac.id	<1%
45	Internet	repository.its.ac.id	<1%
46	Internet	repository.staima-alhikam.ac.id	<1%
47	Internet	repository.ub.ac.id	<1%
48	Internet	djournals.com	<1%
49	Internet	ejurnal.esaunggul.ac.id	<1%
50	Internet	www.coursehero.com	<1%
51	Student papers	STKIP Sumatera Barat on 2026-08-27	<1%
52	Student papers	UPN Veteran Yogyakarta on 2026-06-03	<1%
53	Student papers	Universitas Musamus Merauke on 2026-01-12	<1%

54	Internet	digilib.uin-suka.ac.id	<1%
55	Internet	doku.pub	<1%
56	Internet	dspace.uui.ac.id	<1%
57	Internet	journal.uim.ac.id	<1%
58	Internet	repository.upnjatim.ac.id	<1%
59	Student papers	Fakultas Ekonomi Universitas Indonesia on 2015-03-25	<1%
60	Student papers	UPN Veteran Yogyakarta on 2026-04-22	<1%
61	Internet	ojs.stmik-banjarbaru.ac.id	<1%
62	Student papers	Saint Mary's University on 2026-01-09	<1%
63	Student papers	Universitas Putera Batam on 2026-07-28	<1%
64	Student papers	Universitas Putera Batam on 2026-07-28	<1%
65	Internet	repository.mercubuana.ac.id	<1%
66	Internet	sitasi.upnjatim.ac.id	<1%
67	Publication	Nadindra Maulana Aziz. "PENERAPAN TEKNIK BOUNDARY VALUE ANALYSIS DAN E...	<1%

68	Student papers	Politeknik Manufaktur Negeri Bangka Belitung on 2026-07-28	<1%
69	Publication	Hasan Abdullah, Elyza Gustri Wahyuni. "PENGUJIAN APLIKASI MOBILE VERIFIKASI ...	<1%
70	Student papers	Sekolah Teknik Elektro & Informatika on 2025-07-23	<1%
71	Publication	Winaryo, Eko. "Model Penguatan Brand Loyalty Melalui Brand Awareness, Perceiv...	<1%
72	Internet	journal.ipm2kpe.or.id	<1%
73	Internet	repository.bakrie.ac.id	<1%
74	Internet	repository.unj.ac.id	<1%
75	Internet	repository.usd.ac.id	<1%
76	Student papers	East Union High School on 2026-06-29	<1%
77	Student papers	Universitas Muhammadiyah Purwokerto on 2024-12-30	<1%
78	Student papers	Universitas Pendidikan Ganesha on 2025-07-16	<1%
79	Student papers	STKIP Sumatera Barat on 2026-08-24	<1%
80	Internet	csridjournal.potensi-utama.org	<1%
81	Student papers	poltekssn on 2025-07-16	<1%

82	Student papers	poltekssn on 2025-07-16	<1%
83	Publication	Beta Nurul Awwalin, Indyah Hartami Santi, Sabitul Kirom. "SISTEM INFORMASI M...	<1%
84	Student papers	Fakultas Ekonomi Universitas Indonesia on 2015-06-16	<1%
85	Student papers	Politeknik Negeri Jember on 2026-07-22	<1%
86	Student papers	Universitas Muria Kudus on 2022-09-12	<1%
87	Internet	ejournal.unisi.ac.id	<1%
88	Internet	ojs.udb.ac.id	<1%
89	Internet	sinta.unud.ac.id	<1%
90	Internet	www.ttalk.im	<1%
91	Publication	Amelia Ananda Setiawan, Rangga Gelar Guntara, Btari Mariska Purwaamijaya. "E...	<1%
92	Publication	Gita Nur Aprilia, M Noviansyah Dasaprawira. "Perancangan UI/UX Aplikasi E-Rapo...	<1%
93	Student papers	UPN Veteran Yogyakarta on 2026-07-06	<1%
94	Student papers	Universitas Khairun on 2026-06-22	<1%
95	Student papers	Universitas Pamulang on 2021-06-18	<1%

96	Internet	adoc.pub	<1%
97	Internet	d-nb.info	<1%
98	Internet	digilib.ulm.ac.id	<1%
99	Student papers	STKIP Sumatera Barat on 2026-08-18	<1%
100	Student papers	Universitas Brawijaya on 2017-06-20	<1%
101	Student papers	Universitas Muhammadiyah Purwokerto on 2025-06-16	<1%
102	Student papers	Universitas Putera Batam on 2026-07-28	<1%
103	Internet	repository.unpkediri.ac.id	<1%
104	Internet	repository.upnvj.ac.id	<1%
105	Internet	repository.upstegal.ac.id	<1%
106	Internet	www.codepolitan.com	<1%
107	Publication	Kevin Guan, Marcelo d'Amorim, Owolabi Legunsen. "Faster Explicit-Trace Monitor...	<1%
108	Student papers	UIN Raden Intan Lampung on 2026-07-06	<1%
109	Student papers	UIN Raden Intan Lampung on 2026-07-22	<1%

110	Student papers	UPN Veteran Yogyakarta on 2026-03-02	<1%
111	Student papers	Universitas Putera Batam on 2026-02-02	<1%
112	Internet	apidog.com	<1%
113	Internet	docplayer.info	<1%
114	Internet	ejournal.ihtn.ac.id	<1%
115	Internet	etheses.uin-malang.ac.id	<1%
116	Internet	jurnal.unimed.ac.id	<1%
117	Internet	repository.unissula.ac.id	<1%
118	Internet	santika.upnjatim.ac.id	<1%
119	Publication	Alessandro Alessandro, Azzani Nurfadia Rizky, Wasis Haryono. "Implementasi Sist...	<1%
120	Student papers	Institut Teknologi Sumatera on 2026-06-17	<1%
121	Publication	Muhamat Ilham Ardiansa, Himawan Pramaditya. "RANCANG BANGUN LEARNING ...	<1%
122	Publication	Muhammad Nawwal Izzur Rifqi, Ratih Nindyasari, Alif catur Murti. "Rancang Ban...	<1%
123	Student papers	Sekolah Teknik Elektro & Informatika on 2025-08-20	<1%

124	Student papers	UIN Raden Intan Lampung on 2026-04-02	<1%
125	Student papers	Universitas Ahmad Dahlan on 2025-11-25	<1%
126	Student papers	Universitas Negeri Jakarta on 2023-01-04	<1%
127	Student papers	Universitas Pelita Harapan	<1%
128	Student papers	Universitas Putera Batam on 2026-02-02	<1%
129	Student papers	Universitas Tarumanagara on 2025-12-19	<1%
130	Internet	docobook.com	<1%
131	Internet	e-journal.uajy.ac.id	<1%
132	Internet	ejournal.poltekdedc.ac.id	<1%
133	Internet	ejurnal.seminar-id.com	<1%
134	Internet	eprints.perbanas.ac.id	<1%
135	Internet	journal.unpas.ac.id	<1%
136	Internet	jurnal.stmik-yadika.ac.id	<1%
137	Internet	ojs.ninetyjournal.com	<1%

138	Internet	repository.teknokrat.ac.id	<1%
139	Internet	www.researchgate.net	<1%
140	Student papers	Fakultas Ekonomi Universitas Indonesia on 2015-07-02	<1%
141	Student papers	Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur on 2022-08-08	<1%
142	Student papers	Institut Teknologi Sumatera on 2026-09-07	<1%
143	Student papers	National Institute of Business Management Sri Lanka on 2026-05-09	<1%
144	Publication	Nurhaliza Liza. "PERANCANGAN WEBSITE E-COMMERCE UNTUK PEMASARAN PRO...	<1%
145	Student papers	Politeknik Negeri Bandung on 2017-07-25	<1%
146	Student papers	Politeknik Negeri Bandung on 2019-07-16	<1%
147	Student papers	STKIP Sumatera Barat on 2026-08-22	<1%
148	Student papers	STT PLN on 2021-07-21	<1%
149	Student papers	Telkom University on 2025-06-30	<1%
150	Student papers	UIN Raden Intan Lampung on 2026-05-13	<1%
151	Student papers	UNIVERSITAS BUDI LUHUR on 2025-07-23	<1%

152	Student papers	UPN Veteran Yogyakarta on 2025-12-01	<1%
153	Student papers	UPN Veteran Yogyakarta on 2026-03-02	<1%
154	Student papers	Universitas Brawijaya on 2017-02-03	<1%
155	Student papers	Universitas Brawijaya on 2018-05-14	<1%
156	Student papers	Universitas Brawijaya on 2023-07-14	<1%
157	Student papers	Universitas Budi Luhur on 2026-07-24	<1%
158	Student papers	Universitas Dian Nuswantoro on 2015-08-14	<1%
159	Student papers	Universitas Islam Riau on 2025-11-24	<1%
160	Student papers	Universitas Muhammadiyah Purwokerto on 2021-11-09	<1%
161	Student papers	Universitas Pamulang on 2025-07-11	<1%
162	Student papers	Universitas Pamulang on 2026-01-17	<1%
163	Student papers	Universitas Pancasila on 2026-02-23	<1%
164	Student papers	Universitas Putera Batam on 2026-01-31	<1%
165	Student papers	Universitas Putera Batam on 2026-07-28	<1%

166	Student papers	Universitas Respati Indonesia on 2025-08-30	<1%
167	Student papers	Universitas Sumatera Utara on 2026-05-18	<1%
168	Student papers	University of Wollongong on 2024-02-02	<1%
169	Internet	ejournal.itmnganjuk.ac.id	<1%
170	Internet	ejournal.sisfokomtek.org	<1%
171	Internet	ejournal.uniramalang.ac.id	<1%
172	Internet	iltek.ft-uim.ac.id	<1%
173	Internet	jitt.polman-babel.ac.id	<1%
174	Internet	ocs.machung.ac.id	<1%
175	Internet	ojs.sains.ac.id	<1%
176	Internet	papers.gunadarma.ac.id	<1%
177	Internet	repository.uinsaizu.ac.id	<1%
178	Internet	repository.unej.ac.id	<1%
179	Internet	repository.unhas.ac.id	<1%

180	Internet	repository.unjaya.ac.id	<1%
181	Publication	Arya Dillah, Gigih Forda Nama, Deny Budiyanto, Meizano Ardhi Muhammad. "RA...	<1%
182	Student papers	Fakultas Teknologi Industri on 2026-09-03	<1%
183	Student papers	Sriwijaya University on 2020-08-25	<1%
184	Student papers	Sriwijaya University on 2021-08-25	<1%
185	Student papers	Sriwijaya University on 2025-12-17	<1%
186	Student papers	Universitas Brawijaya on 2017-06-20	<1%
187	Student papers	Universitas Brawijaya on 2019-03-29	<1%
188	Student papers	Universitas Brawijaya on 2020-07-08	<1%
189	Student papers	Universitas Islam Indonesia on 2018-08-07	<1%
190	Student papers	Universitas Putera Batam on 2018-12-01	<1%
191	Internet	jacis.pubmedia.id	<1%
192	Student papers	Boston University on 2026-05-13	<1%
193	Student papers	Institut Agama Islam Al-Zaytun Indonesia on 2025-08-30	<1%

194	Student papers	Institut Teknologi Sumatera on 2026-08-19	<1%
195	Student papers	SDM Universitas Gadjah Mada on 2021-10-08	<1%
196	Student papers	STIKOM Surabaya on 2013-11-20	<1%
197	Student papers	Telkom University on 2026-05-04	<1%
198	Student papers	UPN Veteran Yogyakarta on 2026-06-05	<1%
199	Student papers	Universitas Islam Riau on 2026-08-28	<1%
200	Student papers	Universitas Muhammadiyah Purwokerto on 2025-08-02	<1%
201	Student papers	Universitas Musamus Merauke on 2025-06-13	<1%
202	Student papers	Universitas Negeri Jakarta on 2026-08-01	<1%
203	Student papers	Universitas Negeri Jakarta on 2026-08-01	<1%
204	Student papers	Universitas Sangga Buana YPKP on 2026-07-27	<1%

1

**PENGUJIAN OTOMATIS PADA WEBSITE KONSELING
MAHASISWA (SIKOSA) MENGGUNAKAN JEST**

2

TUGAS AKHIR



Disusun oleh :

MUHAMMAD AIDIL

403221010060

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS ISLAM INDRAGIRI
TEMBILAHAN**

2026

**PENGUJIAN OTOMATIS PADA WEBSITE KONSELING
MAHASISWA (SIKOSA) MENGGUNAKAN JEST**

***AUTOMATED TESTING OF THE STUDENT COUNSELING
WEBSITE (SIKOSA) USING JEST***

Diajukan untuk memenuhi salah satu syarat
memperoleh derajat Sarjana S1



Disusun oleh :

MUHAMMAD AIDIL

403221010060

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS ISLAM INDRAGIRI
TEMBILAHAN
2026**

PERSETUJUAN

PENGUJIAN OTOMATIS PADA WEBSITE KONSELING MAHASISWA (SIKOSA) MENGGUNAKAN JEST

Yang dipersiapkan dan disusun oleh

MUHAMMAD AIDIL

403221010060

Telah disetujui oleh Dosen Pembimbing
pada tanggal 03 Desember 2025

Pembimbing Utama

Pembimbing Pendamping

Muhammad Jibril, S.Kom., M.MSi
NIDN. 1028029201

Fitri Yunita, S.SI., M.Kom
NIDN. 1012089001

Dekan Fakultas Teknik dan Ilmu Komputer

Dr. Bayu Rianto, S.SI., M.Kom
NIPY. 1691 09 321

PENGESAHAN

PENGUJIAN OTOMATIS PADA WEBSITE KONSELING MAHASISWA (SIKOSA) MENGGUNAKAN JEST

Dipersiapkan dan disusun oleh

MUHAMMAD AIDIL
403221010060

Telah Diuji dan Dipertahankan di depan Dewan Penguji
pada Tanggal 14 Juli 2026

Susunan Dewan Penguji

Pembimbing Utama

Dewan Penguji I

Muhammad Jibril, S.Kom., M.MSi
NIDN. 1028029201

Prof. Dr. H. Abdullah, S.Si., M.Kom
NIDN. 1008037001

Pembimbing Pendamping

Dewan Penguji II

Fitri Yunita, S.SI., M.Kom
NIDN. 1012089001

Dr. Bayu Rianto, S.SI., M.Kom
NIDN. 1014039101

Dewan Penguji III

Abdul Muni, S.Kom., M.kom
NIDN. 1026108601

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer

Dekan Fakultas Teknik dan Ilmu Komputer

Ka. Prodi Sistem Informasi

Dr. Bayu Rianto, S.SI., M.Kom
NIPY. 1691 09 321

Fitri Yunita, S.SI., M.Kom
NIPY. 1590 10 293

2

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama Mahasiswa : **MUHAMMAD AIDIL**

NIM : 403221010060

Program Studi : Sistem Informasi

Menyatakan bahwa skripsi sbb:

Judul bahasa Indonesia :

PENGUJIAN OTOMATIS PADA WEBSITE KONSELIN MAHASISWA (SIKOSA) MENGGUNAKAN JEST

Judul bahasa Inggris :

AUTOMATED TESTING OF THE STUDENT COUNSELING WEBSITE (SIKOSA) USING JEST

Pembimbing Utama : **Muhammad Jibril, S.Kom., M.MSi**

Pembimbing Pendamping : Fitri Yunita, S.SI., M.Kom

1. Laporan Tugas Akhir ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas Islam Indragiri maupun di perguruan tinggi lainnya;
2. Laporan Tugas Akhir ini merupakan gagasan, rumusan dan pemikiran SAYA sendiri, tanpa bantuan pihak lain kecuali arahan Tim Dosen Pembimbing;
3. Dalam laporan Tugas Akhir ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada laporan ini;
4. Perangkat lunak yang digunakan dalam Tugas Akhir ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas Islam Indragiri;
5. Pernyataan ini SAYA buat dengan sesungguhnya apabila di kemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku diperguruan tinggi.

Tembilahan, 14 Juli 2026
Yang Menyatakan,

Materai
Rp.10.000

MUHAMMAD AIDIL

28

KATA PENGANTAR

Segala puji bagi Allah SWT, atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan Skripsi yang berjudul “*Pengujian Otomatis pada Website Konseling Mahasiswa (SiKosa) Menggunakan Jest*” ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Teknik dan Ilmu Komputer Universitas Islam Indragiri.

Penyusunan skripsi ini merupakan upaya penulis dalam merancang solusi pengujian perangkat lunak yang lebih efektif dan efisien untuk memastikan sebuah sistem dapat berjalan sesuai kebutuhan.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu, dengan hormat penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Dr. H. Najamuddin, Lc., M.A, selaku Rektor Universitas Islam Indragiri, yang telah memfasilitasi, dukungan, serta kesempatan berharga kepada penulis, sehingga proses penyusunan skripsi ini dapat berjalan dengan lancar hingga selesai.
2. Bapak Dr. Bayu Rianto, S.SI., M.Kom., selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Islam Indragiri sekaligus Dosen Pembimbing Akademik, yang telah memberikan dukungan, arahan, serta bimbingan kepada penulis selama proses perkuliahan.
3. Ibu Fitri Yunita, S.SI., M.Kom, selaku Ketua Program Studi Sistem Informasi sekaligus Pembimbing pendamping, yang telah memberikan bimbingan, perhatian, dukungan, serta masukan yang sangat berharga

74 kepada penulis hingga penyusunan skripsi ini dapat terselesaikan dengan baik.

4. Bapak Muhammad Jibril, S.Kom., M.MSi., selaku Dosen Pembimbing utama, yang dengan penuh dedikasi telah memberikan bimbingan, pengetahuan, serta arahan yang sangat berharga. Beliau membantu penulis memahami berbagai aspek akademik dan metodologis yang diperlukan, sehingga penyusunan skripsi ini dapat terlaksana dengan baik dan terarah.

5. Seluruh dosen, staf, dan civitas akademika Program Studi Sistem Informasi Universitas Islam Indragiri, yang telah memberikan ilmu, bimbingan, serta pengalaman berharga selama perkuliahan, sehingga membantu penulis berkembang dan menyelesaikan skripsi ini dengan baik.

56 182 114 2 Kedua orang tua serta keluarga tercinta, atas kasih sayang, doa, dan pengorbanan yang tak ternilai, yang selalu menjadi sumber semangat dan kekuatan bagi penulis dalam menempuh setiap tahap perjalanan studi hingga penyelesaian skripsi ini.

7. Rekan-rekan tim pengembang SiKosa yang telah memberikan dukungan serta izin kepada penulis untuk melaksanakan penelitian pada sistem SiKosa.

2 8. Teman-teman seperjuangan serta sahabat diskusi, yang senantiasa memberikan dukungan, berbagi wawasan, dan menemani penulis melalui proses panjang dalam penyusunan skripsi ini.

8 Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk

penyempurnaan ke depan. Semoga skripsi ini dapat memberikan manfaat dan menjadi dasar yang kuat bagi penelitian selanjutnya.

Tembilahan, 12 Juli 2026

Penulis

MUHAMMAD AIDIL

403221010060

ABSTRAK

Sistem Konseling Mahasiswa (SiKosa) merupakan platform berbasis web yang dikembangkan untuk menyediakan layanan konseling secara daring bagi mahasiswa. Penelitian ini menerapkan pengujian otomatis pada SiKosa menggunakan *framework Jest* dengan pendekatan *Software Testing Life Cycle* (STLC). Tujuan penelitian adalah memastikan sistem berjalan sesuai kebutuhan, menyusun *test case* yang terstruktur, serta mengevaluasi standar cakupan pengujian. Pengujian dilakukan melalui dua pendekatan, yaitu *unit testing* pada *service layer* dan *integration testing* pada *API endpoint*. Hasil pengujian menunjukkan bahwa seluruh 506 *test case*, yang terdiri atas 248 *unit test* dan 258 *integration test* dalam 26 *test suite*, berhasil dijalankan dengan tingkat keberhasilan 100%. Selain itu, nilai *code coverage* yang diperoleh meliputi *statement coverage* sebesar 82,13%, *branch coverage* 87,06%, *function coverage* 84,87%, dan *line coverage* 82,13%, sehingga seluruhnya telah melampaui target minimum sebesar 80%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa implementasi pengujian otomatis menggunakan *Jest* mampu memverifikasi kebutuhan SiKosa secara efektif, meningkatkan keandalan sistem, serta memberikan dasar yang kuat untuk menjaga kualitas perangkat lunak selama proses pengembangan.

Kata kunci: pengujian otomatis, *Jest*, SiKosa, *Software Testing Life Cycle*, konseling mahasiswa

ABSTRACT

The Student Counseling System (SiKosa) is a web-based platform developed to provide online counseling services for students. This study applied automated testing to SiKosa using the Jest framework and the Software Testing Life Cycle (STLC) approach. The objectives of this study were to ensure the system functions as intended, to develop structured test cases, and to evaluate testing coverage standards. Testing was conducted using two approaches: unit testing at the service layer and integration testing at the API endpoints. The test results show that all 506 test cases—comprising 248 unit tests and 258 integration tests across 26 test suites—were successfully executed with a 100% success rate. In addition, the code coverage values obtained included statement coverage of 82.13%, branch coverage of 87.06%, function coverage of 84.87%, and line coverage of 82.13%, all of which exceeded the minimum target of 80%. Based on these results, it can be concluded that the implementation of automated testing using Jest is capable of effectively verifying SiKosa's requirements, improving system reliability, and providing a strong foundation for maintaining software quality throughout the development process.

Keywords: *automated testing, Jest, SiKosa, Software Testing Life Cycle, student counseling*

117

DAFTAR ISI

15

KATA PENGANTAR.....	vi
ABSTRAK	ix
<i>ABSTRACT</i>	x
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian	6
1.6 Sistematika Penulisan.....	7
BAB II.....	9
TINJAUAN LITERATUR	9
2.1 Landasan Teori.....	9
2.1.1 Pengujian Perangkat Lunak	9

16	2.1.2 Jenis-Jenis Pengujian Perangkat Lunak.....	9
	2.1.3 Pengujian Manual dan Otomatis	10
	2.1.4 Pengujian Otomatis Perangkat Lunak	11
	2.2 Literatur Review	13
50	2.3 Kesimpulan Literatur.....	18
	BAB III.....	20
	METODE PENELITIAN	20
	3.1 Kerangka Penelitian	20
	3.1.1 Fase Awal	21
	3.1.2 Fase Pengujian	22
65	3.1.3 Fase Akhir	25
	BAB IV	26
	HASIL DAN PEMBAHASAN	26
	4.1 Fase Awal.....	26
	4.1.1 Identifikasi Kebutuhan Pengujian	26
	4.2 Fase Pengujian	27
33	4.2.1 <i>Requirement Analysis</i>	27
	4.2.2 <i>Test Planning</i>	30
	4.2.3 <i>Test Case Development</i>	34
	4.2.4 <i>Test Environment Setup</i>	152
	4.2.5 <i>Test Execution</i>	157

4.2.6 <i>Test Closure</i>	160
4.3 Fase Akhir	164
4.3.1 Analisis Hasil Pengujian	164
BAB V	170
PENUTUP	170
5.1 Kesimpulan	170
5.2 Saran	171
DAFTAR PUSTAKA	173
LAMPIRAN	177

DAFTAR TABEL

2	Tabel 2. 1 Review Jurnal Terdahulu.....	13
	Tabel 4. 1 Framework dan Tools Pengujian	30
71	Tabel 4. 2 Rencana Cakupan Pengujian.....	31
	Tabel 4. 3 Jenis Coverage.....	32
	Tabel 4. 4 Kategori Test Coverage.....	32
	Tabel 4. 5 Test Coverage SiKosa	33
	Tabel 4. 6 Parameter Kelengkapan RTM	34
	Tabel 4. 7 Main Map Fitur Mahasiswa	35
19	Tabel 4. 8 <i>Test Scenario</i> Mahasiswa	41
	Tabel 4. 9 <i>Test Case Login</i> Mahasiswa	42
22	Tabel 4. 10 <i>Test Case Registrasi</i> Mahasiswa.....	48
	Tabel 4. 11 <i>Test Case Logout</i> Mahasiswa	65
	Tabel 4. 12 <i>Test Case Kelola</i> Profil Mahasiswa	68
	Tabel 4. 13 <i>Test Case</i> Konsultasi Mahasiswa.....	72
	Tabel 4. 14 <i>Test Case Chat</i> Mahasiswa.....	78
19	Tabel 4. 15 <i>Test Case Chatbot</i> Mahasiswa.....	85
	Tabel 4. 16 RTM Mahasiswa.....	88
	Tabel 4. 17 <i>Main Map</i> Fitur Psikolog	89
	Tabel 4. 18 <i>Test Scenario</i> Psikolog	94
19	Tabel 4. 19 <i>Test Case Login</i> Psikolog	95
	Tabel 4. 20 <i>Test Case Logout</i> Psikolog	99
195	Tabel 4. 21 <i>Test Case</i> Kelola Profil Psikolog	102
	Tabel 4. 22 <i>Test Case</i> Konsultasi Psikolog.....	107

45	Tabel 4. 23 <i>Test Case</i> Chat Psikolog.....	112
	Tabel 4. 24 <i>Test Case</i> Kelola Artikel Psikolog	119
	Tabel 4. 25 RTM Psikolog.....	124
	Tabel 4. 26 <i>Main Map</i> Fitur Admin	126
	Tabel 4. 27 <i>Test Scenario</i> Admin	129
86	Tabel 4. 28 <i>Test Case Login</i> Admin	130
	Tabel 4. 29 <i>Test Case Logout</i> Admin	134
22	Tabel 4. 30 <i>Test Case Kelola User</i> Admin	137
	Tabel 4. 31 <i>Test Case Kelola</i> Artikel (Admin)	143
	Tabel 4. 32 RTM Admin.....	149
	Tabel 4. 33 Rekapitulasi Total <i>Test Case</i> per Modul	150
	Tabel 4. 34 Data Simulasi yang Digunakan	156
	Tabel 4. 35 Hasil Eksekusi <i>Test Case</i> per Modul	158
	Tabel 4. 36 Rincian Hasil Eksekusi per File <i>Test Suite</i>	159
	Tabel 4. 37 Hasil <i>Code Coverage</i> Pengujian.....	161
	Tabel 4. 38 Ringkasan Hasil Test Closure.....	163
	Tabel 4. 39 Perbandingan Hasil <i>Unit Testing</i> dan <i>Integration Testing</i>	166

DAFTAR GAMBAR

Gambar 3. 1 Kerangka Penelitian.....	21
Gambar 3. 2 Tahapan <i>Software Testing Life Cycle</i>	22
Gambar 4. 1 Fitur <i>role</i> mahasiswa.....	28
Gambar 4. 2 Fitur <i>role</i> psikolog.....	28
Gambar 4. 3 Fitur <i>role</i> admin	29
Gambar 4. 4 Halaman <i>login</i> mahasiswa.....	36
Gambar 4. 5 Halaman <i>registrasi</i> mahasiswa.....	37
Gambar 4. 6 Halaman <i>kelola</i> profil	37
Gambar 4. 7 Halaman <i>konsultasi</i>	38
Gambar 4. 8 Halaman <i>chat</i>	39
Gambar 4. 9 Halaman <i>chatbot</i>	40
Gambar 4. 10 Fitur <i>logout</i>	40
Gambar 4. 11 Halaman <i>login</i> psikolog	90
Gambar 4. 12 Halaman <i>kelola</i> profil psikolog	90
Gambar 4. 13 Halaman <i>konsultasi</i> (psikolog)	91
Gambar 4. 14 Halaman <i>chat</i> (psikolog)	92
Gambar 4. 15 Halaman <i>kelola</i> artikel	93
Gambar 4. 16 Fitur <i>logout</i>	93
Gambar 4. 17 Halaman <i>login</i> (admin)	126
Gambar 4. 18 Halaman <i>kelola</i> <i>user</i>	127
Gambar 4. 19 Halaman <i>kelola</i> artikel (admin)	128
Gambar 4. 20 Fitur <i>logout</i>	129
Gambar 4. 21 Cuplikan kode <i>unit test service login</i>	151

55	Gambar 4. 22 Cuplikan kode <i>integration test</i> konsultasi.....	152
	Gambar 4. 23 Instalasi <i>devDependencies</i> pengujian	153
	Gambar 4. 24 Konfigurasi Pengujian.....	154
	Gambar 4. 25 Struktur folder pengujian	155
	Gambar 4. 26 Konfigurasi <i>Custom Reporter</i>	156
	Gambar 4. 27 Proses eksekusi pengujian pada terminal	157
158	Gambar 4. 28 Hasil eksekusi pengujian.....	158
	Gambar 4. 29 Hasil laporan <i>test coverage</i>	161
	Gambar 4. 30 <i>Dashboard</i> hasil pengujian (<i>Jest-stare</i>)	162

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem Konseling Mahasiswa (SiKosa) merupakan sebuah platform berbasis web yang dirancang untuk memfasilitasi kebutuhan konseling mahasiswa secara daring. Melalui SiKosa, proses konseling dapat dilakukan secara online, sehingga mahasiswa memiliki fleksibilitas yang lebih tinggi dalam menentukan waktu dan tempat untuk melakukan sesi konsultasi pribadi. Dalam implementasinya, sistem ini perlu diuji secara menyeluruh untuk memastikan setiap fitur seperti pendaftaran akun, pengajuan konsultasi, dan percakapan daring berfungsi dengan baik tanpa gangguan.

Peningkatan kompleksitas fitur berpotensi menimbulkan bug atau malfungsi, sehingga kualitas dan stabilitas sistem menjadi faktor krusial dalam memastikan kepuasan pengguna [1]. Oleh karena itu, dibutuhkan proses pengujian yang sistematis dan efisien untuk menjamin seluruh fungsi sistem bekerja sesuai kebutuhan pengguna serta bebas dari kesalahan [2]. Selain itu, pengujian merupakan salah satu tahap dalam *System Development Life Cycle* (SDLC), yang secara lebih spesifik dijabarkan melalui *Software Testing Life Cycle* (STLC) sebagai metode khusus dalam proses pengujian [3].

Secara umum, metode pengujian perangkat lunak terbagi menjadi manual dan otomatis [4]. Pengujian manual tergolong mudah untuk diterapkan, namun cenderung memiliki risiko kesalahan yang tinggi dan kurang efisien dalam

pelaksanaannya [5]. Pengujian manual menjadi kurang efektif dalam pengembangan aplikasi web seperti SiKosa yang mengalami pembaruan atau penambahan fitur secara berkala, karena membutuhkan pengulangan uji secara terus-menerus.

Berbagai penelitian sebelumnya menunjukkan bahwa pengujian pada aplikasi berbasis web telah banyak dilakukan, namun sebagian besar masih mengandalkan pendekatan manual. Penelitian yang dilakukan oleh Jailani dan Yaqin mengungkapkan bahwa teknik *Black Box Testing* dengan pendekatan *Boundary Value Analysis* cukup efektif dalam menemukan kesalahan pada batas nilai input, tetapi penerapannya masih sepenuhnya manual tanpa dukungan otomatisasi. Penelitian lain yang disusun oleh Shaleh serta Widhyaestoeti dan rekan-rekan, juga memanfaatkan teknik *Equivalent Partitions* untuk menguji validitas input maupun komponen antarmuka sistem. Namun, proses pengujiannya tetap dijalankan secara manual dan belum memberikan cakupan uji yang luas. Sementara itu, Mintarsih menerapkan pendekatan *State Transition Testing* pada sistem perpustakaan berbasis web untuk memeriksa konsistensi perubahan status di dalam aplikasi, dan Kinasih bersama tim menggunakan *White Box Testing* untuk mengevaluasi kompleksitas logika internal sistem. Walaupun berbagai penelitian tersebut memberikan kontribusi pada praktik pengujian perangkat lunak, semuanya masih berfokus pada metode manual dan belum memanfaatkan pengujian otomatis.

Kondisi ini menunjukkan bahwa masih terdapat kebutuhan kuat untuk menerapkan pengujian otomatis pada aplikasi berbasis web, khususnya sistem layanan seperti SiKosa, agar proses verifikasi dapat dilakukan lebih cepat, konsisten, dan memiliki cakupan yang lebih menyeluruh.

Oleh karena itu, diperlukan metode pengujian yang lebih terstruktur dan efisien, yakni pengujian otomatis. Metode ini memanfaatkan alat bantu untuk menjalankan proses pengujian sehingga mempercepat proses pengujian serta meningkatkan kinerja dan produktivitas [6]. Dengan pendekatan tersebut, proses verifikasi dapat dilakukan lebih cepat dan konsisten.

Dalam penerapan pengujian otomatis, diperlukan alat bantu (*testing framework*) yang mampu menjalankan pengujian dengan cepat, mudah dikonfigurasi, serta sesuai dengan bahasa pemrograman yang digunakan dalam pengembangan aplikasi [7]. Dari berbagai *framework* pengujian otomatis, *Jest* dipilih karena ringan, mudah diintegrasikan dengan proyek berbasis *JavaScript*, serta memiliki dukungan komunitas yang luas. *Jest* mendukung unit, integrasi, dan *snapshot test*, serta dilengkapi fitur seperti *automatic mocking*, *coverage reporting*, dan *parallel execution* [8]. Karena SiKosa dibangun dengan teknologi *JavaScript*, *Jest* menjadi pilihan ideal untuk pengujian website.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menerapkan pengujian otomatis terhadap website Sistem Konseling Mahasiswa (SiKosa) dengan menggunakan *Jest*. Melalui penerapan ini, diharapkan sistem dapat berfungsi secara andal dan stabil, sekaligus meningkatkan efisiensi serta efektivitas dalam proses pengujian.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian, pokok permasalahan dalam penelitian ini adalah implementasi pengujian otomatis pada website Sistem Konseling

Mahasiswa (SiKosa) menggunakan *Jest*. Adapun sub-permasalahan yang dirumuskan adalah sebagai berikut:

1. Website SiKosa masih mengandalkan pengujian manual dan belum menerapkan pengujian otomatis.
2. Rancangan pengujian berupa *test case* terstruktur belum tersedia, serta belum dapat dipastikan apakah website SiKosa telah berjalan sesuai kebutuhan pengguna.
3. Belum terdapat standar dan cakupan pengujian (*test coverage*) yang jelas.

1.3 Batasan Masalah

Agar penelitian ini terfokus dan sesuai dengan tujuan yang telah ditetapkan, maka batasan masalah dalam penelitian ini dirumuskan sebagai berikut:

1. Penelitian hanya berfokus pada pengujian otomatis, tanpa membahas aspek non-fungsional seperti keamanan, performa, atau *usability*.
2. Pengujian dilakukan menggunakan *Jest* sebagai *framework* utama, tanpa membandingkan dengan *tools* otomatisasi lain.
3. Ruang lingkup pengujian dibatasi pada pengujian kode (*code-level testing*), tanpa melibatkan pengujian berbasis *end user*.
4. Ruang lingkup pengujian mencakup fitur-fitur yang ada pada SiKosa:
 - a) Autentikasi, login, registrasi, dan logout pengguna
 - b) Fitur manajemen profil
 - c) Fitur konsultasi
 - d) Fitur chatbot
 - e) Fitur manajemen data user

f) Fitur manajemen artikel

5. Pengujian dilakukan pada versi website SiKosa per 11 November 2025, sehingga hasil pengujian hanya berlaku untuk kondisi sistem pada versi tersebut.
6. Data hasil pengujian terbatas pada laporan eksekusi *test case* yang dihasilkan oleh *Jest*, tanpa analisis mendalam terhadap performa server atau beban sistem.
7. Data pengujian menggunakan data simulasi, sehingga hasil pengujian tidak mencerminkan data pengguna asli.

1.4 Tujuan Penelitian

Berdasarkan pada rumusan masalah sebelumnya, tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan pengujian otomatis pada website SiKosa menggunakan *Jest* untuk memastikan sistem berjalan sesuai kebutuhan, serta meningkatkan efektivitas, efisiensi, dan konsistensi proses pengujian yang juga relevan untuk pengujian sistem berbasis web secara umum.
2. Menyusun *test case* yang sistematis dan melakukan evaluasi hasil pengujian untuk memastikan kesesuaian sistem dengan kebutuhan pengguna, sekaligus menyediakan pendekatan perancangan pengujian yang dapat dijadikan acuan dalam pengembangan perangkat lunak.
3. Menentukan standar dan cakupan pengujian (*test coverage*) pada sistem SiKosa melalui pengujian otomatis, sehingga dapat memberikan gambaran

penerapan kontrol kualitas pengujian yang lebih terukur dan dapat diterapkan pada sistem serupa.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Meningkatkan kualitas dan keandalan sistem melalui penerapan pengujian otomatis, penelitian ini membantu memastikan bahwa sistem dapat berjalan sesuai kebutuhan secara lebih konsisten dan efisien dibandingkan pengujian manual. Selain itu, pendekatan ini dapat diterapkan pada berbagai sistem berbasis web lainnya untuk meningkatkan kualitas perangkat lunak secara umum.
2. Menyediakan acuan dalam perancangan pengujian perangkat lunak, penyusunan *test case* yang sistematis dan terstruktur dalam penelitian ini dapat dijadikan referensi dalam merancang proses pengujian yang lebih terorganisir. Hal ini tidak hanya bermanfaat bagi sistem yang diteliti, tetapi juga dapat diterapkan pada pengembangan sistem lain yang membutuhkan standar pengujian yang jelas.
3. Membantu dalam evaluasi dan pengendalian kualitas sistem, dengan adanya hasil evaluasi pengujian, penelitian ini memberikan gambaran tingkat kesesuaian sistem terhadap kebutuhan pengguna. Selain itu, hasil tersebut dapat digunakan sebagai dasar dalam proses perbaikan dan pengembangan sistem secara berkelanjutan.
4. Memberikan gambaran standar dan cakupan pengujian (*test coverage*), penelitian ini membantu dalam mengidentifikasi bagian sistem yang telah

diuji maupun yang belum diuji, sehingga proses pengujian menjadi lebih terarah. Konsep ini juga dapat diterapkan secara umum dalam praktik pengujian perangkat lunak untuk meningkatkan efektivitas pengujian.

1.6 Sistematika Penulisan

Sistematika penulisan karya ilmiah bertujuan untuk memberikan gambaran umum mengenai isi dan susunan bab dalam penelitian ini. Adapun sistematika penulisan dalam laporan penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi gambaran umum penelitian yang mencakup latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan. Bagian ini memberikan pemahaman awal mengenai alasan, urgensi, dan arah penelitian yang dilakukan.

BAB II TINJAUAN LITERATUR

Bab ini membahas teori, konsep, dan hasil penelitian terdahulu yang relevan dengan topik pengujian perangkat lunak, khususnya pengujian otomatis menggunakan *framework Jest* yang digunakan sebagai landasan teori dalam penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini menguraikan metode dan tahapan penelitian yang dilakukan, meliputi pendekatan penelitian, objek penelitian, perancangan *test case*, serta langkah-langkah implementasi pengujian otomatis menggunakan *Jest*.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penelitian yang diperoleh setelah tahap implementasi sistem. Pembahasan difokuskan pada analisis efektivitas sistem yang dikembangkan, kinerja sistem, serta evaluasi berdasarkan hasil pengujian yang telah dilakukan. Analisis tersebut digunakan untuk menilai sejauh mana sistem memenuhi tujuan dan kebutuhan yang telah ditetapkan.

BAB V PENUTUP

Bab ini memuat kesimpulan yang disusun berdasarkan hasil penelitian dan pembahasan yang telah dilakukan. Selain itu, bab ini juga menyajikan saran sebagai rekomendasi untuk pengembangan sistem maupun penyempurnaan penelitian pada masa mendatang.

59

BAB II

TINJAUAN LITERATUR

Bab ini menguraikan berbagai hasil penelitian sebelumnya yang berkaitan dengan topik yang dibahas, serta menyajikan sumber-sumber rujukan yang mendukung penelitian ini. Dengan demikian, tinjauan literatur berfungsi memberikan gambaran perbandingan yang jelas antara penelitian terdahulu dan penelitian saat ini.

2.1 Landasan Teori

2.1.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan proses penting dalam pengembangan sistem yang bertujuan untuk menjamin kualitas, kinerja, serta keamanan produk yang dihasilkan. Proses ini mencakup evaluasi terhadap sistem atau komponen perangkat lunak untuk mengidentifikasi kesalahan serta memastikan bahwa sistem berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Oleh karena itu, pengujian menjadi tahap krusial karena berperan dalam memastikan kualitas perangkat lunak, baik dari sisi perancangan maupun struktur kontrol pemrogramannya [9].

2.1.2 Jenis-Jenis Pengujian Perangkat Lunak

a) *White Box Testing* dan *Black Box Testing*

Secara umum, terdapat beberapa pendekatan dalam pengujian perangkat lunak, di antaranya *White Box Testing* dan *Black Box Testing*. *White Box Testing*

193

134

24

91

100

189

79

dilakukan dengan menganalisis struktur internal dan kode program untuk mendeteksi kesalahan logika maupun implementasi pada level modul. Sementara itu, *Black Box Testing* berfokus pada pengujian dari sisi eksternal, yaitu dengan mengevaluasi fungsi-fungsi aplikasi, tampilan antarmuka, serta kesesuaian alur sistem dengan spesifikasi yang telah dirancang, tanpa memperhatikan detail kode program di dalamnya [10].

b) *Unit testing* dan *Integration Testing*

Selain itu, terdapat *Unit Testing* dan *Integration Testing* dalam praktik pengujian perangkat lunak. *Unit Testing* dilakukan untuk menguji unit terkecil kode agar berfungsi dengan benar dan konsisten. Selanjutnya, *Integration Testing* bertujuan menguji gabungan beberapa unit untuk memastikan interaksi antar komponen berjalan sesuai harapan [11].

2.1.3 Pengujian Manual dan Otomatis

Pengujian manual merupakan metode pengujian yang dilakukan secara langsung oleh manusia dengan mengikuti skenario yang telah ditentukan untuk mengevaluasi fungsionalitas sistem. Metode ini memiliki keterbatasan, terutama dari segi waktu dan ketergantungan pada kemampuan penguji dalam menemukan serta melaporkan kesalahan. Sebaliknya, pengujian otomatis memanfaatkan perangkat lunak untuk mengeksekusi skenario dan kasus uji secara sistematis. Pendekatan ini lebih efisien karena mampu menghemat waktu dan tenaga, serta efektif dalam menangani pengujian dengan tingkat kompleksitas tinggi dan volume data yang besar [12].

2.1.4 Pengujian Otomatis Perangkat Lunak

a) Pengujian Otomatis / *Automation testing*

Pengujian otomatis merupakan pengembangan dari pengujian manual yang memanfaatkan skrip dan tools untuk menjalankan pengujian secara berulang dan terstruktur. Pendekatan ini umumnya digunakan dalam pengembangan perangkat lunak serta meningkatkan efisiensi pengujian.

b) *Testing Framework*

Untuk mendukung pelaksanaan pengujian otomatis, diperlukan suatu *testing framework* sebagai kerangka kerja yang menyediakan aturan, struktur, serta alat bantu dalam merancang, menjalankan, dan mengevaluasi proses pengujian perangkat lunak. *Framework* ini umumnya dilengkapi dengan fitur pengelolaan *test case*, mekanisme *assertion*, serta pembuatan laporan hasil pengujian. Dengan demikian, penggunaan *testing framework* dapat meningkatkan keteraturan, konsistensi, dan kemudahan pemeliharaan dalam proses pengujian perangkat lunak [13].

c) *Jest*

Jest merupakan *framework* pengujian yang dikembangkan untuk mendukung pengujian unit dan integrasi dengan performa yang cepat serta konfigurasi yang sederhana. *Jest* memiliki berbagai fitur unggulan seperti *zero configuration*, *snapshot testing*, *automatic mocking*, *code coverage reporting*, dan kemampuan menjalankan pengujian secara paralel. Selain itu, *Jest* juga dikenal mampu menangani pengujian secara paralel serta mendukung deteksi permasalahan seperti *flaky test* yang dapat memengaruhi konsistensi hasil pengujian [14].

d) *Test case* dan *Test coverage*

Dalam pengujian otomatis, *test case* digunakan sebagai acuan untuk menguji fungsi atau fitur sistem. *Test case* merupakan kumpulan skenario yang dirancang untuk memverifikasi apakah sistem menghasilkan output sesuai dengan yang diharapkan [15]. Umumnya, *test case* mencakup identitas pengujian, deskripsi, input, langkah pengujian, serta hasil yang diharapkan.

Selain itu, digunakan *test coverage* sebagai metrik untuk mengukur sejauh mana pengujian mencakup bagian kode program. *Test coverage* menunjukkan persentase kode yang telah diuji, dengan beberapa jenis seperti statement coverage, branch coverage, dan function coverage [16]. Semakin tinggi nilai coverage, semakin luas cakupan pengujian, meskipun tidak menjamin sistem sepenuhnya bebas dari kesalahan.

e) *Software Testing Life Cycle* (STLC)

Agar proses pengujian berjalan secara sistematis dan terstruktur, digunakan pendekatan *Software Testing Life Cycle* (STLC). STLC merupakan serangkaian tahapan yang dilakukan dalam proses pengujian perangkat lunak, mulai dari analisis kebutuhan hingga evaluasi hasil pengujian [17]. Dengan menerapkan STLC, proses pengujian dapat dilakukan secara terarah, terdokumentasi dengan baik, serta mampu meningkatkan kualitas dan keandalan sistem secara keseluruhan.

Berdasarkan uraian tersebut, kombinasi pengujian otomatis dengan *framework* seperti *Jest*, *test case* terstruktur, *test coverage*, dan penerapan STLC efektif meningkatkan kualitas perangkat lunak. Pendekatan ini membuat pengujian

lebih efisien, konsisten, dan andal dalam mendukung pengembangan sistem web seperti SiKosa.

2.2 Literatur Review

Berdasarkan telaah terhadap jurnal-jurnal sebelumnya, diperoleh hasil perbandingan yang disajikan pada tabel berikut.

Tabel 2. 1 Review Jurnal Terdahulu

No	Tinjauan Pustaka	Kesimpulan
(1)	(2)	(3)
1	Ahmad Jailani, Muhammad Ainul yaqin (2024) Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik <i>Boundary Value Analysis</i>	Artikel ini menekankan pentingnya pengujian perangkat lunak untuk menjaga kualitas aplikasi akademik dengan menggunakan metode <i>Black Box Testing</i> dan teknik <i>Boundary Value Analysis</i> (BVA). Pengujian ini efektif mendeteksi error pada batas input dan menghasilkan temuan bahwa hanya 60% pengujian yang berhasil. Namun, penelitian memiliki keterbatasan, seperti tidak dianalisisnya penyebab kegagalan secara mendalam, tidak adanya otomatisasi pengujian, serta cakupan pengujian yang belum menyeluruh.
2	Ibnu Adha Shaleh, Juma Prayogi, Perdi Pirdaus, Riky Syawal, Aries Saifudin (2021) Pengujian <i>Black Box</i> pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik <i>Equivalent Partitions</i>	Artikel ini menyoroti pentingnya kualitas perangkat lunak pada sistem informasi berbasis web dan menggunakan <i>Black Box Testing</i> dengan teknik <i>Equivalent Partitions</i> untuk menguji validitas input pada form aplikasi penjualan buku. Teknik ini memberikan struktur pengujian yang lebih baik dan relevan untuk sistem berbasis web. Namun, penelitian ini memiliki keterbatasan karena seluruh pengujian dilakukan secara manual dan tidak menyertakan data kuantitatif seperti jumlah keberhasilan atau kegagalan <i>test case</i> .
3	Dahlia Widhyaestoeti, Saidul Iqram, Siti Nur Mutiyah, Yasmin Khairunnisa (2021) <i>Black Box Testing</i> Equivalence Partitions Untuk Pengujian Front-End Pada Sistem Akademik Sitoda	Penelitian ini menguji front-end sistem akademik SITODA berbasis web menggunakan metode <i>Black Box Testing</i> dengan teknik <i>Equivalence Partitioning</i> . Pengujian disusun melalui tabel perencanaan dan tabel fungsional, serta melibatkan end-user untuk memastikan relevansi kebutuhan. Teknik ini membuat <i>test case</i> lebih terstruktur dan meningkatkan keandalan antarmuka. Namun, pengujian hanya mencakup front-end, dilakukan secara manual, dan hasilnya bersifat deskriptif tanpa data kuantitatif.

Tabel 2.1 Review Jurnal Terdahulu (Lanjutan)

(1)	(2)	(3)
4	Mintarsih (2023) Pengujian <i>Black Box</i> Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation	Penelitian ini menguji Sistem Informasi Perpustakaan SMC Foundation berbasis web menggunakan metode <i>Black Box Testing</i> dengan teknik <i>State Transition Testing</i> . Teknik ini memastikan setiap perpindahan state dalam sistem berjalan sesuai ekspektasi. Pendekatan tersebut relevan untuk sistem dengan banyak perubahan status dan mudah diterapkan karena berfokus pada input-output. Namun, pengujian masih dilakukan secara manual dan tidak disertai data kuantitatif mengenai jumlah transisi atau tingkat keberhasilan pengujian.
5	Muhammad Jibril, Zulrahmadi, Muhammad Amin (2024) Pengujian Sistem Informasi E-Modul Pada Smpn 1 Tempuling Menggunakan <i>Black Box Testing</i>	Penelitian ini menguji fungsionalitas sistem informasi e-modul di SMPN 1 Tempuling menggunakan metode <i>Black Box Testing</i> , yang berfokus pada perilaku aplikasi tanpa melihat kode sumber. Hasil pengujian menunjukkan bahwa sistem telah sesuai spesifikasi dan mudah diuji karena metodenya sederhana. Namun, pengujian dilakukan secara manual dan tidak disertai data kuantitatif seperti jumlah <i>test case</i> atau tingkat keberhasilan.
6	Dona Yulawati, Anggi Andriyadi, Nursiyanto (2022) Pengujian Sistem Informasi E-Monitoring Pengelolaan Pembangunan Desa Dengan Menggunakan Metode <i>Black Box Testing</i>	Penelitian ini menguji sistem informasi e-monitoring pembangunan desa menggunakan metode <i>Black Box Testing</i> dengan total 42 skenario uji untuk menilai kesesuaian fungsi sistem. Hasil menunjukkan 95% skenario berjalan valid, sementara beberapa lainnya masih memerlukan perbaikan. Pengujian dinilai objektif karena menyertakan data kuantitatif, meskipun seluruh proses masih dilakukan secara manual.
7	Jafar Shadiq, Ahmad Safei, Rayhan Wahyudin Ratu Loly (2021) Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan <i>Black Box Testing</i>	Penelitian ini menguji aplikasi sistem informasi peminjaman kendaraan operasional menggunakan metode <i>Black Box Testing</i> dengan teknik <i>Equivalence Partitioning</i> . Teknik ini mengelompokkan input ke dalam kelas valid dan tidak valid untuk memastikan sistem memberikan output sesuai harapan. Pendekatan ini membuat pengujian lebih terstruktur dan sistematis, meskipun masih dilakukan secara manual dan tidak disertai data kuantitatif mengenai hasil <i>test case</i> .

Tabel 2.1 Review Jurnal Terdahulu (Lanjutan)

(1)	(2)	(3)
8	Arif Samdono, Anggraini Puspita Sari, Firza Prima Aditiawan (2024) Pengujian <i>Black Box</i> Pada Sistem Informasi Stok dan Penjualan Berbasis Website Menggunakan Metode <i>Equivalence Partitioning</i> (Studi Kasus: Cv. Algani Karya Mandiri)	Artikel ini menguji Sistem Informasi Stok dan Penjualan berbasis web di CV. Algani Karya Mandiri menggunakan metode <i>Black Box Testing</i> dengan teknik <i>Equivalence Partitioning</i> untuk memastikan fungsionalitas sistem dan mendeteksi potensi kesalahan. Hasil menunjukkan efektivitas sistem sebesar 91,87%, menandakan tingkat keandalan yang cukup baik. Namun, beberapa masalah masih ditemukan dan pengujian dilakukan sepenuhnya secara manual tanpa pembahasan mengenai otomatisasi
9	Amanda Amalia, Salva Wanda Putri Hamidah, Titus Kristanto (2021) Pengujian <i>Black Box</i> Menggunakan Teknik <i>Equivalence Partitions</i> Pada Aplikasi E-Learning Berbasis Web	Artikel ini menguji kualitas aplikasi E-Learning berbasis web di Institut Teknologi Telkom Surabaya menggunakan metode <i>Black Box Testing</i> dengan teknik <i>Equivalence Partitioning</i>. Pengujian dilakukan melalui identifikasi fungsi, perancangan <i>test case</i>, eksekusi uji, dan penarikan kesimpulan. Teknik ini efektif karena membagi data uji ke dalam kelas valid dan tidak valid. Namun, hasil pengujian hanya menyatakan bahwa aplikasi layak digunakan tanpa data kuantitatif, dan proses pengujian masih sepenuhnya manual tanpa pembahasan otomasi.
10	Fitriana Sekar Kinasih, Habib Nur Gian, Hanifah Permatasari (2023) Pengujian Sistem Informasi Inventaris Barang Berbasis Website Menggunakan Metode <i>Whitebox Testing</i>	Artikel ini menguji sistem informasi inventaris barang berbasis web pada halaman distributor dengan menggunakan metode <i>White Box Testing</i> melalui analisis <i>Cyclomatic Complexity</i>. Pengujian ini mengevaluasi alur logika internal sistem untuk mengukur kompleksitas dan mengidentifikasi jalur independen yang berpotensi menimbulkan error. Namun, objek uji hanya terbatas pada satu halaman, tidak membahas aspek fungsional dari sisi pengguna, dan belum mengombinasikan metode lain untuk hasil yang lebih menyeluruh.
11	Philipp Straubinger, Tommaso Fulcini, dkk (2024) <i>IntelliGame in Action: An Experience Report on Gamifying JavaScript Unit Tests</i>	Penelitian ini mengkaji integrasi IntelliGame, yaitu plugin berbasis gamifikasi, dalam pengujian unit <i>JavaScript</i> menggunakan <i>Jest</i> melalui eksperimen terkontrol terhadap 152 partisipan. Hasil penelitian menunjukkan bahwa penerapan gamifikasi dalam proses pengujian mampu meningkatkan keterlibatan serta memengaruhi perilaku developer dalam menulis dan menjalankan pengujian.

Tabel 2.1 Review Jurnal Terdahulu (Lanjutan)

(1)	(2)	(3)
12	Negar Hashemi, dkk (2025) Detecting and Evaluating Order-Dependent Flaky tests in JavaScript	Penelitian ini mengkaji fenomena <i>flaky test</i> pada pengujian <i>JavaScript</i> menggunakan <i>Jest</i> , khususnya yang disebabkan oleh ketergantungan urutan eksekusi (test order dependency). Melalui eksperimen terhadap 81 proyek, dilakukan pengacakan urutan test dan eksekusi ulang untuk mendeteksi perubahan hasil pengujian. Hasil menunjukkan terdapat 55 kasus order-dependent tests pada 10 proyek, yang sebagian besar terjadi antar test. Penyebab utama yang ditemukan adalah penggunaan file bersama (shared files) dan kondisi mocking yang tidak terisolasi (shared mocking state). Temuan ini menegaskan bahwa selain faktor umum, terdapat penyebab baru pada <i>Jest</i> yang dapat memengaruhi stabilitas pengujian, sehingga perancangan test yang baik menjadi penting untuk menjaga keandalan hasil pengujian.

Berdasarkan tabel 2.1, ringkasan dari berbagai penelitian tersebut menunjukkan bahwa, Ahmad Jailani dan Muhammad Ainul Yaqin menekankan pentingnya pengujian perangkat lunak menggunakan *Black Box Testing* dan teknik *Boundary Value Analysis*, namun masih memiliki kelemahan berupa cakupan pengujian yang terbatas dan belum adanya otomatisasi [18]. Selanjutnya, Ibnu Adha Shaleh dkk menunjukkan bahwa teknik *Equivalent Partitions* dapat meningkatkan struktur pengujian input pada sistem berbasis web, meskipun seluruh proses masih dilakukan secara manual dan tanpa data kuantitatif [19]. Menurut Dahlia Widhyaestoeti dkk teknik *Equivalence Partitioning* juga efektif untuk menyusun *test case* front-end secara terstruktur, tetapi pengujian mereka masih manual dan hanya mencakup sisi antarmuka tanpa hasil kuantitatif [20]. Penelitian Mintarsih menegaskan bahwa teknik *State Transition Testing* relevan untuk sistem dengan banyak perubahan status, namun kembali ditemukan bahwa pengujian masih manual dan tanpa data kuantitatif pendukung [21].

174 Kemudian, Muhammad Jibril dkk membuktikan bahwa *Black Box Testing* dapat memastikan kesesuaian fungsionalitas sistem e-modul, tetapi pengujiannya masih manual dan tidak menyertakan jumlah *test case* maupun tingkat keberhasilan [22]. Sementara itu, Dona Yuliawati dkk memberikan temuan yang lebih objektif dengan mencatat 95% skenario pengujian berjalan valid dari total 42 skenario, meski seluruh proses tetap dilakukan secara manual [23]. Penelitian oleh Jafar Shadiq dkk menunjukkan bahwa teknik *Equivalence Partitioning* dapat membuat pengujian lebih terstruktur dan sistematis, namun kelemahannya serupa, masih manual dan tanpa data kuantitatif [24]. Menurut Arif Samdono dkk efektivitas sistem mencapai 91,87% melalui pengujian *Equivalence Partitioning*, menandakan reliabilitas yang baik, tetapi penelitian belum membahas otomatisasi pengujian [25]. Penelitian Amanda Amalia dkk juga menegaskan efektivitas teknik *Equivalence Partitioning* pada aplikasi e-learning, tetapi hasil pengujian bersifat deskriptif tanpa data kuantitatif dan tetap manual [26].

Fitriana Sekar Kinasih dkk menggunakan *White Box Testing* untuk mengukur kompleksitas logika sistem, namun pengujiannya terbatas pada satu halaman dan tidak mencakup aspek fungsional dari sudut pandang pengguna [27]. Philipp Straubinger dkk menunjukkan bahwa integrasi IntelliGame dengan *Jest* dalam pengujian unit *JavaScript* mampu meningkatkan keterlibatan dan perilaku developer dalam proses *testing*. Selain itu, pendekatan ini memberikan pengalaman pengujian yang lebih interaktif dan berpotensi meningkatkan kualitas serta efisiensi pengujian perangkat lunak [28]. Negar Hashemi dkk menunjukkan bahwa *flaky test* pada *Jest* banyak disebabkan oleh ketergantungan urutan eksekusi, dengan 55 kasus ditemukan pada 81 proyek. Penyebab utamanya adalah penggunaan *shared files* dan

shared mocking state, sehingga diperlukan perancangan test yang baik untuk menjaga stabilitas pengujian [29].

2.3 Kesimpulan Literatur

Berdasarkan hasil tinjauan dari berbagai penelitian sebelumnya, dapat disimpulkan bahwa mayoritas penelitian terkait pengujian perangkat lunak masih berfokus pada *Black Box Testing* yang dilakukan secara manual dengan menerapkan berbagai teknik seperti *Equivalence Partitioning*, *Boundary Value Analysis*, dan *State Transition*. Beberapa penelitian lain menerapkan pendekatan *White Box Testing*, seperti analisis *Cyclomatic Complexity* untuk mengukur kompleksitas kode dan jalur eksekusi.

Secara umum, hasil pengujian pada penelitian-penelitian tersebut menunjukkan bahwa sistem yang diuji telah berjalan sesuai kebutuhan. Namun, terdapat perbedaan antara penelitian terdahulu dengan penelitian ini dijelaskan sebagai berikut:

1. Penelitian terdahulu umumnya masih melakukan pengujian secara manual sehingga membutuhkan waktu lebih lama, kurang efisien, dan berpotensi menimbulkan *human error*. Sebaliknya, penelitian ini menerapkan otomatisasi menggunakan *Jest* sehingga pengujian unit dan integrasi dapat dijalankan secara berulang, lebih cepat, konsisten, dan mudah dipelihara.
2. Penelitian terdahulu umumnya belum memiliki dokumentasi, standar, dan cakupan pengujian yang terukur sehingga hasil pengujian sulit dilacak maupun direplikasi. Sebaliknya, penelitian ini menerapkan *Software Testing Life Cycle* (STLC) yang dilengkapi *test scenario*, *test case*, *Requirements Traceability Matrix* (RTM), serta pengukuran *test coverage*, sehingga setiap

kebutuhan sistem dapat ditelusuri dan tingkat cakupan pengujiannya dapat dievaluasi secara lebih sistematis.

3. Penelitian terdahulu cenderung hanya menguji modul atau fitur tertentu, seperti halaman admin, form input, atau fungsi tunggal, sehingga cakupan pengujian belum mewakili keseluruhan alur kerja sistem. Sebaliknya, penelitian ini melakukan pengujian pada modul-modul utama SiKosa melalui pengujian unit dan integrasi sehingga cakupan pengujian menjadi lebih komprehensif.
4. Penelitian terdahulu lebih berfokus pada validasi fungsionalitas sistem, sedangkan pengujian terhadap kualitas kode, seperti pengujian unit berbasis modular dan *mocking*, masih belum banyak diterapkan. Sebaliknya, penelitian ini mengadopsi pendekatan modular dengan dukungan *mocking* untuk memastikan stabilitas, kualitas, dan *maintainability* aplikasi dalam jangka panjang.
5. Penelitian terdahulu belum mendukung pendeteksian kesalahan secara berkelanjutan ketika terjadi perubahan kode. Sebaliknya, penelitian ini memanfaatkan fitur *watch mode* pada *Jest* untuk mendeteksi *regression bug* lebih cepat, sehingga perubahan yang berpotensi memengaruhi fungsi yang telah berjalan stabil dapat segera diidentifikasi.

27

BAB III

METODE PENELITIAN

3

Pendekatan penelitian ini adalah kuantitatif deskriptif, dengan fokus pada pengukuran hasil pengujian otomatis dalam bentuk data numerik, seperti jumlah *test case*, status berhasil atau gagal, serta persentase keberhasilan pengujian. Pendekatan ini digunakan untuk menggambarkan tingkat keberhasilan fungsi aplikasi berdasarkan hasil eksekusi pengujian menggunakan *Jest*.

11

Metode yang diterapkan adalah *Software Testing Life Cycle (STLC)*, yaitu pendekatan sistematis yang secara khusus berfokus pada proses pengujian perangkat lunak, mulai dari *test planning*, *test analysis and design*, *test execution*, serta *test closure*. Dalam penelitian ini, tahapan *test closure* digunakan sebagai bentuk evaluasi hasil pengujian, yang dilakukan melalui analisis hasil eksekusi *test case* dan perhitungan tingkat keberhasilan pengujian. Berbeda dengan SDLC yang mencakup seluruh siklus pengembangan, STLC hanya berorientasi pada aktivitas pengujian.

145

1

151

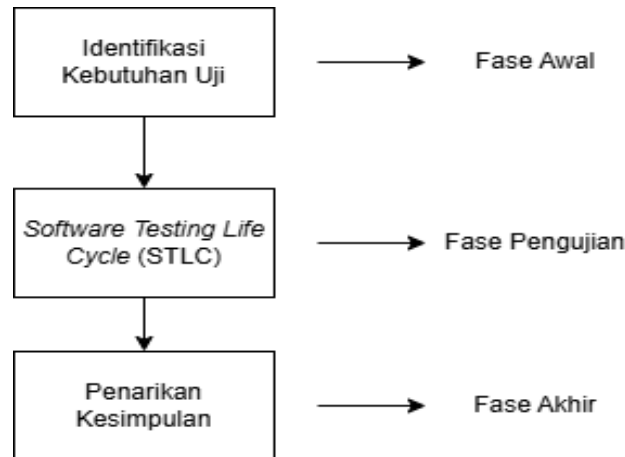
Oleh karena itu, penelitian ini tidak bertujuan menguji hipotesis, melainkan menyajikan analisis kinerja sistem berdasarkan hasil pengujian otomatis menggunakan *Jest*.

166

3.1 Kerangka Penelitian

Kerangka penelitian merupakan suatu struktur atau alur pemikiran yang digunakan untuk mengarahkan proses penelitian secara keseluruhan. Dengan adanya kerangka penelitian, setiap langkah dalam penelitian dapat disusun secara

lebih sistematis, logis, dan terarah sehingga proses penelitian menjadi lebih konsisten, mudah dipahami, serta mampu menghasilkan temuan yang sesuai dengan tujuan penelitian yang telah ditetapkan.



Gambar 3. 1 Kerangka Penelitian

Gambar 3.1 menyajikan kerangka alur penelitian yang berfungsi sebagai panduan sistematis dalam pelaksanaan studi ini. Alur penelitian diawali dengan identifikasi kebutuhan pengujian, kemudian dilanjutkan melalui tahapan metode pengujian *Software Testing Life Cycle* (STLC) hingga mencapai tahap akhir berupa penarikan kesimpulan. Kerangka ini membantu memastikan setiap proses penelitian terdokumentasi secara jelas dan terstruktur, sehingga tujuan penelitian dapat dicapai dengan lebih efektif dan terukur.

Penjelasan secara rinci disampaikan untuk menggambarkan setiap tahapan yang terdapat dalam kerangka penelitian, yang diuraikan sebagai berikut:

3.1.1 Fase Awal

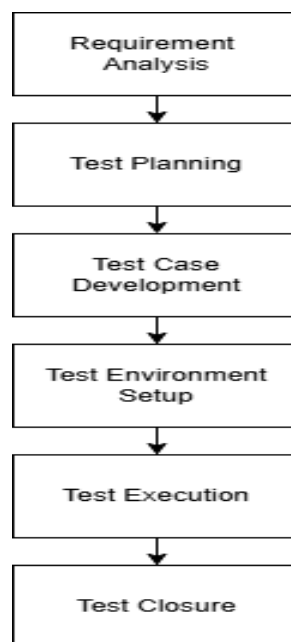
Pada tahap awal, dilakukan proses identifikasi kebutuhan pengujian pada sistem SiKosa sebagai fokus utama pengujian. Proses ini mencakup peninjauan

menyeluruh terhadap alur kerja sistem untuk menentukan fitur mana saja yang memiliki peran paling signifikan dalam mendukung sistem.

Dengan mempertimbangkan pentingnya peran setiap fitur, diperlukan jaminan bahwa seluruh fitur tersebut bekerja secara konsisten dan sesuai dengan kebutuhan yang telah ditetapkan. Langkah identifikasi ini menjadi fondasi penting agar pengujian yang dilakukan benar-benar fokus pada aspek-aspek krusial, sehingga kualitas, stabilitas, dan keandalan sistem SiKosa dapat terus terjaga dan mendukung pengalaman pengguna secara optimal.

3.1.2 Fase Pengujian

Untuk memastikan bahwa proses pengujian berjalan secara sistematis dan terukur, penelitian ini menggunakan pendekatan *Software Testing Life Cycle* (STLC). STLC merupakan serangkaian tahapan yang sistematis dalam proses pengujian perangkat lunak untuk memastikan bahwa aplikasi yang dikembangkan berfungsi sesuai dengan kebutuhan [30].



Gambar 3. 2 Tahapan *Software Testing Life Cycle*

1. *Requirement Analysis*

Tahap awal dalam STLC adalah analisis kebutuhan [31]. Pada tahap ini dilakukan analisis terhadap seluruh kebutuhan sistem SiKosa untuk memahami fitur, alur kerja, dan spesifikasi yang harus dipenuhi.

2. *Test Planning*

Tahap ini bertujuan untuk merumuskan strategi pengujian secara menyeluruh, termasuk penentuan pendekatan dan perangkat yang akan digunakan [32]. Beberapa hal yang ditetapkan antara lain:

- a) Pemilihan *Jest* sebagai *testing framework* utama, karena mendukung pengujian otomatis yang cepat, konsisten, dan mudah diintegrasikan pada lingkungan pengembangan *JavaScript*.
- b) Penentuan fokus pengujian pada sistem SiKosa dan spesifikasi cakupan pengujian (*test coverage*) berdasarkan kebutuhan sistem.

3. *Test case Development*

Test case digunakan sebagai acuan formal untuk menjelaskan mekanisme pengujian suatu fitur. Dengan demikian, setiap *test case* memberikan gambaran yang jelas dan terstruktur mengenai proses verifikasi sistem [33]. Setiap *test case* mencakup:

- a) *Scenario ID*
- b) *Test case ID*
- c) *Scenario Description*
- d) *Test Steps*
- e) *Expected Result*

4. *Test Environment Setup*

Tahap ini berfokus pada penyiapan lingkungan pengujian otomatis agar seluruh skenario yang telah dirancang dapat dijalankan dengan optimal [34]. Proses yang dilakukan mencakup beberapa langkah penting, antara lain:

1) Instalasi *tools* pengujian

Framework pengujian diinstal sebagai dependensi utama untuk memastikan seluruh skenario dapat dijalankan secara konsisten dalam lingkungan yang seragam.

2) Konfigurasi berkas pengujian

Dilakukan konfigurasi agar setiap *test case* dapat mengakses halaman atau komponen yang diuji tanpa harus menuliskan pengaturan yang berulang.

3) Penyusunan struktur direktori pengujian

Struktur folder, seperti direktori khusus untuk pengujian disusun secara sistematis untuk memudahkan organisasi *test case* serta mempermudah proses pemeliharaan di tahap selanjutnya.

4) Persiapan data uji

Data uji yang digunakan berupa data dummy, seperti akun mahasiswa, akun psikolog, serta berbagai data pendukung lainnya yang telah disiapkan untuk merepresentasikan kondisi pengujian. Data tersebut digunakan untuk mendukung pelaksanaan berbagai skenario.

5. *Test Execution*

Pada tahap ini, pengujian dilakukan sesuai dengan kasus yang sudah dirancang [35]. *Jest* mengeksekusi semua *test case* melalui *Command Line Interface* (CLI). Hasil dari setiap eksekusi direkam secara otomatis oleh *Jest*, yang mencakup:

- a) Status pengujian, menunjukkan apakah pengujian berhasil (*pass*) atau gagal (*fail*).
- b) *Error log* menunjukkan detail *error*, seperti validasi gagal.

6. *Test Closure*

Tahap ini melibatkan evaluasi hasil pengujian untuk mengetahui sejauh mana sistem telah memenuhi kebutuhan pengguna. Laporan hasil pengujian disusun untuk mendokumentasikan hasil uji, dan temuan bug.

3.1.3 Fase Akhir

Pada tahap akhir, dilakukan proses analisis untuk mengevaluasi hasil pengujian otomatis. Analisis dilakukan dengan menghitung persentase keberhasilan *test case*, yaitu jumlah *test case* yang lulus dibandingkan dengan total *test case* yang dieksekusi. Hasil analisis ini kemudian dirangkum dalam bentuk kesimpulan yang mencakup:

- a) Penjabaran hasil pengujian otomatis menggunakan *Jest* pada sistem SiKosa berdasarkan hasil eksekusi *test case*.
- b) Evaluasi tingkat keberhasilan pengujian sistem berdasarkan metrik hasil pengujian (*test success rate*), bukan berdasarkan subjektivitas, melainkan berdasarkan hasil kuantitatif pengujian.

105

BAB IV

HASIL DAN PEMBAHASAN

17

78

1

Bab ini menyajikan hasil pengujian otomatis pada sistem Konseling Mahasiswa (SiKosa) menggunakan *Jest*, yang dilaksanakan sesuai dengan kerangka penelitian yang telah ditetapkan pada Bab III. Pembahasan mencakup seluruh tahapan mulai dari fase awal hingga fase akhir, termasuk tahapan *Software Testing Life Cycle* (STLC) pada fase pengujian.

4.1 Fase Awal

4.1.1 Identifikasi Kebutuhan Pengujian

Pada fase awal, dilakukan identifikasi terhadap fitur-fitur utama sistem SiKosa yang menjadi fokus pengujian. Berdasarkan hasil analisis menyeluruh terhadap alur kerja sistem, terdapat tujuh kelompok fitur yang diidentifikasi sebagai komponen krusial yang memerlukan pengujian, yaitu:

7

1

1. Autentikasi mencakup proses *registrasi* akun pengguna baru, *login* ke dalam sistem, dan *logout*. Fitur ini menjadi fondasi sistem karena seluruh akses ke fitur lain bergantung pada keberhasilan autentikasi.
2. Manajemen Profil Mahasiswa meliputi kemampuan mahasiswa untuk memperbarui data profil seperti nama lengkap, dan foto profil.
3. Manajemen Profil Psikolog meliputi kemampuan psikolog untuk memperbarui data profil seperti nama lengkap, deskripsi, spesialisasi, latar belakang pendidikan, dan foto profil.

4. **Konsultasi** mencakup pengajuan sesi **konsultasi** oleh mahasiswa, **serta** penerimaan atau **penolakan** sesi oleh psikolog. Ini merupakan fitur inti dari sistem SiKosa.
5. *Chat* fitur percakapan daring antara mahasiswa dan psikolog dalam sesi konsultasi yang telah diterima, termasuk pengiriman pesan dan riwayat percakapan.
6. *Chatbot* fitur asisten berbasis kecerdasan buatan yang melayani pertanyaan awal mahasiswa sebelum atau di luar sesi konsultasi formal.
7. Manajemen data *User* dan Artikel pengelolaan data pengguna (tambah, ubah, hapus) oleh administrator, serta pengelolaan artikel oleh psikolog dan admin (buat, ubah, hapus, lihat).

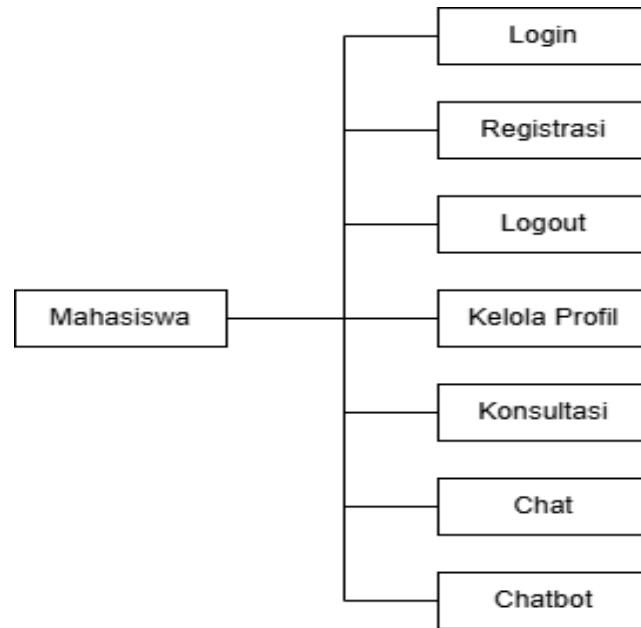
Identifikasi ini menjadi dasar penetapan **ruang lingkup** pengujian yang akan dilaksanakan pada fase pengujian. Seluruh fitur di atas diuji **untuk memastikan** sistem berjalan sesuai kebutuhan pengguna **secara konsisten**.

4.2 Fase Pengujian

Fase pengujian dilaksanakan menggunakan pendekatan *Software Testing Life Cycle* (STLC) yang mencakup enam tahapan sistematis, mulai dari analisis kebutuhan hingga evaluasi hasil pengujian (*test closure*).

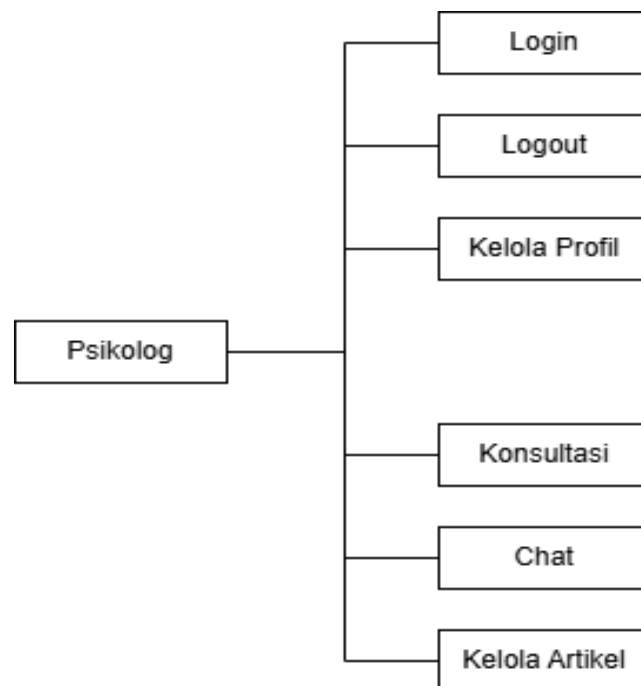
4.2.1 Requirement Analysis

Pada tahap ini dilakukan analisis terhadap seluruh **kebutuhan** sistem SiKosa untuk memahami fitur, alur kerja, dan spesifikasi yang harus dipenuhi, SiKosa memiliki tiga peran **pengguna dengan hak akses yang berbeda**.



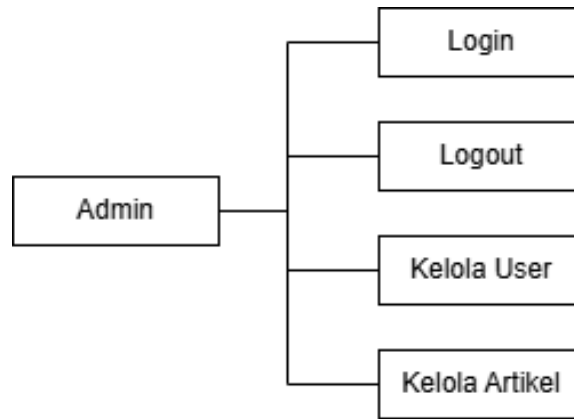
Gambar 4. 1 Fitur *role* mahasiswa

Gambar 4.1 menunjukkan fitur-fitur yang dapat diakses oleh pengguna dengan peran mahasiswa. Fitur tersebut terdiri atas tujuh fungsi utama, yaitu *login*, *registrasi*, *logout*, pengelolaan profil, konsultasi, *chat*, dan *chatbot*.



Gambar 4. 2 Fitur *role* psikolog

Gambar 4.2 menunjukkan fitur-fitur yang dapat diakses oleh pengguna dengan peran psikolog. Fitur tersebut terdiri atas enam fungsi utama, yaitu *login*, *logout*, kelola profil, konsultasi, *chat*, dan kelola artikel.



Gambar 4. 3 Fitur *role* admin

Gambar 4.3 menunjukkan fitur-fitur yang dapat diakses oleh pengguna dengan peran admin. Fitur tersebut terdiri atas empat fungsi utama, yaitu *login*, *logout*, kelola *user* dan kelola artikel.

Berdasarkan hasil analisis kebutuhan pengujian, ditetapkan dua pendekatan pengujian yang akan diterapkan dalam penelitian ini, yaitu:

1. *Unit Testing* digunakan untuk menguji logika pada *service* layer secara terpisah. Dependensi eksternal digantikan dengan *mock* menggunakan *Jest* agar pengujian lebih konsisten dan tidak bergantung pada *database*.
2. *Integration Testing* digunakan untuk menguji integrasi antar komponen sistem melalui API *endpoint* dan *database* MongoDB menggunakan HTTP *request*. Pengujian ini memastikan bahwa setiap komponen sistem dapat bekerja bersama dengan baik.

4.2.2 Test Planning

Pada tahap ini ditetapkan strategi pengujian secara menyeluruh, mencakup *framework*, pendekatan, dan cakupan pengujian yang akan digunakan.

a) Framework dan Tools Pengujian

Infrastruktur dasar dan *tools* pendukung yang memfasilitasi pembuatan dan eksekusi pengujian otomatis.

Tabel 4. 1 Framework dan Tools Pengujian

Tools	Versi	Fungsi
Jest	v30.3.0	Framework pengujian utama
supertest	v7.2.2	HTTP assertion untuk integration test
Jest-html-reporters	v3.1.7	Laporan HTML interaktif
Jest-stare	v2.5.3	Dashboard laporan hasil test
Mongoose	v8.9.2	ODM untuk MongoDB/Database

Berdasarkan Tabel 4.1 di atas, proses pengujian sistem dilakukan menggunakan *Jest* sebagai *framework* utama dan *Supertest* untuk pengujian endpoint API. Hasil pengujian didokumentasikan melalui laporan HTML, dashboard visual menggunakan *Jest-html-reporters* dan *Jest-stare*. *Mongoose* digunakan sebagai penghubung antara aplikasi dan database.

b) Strategi Pengujian

Rencana terstruktur yang menentukan pendekatan, ruang lingkup, dan langkah-langkah yang akan diambil untuk menguji suatu perangkat lunak.

1. Tipe pengujian: *Unit Testing* dan *Integration Testing*.
2. *Test runner*: *Jest*
3. *Coverage*: *Statement*, *Branch*, *Function*, dan *Line Coverage*.

c) Cakupan Pengujian per Modul

Daftar penjelasan mengenai apa saja yang harus dilakukan pengujian pada modul yang ada pada SiKosa.

Tabel 4. 2 Rencana Cakupan Pengujian

<i>Modul</i>	<i>Sub Modul</i>	<i>Unit Testing</i>	<i>Integration Testing</i>
Autentikasi	Login (Mahasiswa, Psikolog, Admin)	Ya	Ya
	Registrasi (Mahasiswa)	Ya	Ya
	Logout (Mahasiswa, Psikolog, Admin)	Ya	Ya
Manajemen Profil	Update Profil Mahasiswa	Ya	Ya
	Update Profil Psikolog	Ya	Ya
Konsultasi	Ajukan Konsultasi (Mahasiswa)	Ya	Ya
	Riwayat & Detail Konsultasi (Mahasiswa)	Ya	Ya
	Update Status Konsultasi (Psikolog)	Ya	Ya
	Notifikasi Konsultasi (Psikolog)		
Chat	Daftar Ruang Chat	Ya	Ya
	Kirim & Dapatkan Pesan	Ya	Ya
	Tutup Ruang Chat	Ya	Ya
Chatbot	Kirim Chat ke AI	Ya	Ya
Manajemen Data User	Lihat Semua User / By ID	Ya	Ya
	Buat User Baru	Ya	Ya
	Update User	Ya	Ya
	Hapus User	Ya	Ya
Manajemen Artikel	Buat Artikel	Ya	Ya
	Lihat Artikel	Ya	Ya
	Update Artikel	Ya	Ya
	Hapus Artikel	Ya	Ya

Berdasarkan Tabel 4.2, pengujian diterapkan pada seluruh modul utama sistem SiKosa menggunakan *Unit Testing* dan *Integration Testing*. Modul tersebut

mencakup autentikasi, manajemen profil, konsultasi, chat, chatbot, manajemen data pengguna, dan manajemen artikel.

d) Target Cakupan Pengujian (*Test Coverage*)

Cakupan pengujian (*test coverage*) merupakan metrik yang mengukur persentase kode program yang dieksekusi oleh *test case*, meliputi *statement*, *branch*, *function*, dan *line coverage*. *Test coverage* dipakai sebagai indikator kecukupan (*adequacy*) sekaligus ukuran efektivitas *test suite* [36]. Semakin tinggi cakupan, semakin besar bagian kode yang terverifikasi.

Tabel 4. 3 Jenis Coverage

Jenis Coverage	Deskripsi
<i>Statement Coverage</i>	Mengukur persentase pernyataan (<i>statement</i>) dalam kode program yang berhasil dieksekusi selama proses pengujian.
<i>Branch Coverage</i>	Mengukur persentase percabangan atau jalur keputusan yang telah dieksekusi, baik kondisi bernilai benar maupun salah.
<i>Function Coverage</i>	Mengukur persentase fungsi atau metode dalam program yang berhasil dipanggil dan dieksekusi selama pengujian.
<i>Line Coverage</i>	Mengukur persentase baris kode yang berhasil dieksekusi ketika pengujian dijalankan. Metrik ini menunjukkan seberapa banyak baris program yang telah tercakup oleh pengujian.

Tabel 4.3 menjelaskan jenis-jenis *code coverage* yang digunakan untuk mengevaluasi tingkat cakupan pengujian terhadap kode program. Metrik yang digunakan meliputi *Statement Coverage*, *Branch Coverage*, *Function Coverage*, dan *Line Coverage*.

Tabel 4. 4 Kategori Test Coverage

Tingkat Cakupan	Kategori Kememadaian
< 60%	Belum memadai
60% – 74%	Cukup
75% – 79%	Baik
≥ 80%	Memadai (acuan praktis penelitian)
≥ 90%	Sangat baik

Tabel 4.4 menunjukkan kategori tingkat kecukupan *test coverage* yang digunakan dalam penelitian ini. Cakupan pengujian di bawah 60% dikategorikan belum memadai, sedangkan cakupan 80% atau lebih dianggap telah memenuhi standar pengujian yang memadai.

Tabel 4. 5 Test Coverage SiKosa

Jenis <i>Test Coverage</i>	Tingkat Cakupan
<i>Statement</i>	$\geq 80\%$
<i>Branch</i>	$\geq 80\%$
<i>Function</i>	$\geq 80\%$
<i>Line Coverage</i>	$\geq 80\%$

Tabel 4.5 menunjukkan target *test coverage* minimum yang ditetapkan dalam penelitian ini, yaitu sebesar 80% untuk seluruh metrik pengujian. Target ini digunakan sebagai indikator bahwa pengujian telah mencakup sebagian besar kode program sehingga dapat memberikan tingkat kepercayaan yang memadai terhadap kualitas dan keandalan sistem yang diuji.

e) Target Keterlacakan Kebutuhan (*Requirement Traceability Matrix*)

Requirement Traceability Matrix (RTM) adalah matriks yang memetakan setiap kebutuhan sistem ke *test case* yang memverifikasinya. RTM berfungsi memastikan tidak ada kebutuhan yang luput dari pengujian [37].

RTM dipandang lengkap apabila seluruh kebutuhan (100%) telah tertelusuri ke minimal satu *test case*, serta tidak terdapat kebutuhan yang belum diuji (*orphan requirement*) maupun *test case* yang tidak terkait dengan kebutuhan (*orphan test case*).

Tabel 4. 6 Parameter Kelengkapan RTM

Parameter Evaluasi	Target Indikator	Penjelasan Operasional
Cakupan Keterlacakan (<i>Traceability Coverage</i>)	100%	Seluruh kebutuhan sistem telah berhasil ditelusuri dan dipetakan ke minimal satu <i>test case</i> .
Keberadaan <i>Orphan Requirement</i>	Nihil (0 Kasus)	Tidak terdapat kebutuhan sistem yang terlepas dari skenario pengujian (seluruh kebutuhan dipastikan akan diuji).
Keberadaan <i>Orphan Test Case</i>	Nihil (0 Kasus)	Tidak terdapat <i>test case</i> yang dibuat tanpa rujukan atau relasi yang valid terhadap kebutuhan sistem.

Tabel 4.6 menguraikan parameter kelengkapan *Requirement Traceability Matrix* (RTM) yang mensyaratkan 100% cakupan keterlacakan, dengan memastikan tidak adanya *orphan requirement* maupun *orphan test case*.

4.2.3 Test Case Development

Test case dikembangkan berdasarkan fitur-fitur yang telah diidentifikasi pada fase awal, penyusunan dilakukan secara bertahap dimulai dari peta fitur (*main map*), kemudian skenario pengujian (*test scenario*) sebagai dasar penyusunan *test case*, dan penyusunan *test case*.

A. Pengujian Role Mahasiswa

1. Main Map Fitur Mahasiswa

Berikut merupakan seluruh fitur yang dapat diakses dan dimanfaatkan oleh pengguna dengan peran sebagai mahasiswa pada sistem SiKosa. Fitur-fitur tersebut dirancang untuk mendukung berbagai kebutuhan mahasiswa dalam memanfaatkan layanan konsultasi secara daring, mulai dari mengakses informasi hingga melakukan proses konsultasi dengan lebih mudah, efektif, dan sesuai dengan fungsionalitas yang tersedia pada sistem.

Tabel 4. 7 Main Map Fitur Mahasiswa

No	Fitur	Deskripsi
1	<i>Login</i>	Masuk ke sistem menggunakan akun yang telah terdaftar
2	<i>Registrasi</i>	Mendaftarkan akun baru sebagai mahasiswa
3	<i>Logout</i>	Keluar dari sistem
4	Kelola Profil	Memperbarui nama lengkap, NIM, dan foto profil
5	Konsultasi	Memilih psikolog dan mengajukan sesi konsultasi
6	<i>Chat</i>	Mengirim dan membaca pesan dalam sesi konsultasi aktif
7	<i>Chatbot</i>	Berinteraksi dengan asisten AI untuk pertanyaan awal

Tabel 4.7 menyajikan daftar fitur yang dapat diakses oleh pengguna dengan peran sebagai mahasiswa pada sistem SiKosa. Fitur-fitur tersebut mencakup proses *registrasi*, *login*, pengelolaan profil, konsultasi, layanan *chat*, *chatbot*, hingga proses *logout*. Seluruh fitur tersebut merupakan ruang lingkup utama dalam penelitian ini dan akan menjadi objek verifikasi pada tahap pengujian untuk memastikan setiap fungsionalitas berjalan sesuai dengan kebutuhan serta spesifikasi sistem yang telah ditetapkan.

a) Fitur *Login* (Mahasiswa)

Fitur *login* mahasiswa merupakan mekanisme autentikasi yang dirancang untuk memverifikasi identitas pengguna sebelum memperoleh akses ke dalam sistem SiKosa. Proses autentikasi dilakukan dengan memasukkan kredensial yang telah terdaftar, yaitu alamat email dan kata sandi (*password*). Apabila data yang dimasukkan valid, mahasiswa akan berhasil masuk ke dalam sistem dan dapat mengakses seluruh fitur yang tersedia sesuai dengan hak akses yang dimilikinya.



Sign In

If you don't have an account

You can [Register here!](#)

Email

✉ Enter your Email

Password

🔒 Enter your password



Sign In

52

Gambar 4. 4 Halaman *login* mahasiswa

Gambar 4.4 halaman *login* Mahasiswa menampilkan antarmuka yang digunakan oleh mahasiswa untuk masuk ke dalam sistem SiKosa. Halaman ini berfungsi sebagai gerbang utama untuk menjaga keamanan serta membatasi akses pengguna sesuai dengan hak akses yang dimiliki.

10

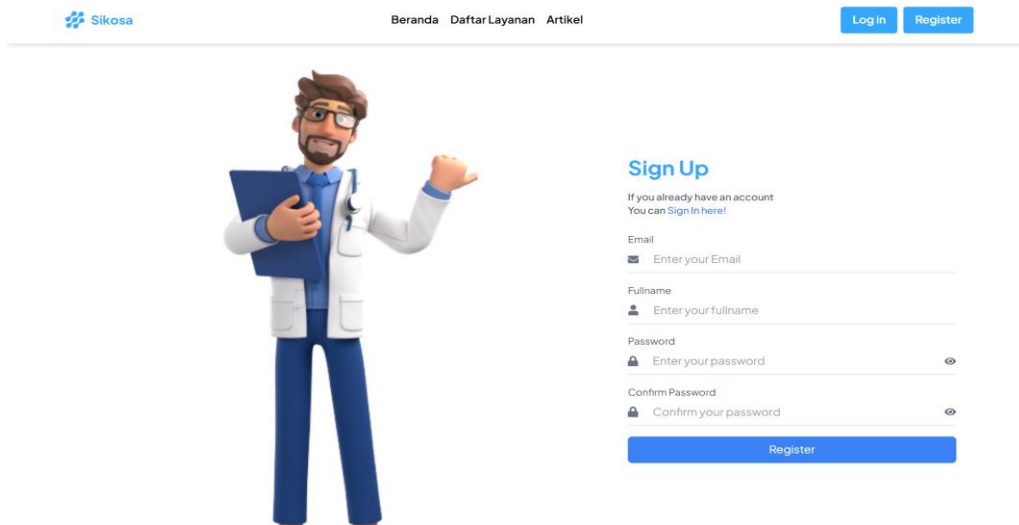
b) Fitur *Registrasi*

68

Fitur *registrasi* memungkinkan mahasiswa membuat akun baru pada sistem SiKosa dengan mengisi data berupa email, nama lengkap (fullname), *password*, dan konfirmasi *password* (confirm *password*). Data yang dimasukkan akan divalidasi terlebih dahulu sebelum akun berhasil didaftarkan ke dalam sistem.

186

93

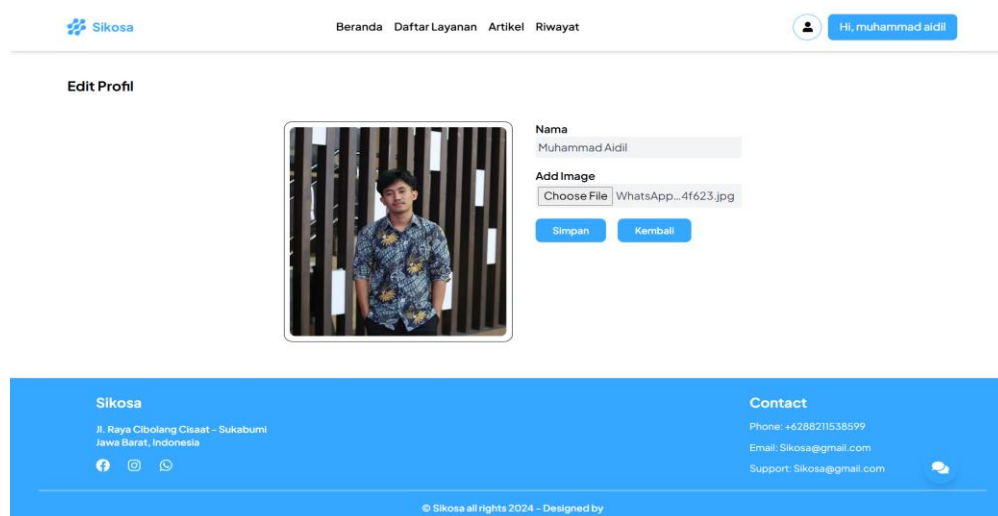


Gambar 4. 5 Halaman *registrasi* mahasiswa

Gambar 4.5 halaman *registrasi* Mahasiswa menampilkan antarmuka pendaftaran akun yang digunakan oleh mahasiswa untuk membuat akun baru pada sistem SiKosa. Setelah data berhasil divalidasi dan disimpan, akun mahasiswa akan terdaftar sehingga dapat digunakan untuk mengakses sistem.

c) Fitur Kelola Profil

Fitur kelola profil memungkinkan mahasiswa untuk melihat dan memperbarui informasi profil yang tersimpan pada sistem SiKosa.

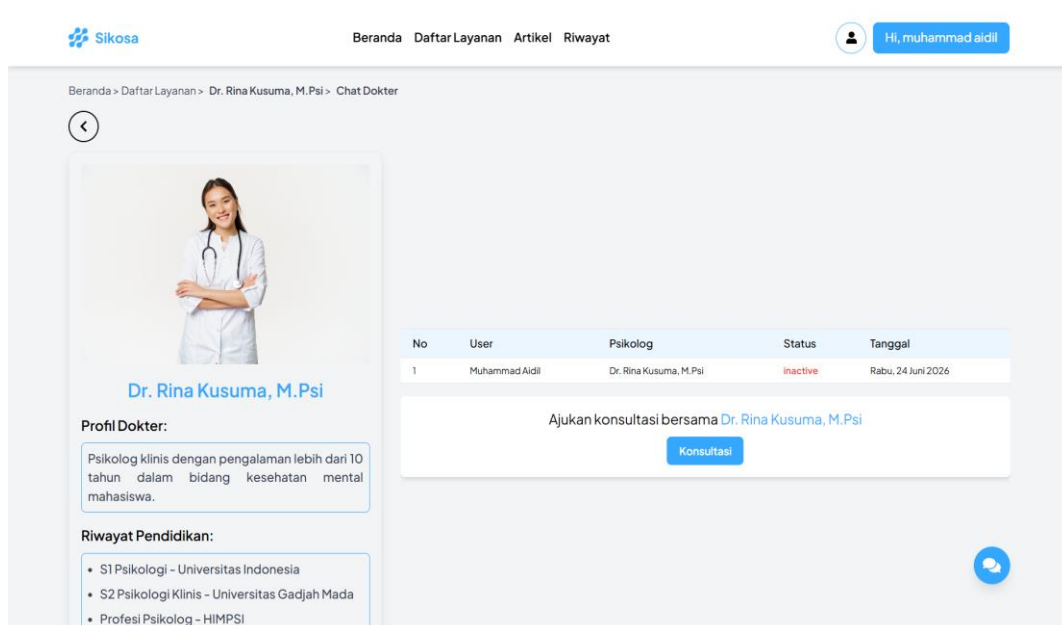


Gambar 4. 6 Halaman kelola profil

Gambar 4.6 halaman kelola profil menampilkan antarmuka yang digunakan mahasiswa untuk melihat dan memperbarui informasi profil pribadi pada sistem SiKosa. Melalui halaman ini, mahasiswa dapat mengubah data seperti nama lengkap dan foto profil.

d) Fitur Konsultasi

Fitur Konsultasi merupakan fitur utama pada sistem SiKosa yang memungkinkan mahasiswa mengajukan permohonan sesi konsultasi kepada psikolog.

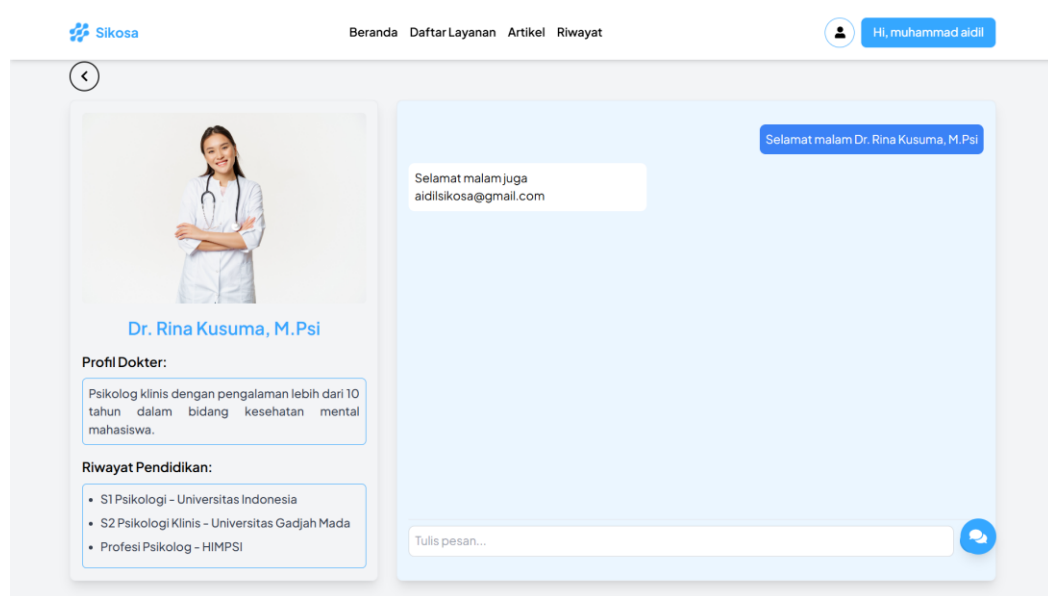


Gambar 4. 7 Halaman konsultasi

Gambar 4.7 halaman konsultasi menampilkan antarmuka yang digunakan mahasiswa untuk mengelola konsultasi pada sistem SiKosa. Pada halaman ini, mahasiswa dapat melihat daftar psikolog yang tersedia, mengajukan sesi konsultasi, serta memantau status konsultasi yang telah diajukan.

e) Fitur *Chat*

Fitur *chat* memungkinkan mahasiswa dan psikolog berkomunikasi secara langsung melalui pesan teks dalam sesi konsultasi yang telah disetujui. Fitur ini digunakan sebagai media pelaksanaan konsultasi daring, sehingga kedua pihak dapat bertukar informasi, memberikan tanggapan, serta mendiskusikan permasalahan yang dihadapi mahasiswa secara *real-time*.

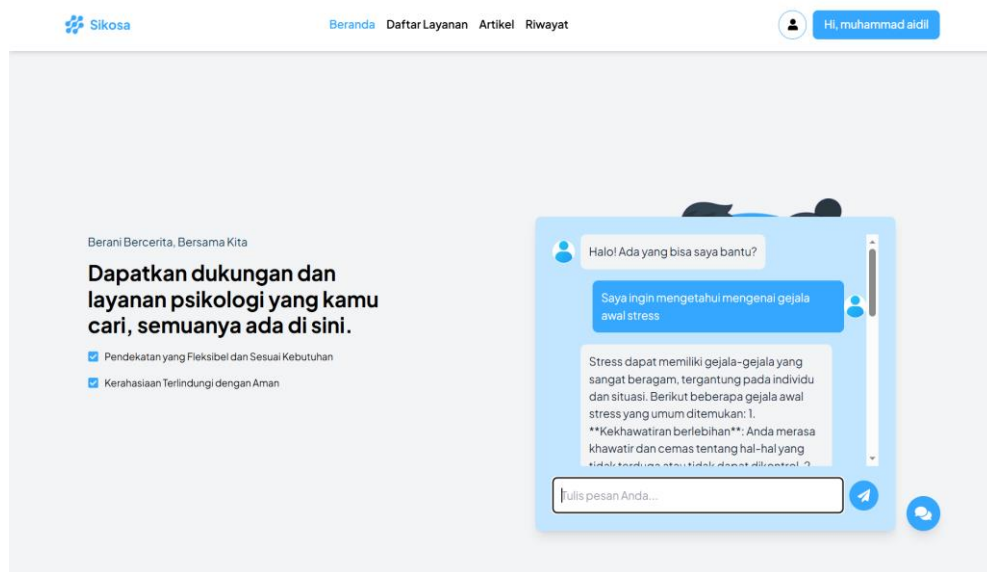


Gambar 4. 8 Halaman *chat*

Gambar 4.8 halaman *chat* menampilkan antarmuka percakapan antara mahasiswa dan psikolog pada sistem SiKosa. Pada halaman ini, pengguna dapat melihat riwayat pesan yang telah dikirim, membaca pesan baru, serta mengirimkan pesan melalui kolom *input* yang tersedia.

f) Fitur *Chatbot*

Fitur *Chatbot* merupakan layanan asisten virtual berbasis kecerdasan buatan yang dirancang untuk membantu mahasiswa memperoleh informasi dan dukungan awal terkait permasalahan yang dihadapi.



Gambar 4. 9 Halaman *chatbot*

Gambar 4.9 halaman *chatbot* menampilkan antarmuka percakapan antara mahasiswa dan *chatbot* pada sistem SiKosa. Pada halaman ini, mahasiswa dapat mengetik pertanyaan atau pesan melalui kolom *input* yang tersedia, kemudian *chatbot* akan memberikan respons secara otomatis dalam bentuk percakapan.

g) Fitur *Logout*

Fitur *logout* digunakan untuk mengakhiri sesi pengguna yang sedang aktif di dalam sistem. Ketika pengguna memilih tombol *logout*, sistem akan menghapus sesi autentikasi atau *token*



Gambar 4. 10 Fitur *logout*

Gambar 4.10 menampilkan proses *logout* yang dilakukan oleh pengguna. Setelah tombol *logout* dipilih, sistem mengakhiri sesi pengguna dan mengarahkan kembali ke halaman *login* sehingga pengguna harus melakukan autentikasi ulang untuk dapat mengakses sistem kembali.

2. Test Scenario Mahasiswa

Test scenario disusun berdasarkan setiap fitur yang ada dalam sistem dengan tujuan sebagai dasar dalam penyusunan *test case* yang lebih rinci, terstruktur, dan mudah dieksekusi.

Tabel 4. 8 Test Scenario Mahasiswa

ID Skenario	Skenario Pengujian
TS-MHS-01	Autentikasi: Proses pendaftaran akun baru, masuk ke sistem, dan keluar dari sistem
TS-MHS-02	Kelola Profil: Mengubah data diri seperti nama, NIM, dan foto profil
TS-MHS-03	Konsultasi: Mengajukan konsultasi ke psikolog, riwayat dan detail konsultasi
TS-MHS-04	Chat: Berkomunikasi melalui pesan teks dengan psikolog
TS-MHS-05	Chatbot AI: Berinteraksi dengan asisten virtual berbasis AI untuk mendapatkan informasi atau bantuan awal

Tabel 4.8 menunjukkan daftar *test scenario* yang disusun untuk menguji seluruh fitur yang dapat diakses oleh mahasiswa pada sistem SiKosa. Skenario pengujian tersebut mencakup proses *registrasi* akun, *login* dan *logout*, pengelolaan profil, pengajuan dan pengelolaan konsultasi, komunikasi melalui fitur *chat*, serta interaksi dengan *chatbot*.

3. Test Case (Mahasiswa)

Berikut disajikan kumpulan *test case* yang ditujukan untuk peran mahasiswa, yang digunakan sebagai acuan dalam melakukan pengujian otomatis.

Tabel 4. 9 Test Case Login Mahasiswa

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-LOGIN-01	Login dengan email kosong. Kondisi awal: Belum ada sesi masuk sebelumnya.	1. Siapkan data <i>login</i> dengan mengosongkan kolom email dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400 - data tidak valid). Sistem menampilkan pesan bahwa email wajib diisi.
	TC-LOGIN-02	Login dengan kata sandi kosong. Kondisi awal: Belum ada sesi masuk sebelumnya.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengosongkan kolom kata sandi. 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400 - data tidak valid). Sistem menampilkan pesan bahwa kata sandi wajib diisi.
	TC-LOGIN-03	Login dengan format email salah. Kondisi awal: Belum ada sesi masuk sebelumnya.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "invalid" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400 - data tidak valid). Sistem menampilkan pesan format email tidak sesuai.
	TC-LOGIN-04	Login dengan kata sandi terlalu pendek (3 karakter). Kondisi awal: Belum ada sesi masuk sebelumnya.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan "123". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400 - data tidak valid). Sistem menampilkan pesan kata sandi minimal 8 karakter.
	TC-LOGIN-05	Login dengan email terlalu panjang (269 karakter). Kondisi awal: Belum ada sesi masuk sebelumnya.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan teks 269 karakter dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400 - data tidak valid). Email melebihi batas maksimal.
	TC-LOGIN-06	Login dengan email yang belum terdaftar. Kondisi awal: Email "notfound@mail.com" tidak ada di dalam sistem.	1. Siapkan data <i>login</i> dengan email dengan "notfound@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (404 - tidak ditemukan). Akun dengan email tersebut tidak ditemukan.

Tabel 4.9 Test Case Login Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-LOGIN-07	Login dengan kata sandi salah. Kondisi awal: Akun mahasiswa dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" sudah terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan "Wrong123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (401 - tidak sah). Kombinasi email dan kata sandi tidak sesuai.
	TC-LOGIN-08	Login dengan email yang mengandung spasi di awal dan akhir. Kondisi awal: Akun mahasiswa dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" sudah terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan " mahasiswa@mail.com " dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan berhasil (200). Sistem berhasil memproses email dengan spasi dan masuk sukses.
	TC-LOGIN-09	Login dengan kata sandi yang mengandung spasi. Kondisi awal: Akun mahasiswa dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" sudah terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan " Valid123! ". 2. Kirim data <i>login</i> ke sistem.	Permintaan berhasil (200). Sistem menerima kata sandi dengan spasi.
	TC-LOGIN-10	Login sebagai mahasiswa berhasil. Kondisi awal: Akun mahasiswa dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" sudah terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan berhasil (200). Sistem mengembalikan data user dengan peran mahasiswa, beserta kode akses dan kode penyegar.
	TC-INT-LOGIN-001	Login sebagai mahasiswa - session tercatat. Kondisi awal: Akun mahasiswa terdaftar, belum ada sesi.	1. Siapkan data <i>login</i> email dengan "mahasiswa@mail.com" dan kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan berhasil (200). Basis data berisi 1 sesi milik mahasiswa.

Tabel 4.9 Test Case Login Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-INT-LOGIN-002	<i>Login ganda - dua session aktif.</i> Kondisi awal: Akun mahasiswa terdaftar, belum ada sesi.	1. Kirim data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" (pertama). 2. Kirim data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" (kedua). 3. Periksa data di basis data sesi.	Kedua berhasil. Basis data berisi 2 sesi.
	TC-INT-LOGIN-003	Body <i>login</i> kosong. Kondisi awal: Tidak ada data khusus.	1. Kosongkan seluruh body <i>login</i> (tanpa mengisi kolom email dan kata sandi). 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Sistem mengembalikan daftar kesalahan.
	TC-INT-LOGIN-004	Field email tidak dikirim. Kondisi awal: Tidak ada data khusus.	1. Siapkan data <i>login</i> tanpa menyertakan field email dan dengan kata sandi "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Field email wajib diisi.
	TC-INT-LOGIN-005	Field kata sandi tidak dikirim. Kondisi awal: Tidak ada data khusus.	1. Siapkan data <i>login</i> dengan email "mahasiswa@mail.com" dan tanpa menyertakan field kata sandi. 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Field kata sandi wajib diisi.
	TC-INT-LOGIN-006	<i>Login</i> - respons berisi kode akses. Kondisi awal: Akun mahasiswa terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Respons berisi "accessToken" bertipe teks.
	TC-INT-LOGIN-007	<i>Login</i> - respons berisi kode penyegar. Kondisi awal: Akun mahasiswa terdaftar.	1. Siapkan data <i>login</i> dengan email "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Respons berisi "refreshToken".

Tabel 4.9 Test Case Login Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-INT-LOGIN-008	Cookie kode akses tersimpan. Kondisi awal: Akun mahasiswa terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". Kirim data <i>login</i> ke sistem. 2. Periksa header respons Set-Cookie.	Cookie mengandung "accessToken".
	TC-INT-LOGIN-009	Cookie kode penyegar tersimpan. Kondisi awal: Akun mahasiswa terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "mahasiswa@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". Kirim data	Cookie mengandung "refreshToken".
	TC-INT-LOGIN-010	<i>Login</i> user yang belum verifikasi. Kondisi awal: Akun mahasiswa dengan "verified": false terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan email akun mahasiswa yang belum diverifikasi dan mengisi kolom kata sandi dengan kata sandi akun tersebut. Kirim data <i>login</i> ke sistem. 2. Periksa field "verified".	Permintaan berhasil (200). "verified" bernilai "false".
	TC-INT-LOGIN-011	<i>Login</i> - akun dinonaktifkan. Kondisi awal: Akun mahasiswa dinonaktifkan admin.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan email akun mahasiswa yang dinonaktifkan dan mengisi kolom kata sandi dengan kata sandi akun tersebut. 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (403). Akun dinonaktifkan.
	TC-INT-LOGIN-012	<i>Login</i> - akun dihapus. Kondisi awal: Akun mahasiswa sudah dihapus dari basis data.	1. Siapkan data <i>login</i> dengan email akun mahasiswa yang sudah dihapus dan mengisi kolom kata sandi dengan kata sandi akun tersebut. 2. Kirim data <i>login</i> ke sistem	Permintaan ditolak (404). Akun tidak ditemukan.

Tabel 4.9 Test Case Login Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-INT-LOGIN-013	Login - email huruf besar. Kondisi awal: Akun mahasiswa dengan email campuran huruf besar/kecil.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "Mahasiswa@Mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan berhasil (200). Email dikonversi ke huruf kecil.
	TC-INT-LOGIN-014	Login - login setelah register langsung. Kondisi awal: Akun baru saja dibuat.	1. Siapkan data <i>login</i> menggunakan akun yang baru dibuat 2. Kirim data <i>login</i> ke sistem.	Permintaan berhasil (200).
	TC-INT-LOGIN-015	Login berkali-kali tidak menyebabkan error. Kondisi awal: Akun mahasiswa terdaftar.	1. Siapkan data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Kirim data <i>login</i> ke sistem sebanyak 5 kali berturut-turut.	Semua berhasil (200).
	TC-INT-LOGIN-016	Login - token akses memiliki masa berlaku. Kondisi awal: Akun mahasiswa terdaftar.	1. Kirim data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Periksa token akses.	Kode akses memiliki masa berlaku.
	TC-INT-LOGIN-017	Login - data user sesuai basis data. Kondisi awal: Akun mahasiswa terdaftar.	1. Kirim data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Periksa data user pada respons.	Data user sesuai dengan basis data.
	TC-INT-LOGIN-018	Login - tidak mengubah data user. Kondisi awal: Akun mahasiswa terdaftar.	1. Kirim data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Periksa data user sebelum dan sesudah.	Data user tidak berubah.
	TC-INT-LOGIN-019	Login - status sukses. Kondisi awal: Akun mahasiswa terdaftar.	1. Kirim data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Periksa status respons.	Status "success".

Tabel 4.9 Test Case Login Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-INT-LOGIN-020	<i>Login</i> - header tipe konten JSON. Kondisi awal: Akun mahasiswa terdaftar.	1. Kirim data <i>login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Periksa header respons Content-Type.	"application/json".
	TC-INT-LOGIN-021	<i>Login</i> dengan data palsu - tidak membuat sesi. Kondisi awal: Tidak ada akun dengan data tersebut.	1. Siapkan data <i>login</i> dengan email "fake@mail.com" dan kata sandi "Fake123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (401). Tidak ada sesi baru.
	TC-INT-LOGIN-022	<i>Login</i> - cookie dibersihkan saat <i>logout</i> lalu <i>login</i>. Kondisi awal: Akun mahasiswa terdaftar.	1. <i>Login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. <i>Logout</i>. 3. <i>Login</i> lagi dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!".	<i>Login</i> berhasil (200). Cookie baru dibuat.
	TC-INT-LOGIN-023	<i>Login</i> - session setelah <i>logout</i>. Kondisi awal: Sesi sebelumnya dihapus.	1. <i>Login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" setelah <i>logout</i>. 2. Periksa data di basis data sesi.	Sesi baru tercatat.
	TC-INT-LOGIN-024	<i>Login</i> - kata sandi tepat 8 karakter. Kondisi awal: Akun dengan kata sandi tepat 8 karakter.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan email akun yang memiliki kata sandi 8 karakter dan mengisi kolom kata sandi dengan kata sandi 8 karakter. 2. Kirim data <i>login</i> ke sistem.	Permintaan berhasil (200).

Tabel 4.9 menyajikan *test case* proses *login* untuk peran mahasiswa yang mencakup skenario *login* berhasil serta validasi kesalahan *input* seperti kolom email

atau kata sandi yang dikosongkan dan penggunaan kata sandi yang salah, lengkap dengan kondisi awal, langkah pengujian, dan hasil yang diharapkan.

Tabel 4. 10 Test Case Registrasi Mahasiswa

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-AUTH-01	Daftar dengan data kosong. Kondisi awal: Basis data bersih.	1. Biarkan seluruh data pendaftaran kosong (kosongkan kolom email, kata sandi, konfirmasi kata sandi, nama, NIM, dan tidak memilih peran). 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400 - data tidak valid). Sistem menampilkan pesan kesalahan.
	TC-AUTH-02	Daftar dengan format email salah. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "invalid", kata sandi "Password1!", konfirmasi kata sandi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Format email tidak sesuai.
	TC-AUTH-03	Daftar dengan email kosong. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan mengosongkan kolom email, mengisi kata sandi "Password1!", konfirmasi kata sandi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Email wajib diisi.
	TC-AUTH-04	Daftar dengan konfirmasi kata sandi tidak cocok. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "wrong", peran "mahasiswa", NIM "12345678", nama "Aidil". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Kata sandi dan konfirmasi tidak sesuai.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-AUTH-05	Daftar dengan kata sandi kurang dari 6 karakter. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "123", konfirmasi kata sandi "123", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Kata sandi terlalu pendek.
	TC-AUTH-06	Daftar dengan peran tidak dikenal. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "user", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Peran tidak dikenali.
	TC-AUTH-07	Daftar tanpa peran - otomatis mahasiswa. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", tanpa menyertakan field peran, nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201). Peran default mahasiswa.
	TC-AUTH-08	Daftar mahasiswa dengan data lengkap. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", NIM "12345678", nama "Aidil". 2. Kirim data pendaftaran ke sistem. 3. Periksa data di basis data.	Permintaan berhasil (201). User baru tersimpan.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-AUTH-09	Daftar dengan peran admin ditolak. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "admin", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (403). Tidak diizinkan.
	TC-AUTH-10	Daftar dengan peran psikolog ditolak. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "psikolog", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (403). Tidak diizinkan.
	TC-AUTH-11	Daftar dengan email yang sudah terpakai. Kondisi awal: User dengan email "user@mail.com" sudah terdaftar.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (409). Email sudah terdaftar.
	TC-AUTH-12	Daftar dengan kata sandi tanpa huruf kapital. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "password1!", konfirmasi "password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Harus mengandung huruf kapital.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-AUTH-13	Daftar dengan kata sandi tanpa angka. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password!", konfirmasi "Password!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Harus mengandung angka.
	TC-AUTH-14	Daftar dengan kata sandi tanpa simbol. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1", konfirmasi "Password1", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Harus mengandung simbol.
	TC-AUTH-15	Daftar dengan kata sandi kurang dari 8 karakter (validasi sistem). Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Pass1!", konfirmasi "Pass1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Minimal 8 karakter.
	TC-AUTH-16	Daftar dengan kata sandi valid. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201).
	TC-AUTH-17	Daftar tanpa data profil. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!",	Permintaan ditolak (400). Profil wajib diisi.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01			konfirmasi "Password1!", peran "mahasiswa", dan tanpa menyertakan field profil. 2. Kirim data pendaftaran ke sistem.	
	TC-AUTH-18	Daftar dengan nama kosong. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa",	Permintaan ditolak (400). Nama wajib diisi.
	TC-AUTH-19	Daftar dengan nama mengandung angka. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "123", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Nama tidak boleh angka.
	TC-AUTH-20	Daftar dengan nama terlalu pendek (1 karakter). Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "A", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Minimal 2 karakter.
	TC-AUTH-21	Daftar dengan nama terlalu panjang (60 karakter). Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama 60 karakter, dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Melebihi batas.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-AUTH-22	Daftar dengan nama valid. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201).
	TC-AUTH-23	Daftar tanpa foto - tetap berhasil. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678", dan tanpa menyertakan field foto dalam profil. 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201). Foto opsional.
	TC-AUTH-24	Daftar dengan URL foto tidak valid. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678", dan URL foto "abc". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Format URL tidak sesuai.
	TC-AUTH-25	Daftar dengan URL foto bukan gambar. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM	Permintaan ditolak (400). Harus file gambar.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01			"12345678", dan URL foto "file.pdf". 2. Kirim data pendaftaran ke sistem.	
	TC-AUTH-26	Daftar dengan URL foto terlalu panjang. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678", dan URL foto 300+ karakter. 2. Kirim data ke sistem	Permintaan ditolak (400). Melebihi batas.
	TC-AUTH-27	Daftar dengan URL foto valid. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678", dan URL foto "http://image.com/a.jpg". 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201).
	TC-AUTH-28	Daftar dengan NIM valid (8 digit). Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201).
	TC-AUTH-29	Daftar dengan NIM bukan angka. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "ABC". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). NIM harus angka.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-AUTH-30	Daftar dengan NIM terlalu pendek (3 digit). Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "123". 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400). Panjang tidak sesuai.
	TC-INT-REG-001	Daftar dengan NIM terlalu panjang (20 digit). Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM 20 digit. 2. Kirim data pendaftaran ke sistem.	Permintaan ditolak (400).
	TC-INT-REG-002	Daftar dengan NIM kosong - berhasil. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan mengosongkan kolom NIM. 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201). NIM opsional.
	TC-INT-REG-003	Daftar dengan email huruf besar. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "TEST@MAIL.COM", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201). Email ke huruf kecil.
	TC-INT-REG-004	Daftar dengan nama hanya spasi.	1. Siapkan data pendaftaran dengan email "user@mail.com",	Permintaan ditolak (400). Tidak boleh

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01		Kondisi awal: Basis data bersih.	kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama " ", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem.	hanya spasi.
	TC-INT-REG-005	Daftar lengkap - user, sesi, kode verifikasi, cookie. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa data di basis data user, sesi, kode verifikasi. 4. Periksa data pada cookie.	Permintaan berhasil (201). Semua tercatat.
	TC-INT-REG-006	Daftar - kata sandi terenkripsi di basis data. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". Kirim data pendaftaran ke sistem. 2. Periksa data di basis data.	Kata sandi tersimpan tidak sama dengan aslinya.
	TC-INT-REG-007	Daftar - kode verifikasi dibuat. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa data di basis data kode verifikasi.	Kode verifikasi tercatat.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
	TC-INT-REG-008	Daftar - sesi langsung dibuat. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa data di basis data sesi.	Sesi aktif tercatat.
	TC-INT-REG-009	Daftar - cookie kode akses langsung tersimpan. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa data pada cookie.	Cookie mengandung "accessToken".
	TC-INT-REG-010	Daftar - cookie kode penyegar langsung tersimpan. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa data pada cookie.	Cookie mengandung "refreshToken".
	TC-INT-REG-011	Daftar dua user berbeda - keduanya berhasil. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran user A dengan email "user1@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM	Kedua user tersimpan.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
			"12345678". 2. Siapkan data pendaftaran user B dengan email "user2@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Budi", NIM "87654321". 3. Kirim data pendaftaran untuk kedua user ke sistem. 4. Periksa data di basis data.	
	TC-INT-REG-012	Daftar - respons berisi data user. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa respons.	Respons berisi data user baru.
	TC-INT-REG-013	Daftar - status respons 201. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa kode status.	201 (berhasil dibuat).
	TC-INT-REG-014	Daftar - header tipe konten JSON. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi	"application/json".

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01			"Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa Content-Type.	
	TC-INT-REG-015	Daftar dengan data sangat panjang. Kondisi awal: Basis data bersih.	1. Kirim data dengan email 300 karakter, kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan ditolak (400).
	TC-INT-REG-016	Daftar - field tidak dikenal diabaikan. Kondisi awal: Basis data bersih.	1. Kirim data dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678", dan field tambahan "hobi": "membaca".	Permintaan berhasil (201). Field aneh diabaikan.
	TC-INT-REG-017	Daftar tanpa NIM - berhasil. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan tanpa menyertakan NIM. 2. Kirim data pendaftaran ke sistem.	Permintaan berhasil (201).
	TC-INT-REG-018	Daftar user baru punya peran mahasiswa. Kondisi awal: Basis data bersih.	1. Kirim data email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Periksa data di basis data.	Peran mahasiswa tersimpan.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
	TC-INT-REG-019	Daftar - sesi baru punya kode akses valid. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Gunakan token akses untuk akses fitur.	Kode akses valid.
	TC-INT-REG-020	Daftar - email duplikat dalam waktu bersamaan. Kondisi awal: Basis data bersih.	1. Kirim dua permintaan pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678" hampir bersamaan.	Satu berhasil (201), satu ditolak (409).
	TC-INT-REG-021	Daftar - nama dengan spasi di awal. Kondisi awal: Basis data bersih.	1. Kirim data dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan berhasil (201).
	TC-INT-REG-022	Daftar - nama panjang 50 karakter. Kondisi awal: Basis data bersih.	1. Kirim data dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama 50 karakter, dan NIM "12345678".	Permintaan berhasil (201).
	TC-INT-REG-023	Daftar - respons tidak mengandung kata sandi. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!",	Respons tidak berisi kata sandi.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01			konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Periksa respons.	
	TC-INT-REG-024	Daftar dengan data kosong - semua field error. Kondisi awal: Basis data bersih.	1. Siapkan data dengan field dikosongkan (email "", kata sandi "", konfirmasi "", nama "", tanpa peran, tanpa NIM). 2. Kirim data ke sistem.	Permintaan ditolak (400). Banyak error.
	TC-INT-REG-025	Daftar - konfirmasi kata sandi beda kapital. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan ditolak (400). Tidak cocok.
	TC-INT-REG-026	Daftar - NIM dengan spasi. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan ditolak (400). NIM tidak boleh spasi.
	TC-INT-REG-027	Daftar - profil tanpa nama. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", dan profil kosong (tanpa field name dan nim).	Permintaan ditolak (400). Profil tidak lengkap.
	TC-INT-REG-028	Daftar dengan data minimum - berhasil. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi	Permintaan berhasil (201).

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01			"Password1!", konfirmasi "Password1!", peran "mahasiswa", dan nama "Aidil" (tanpa NIM dan foto).	
	TC-INT-REG-029	Daftar - NIM 8 digit angka valid. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "87654321".	Permintaan berhasil (201).
	TC-INT-REG-030	Daftar - password dengan karakter unik. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "P@ssw0rd!!", konfirmasi "P@ssw0rd!!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan berhasil (201).
	TC-INT-REG-031	Daftar - foto URL http dan https. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran pertamadengan "user1@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678", URL foto "http://image.com/a.jpg". 2. Siapkan data pendaftaran kedua dengan email "user2@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Budi", NIM "87654321", URL foto "https://image.com/a.jpg".	Keduanya berhasil (201).
	TC-INT-REG-032	Daftar - email dengan tanda plus. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan "user+test@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan berhasil (201).
	TC-INT-REG-033	Daftar - nama 2 karakter (minimal). Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Ab", dan NIM "12345678".	Permintaan berhasil (201).

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-INT-REG-034	Daftar - semua field string kosong. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan semua field string dikosongkan (email "", kata sandi "", konfirmasi "", peran "", nama "", NIM "").	Permintaan ditolak (400).
	TC-INT-REG-035	Daftar lalu login - sukses. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678". 2. Kirim data pendaftaran ke sistem. 3. Logout. 4. Login dengan email "user@mail.com" dan kata sandi "Password1!".	Login berhasil (200).
	TC-INT-REG-036	Daftar - email dengan subdomain. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.university.ac.id", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan berhasil (201).
	TC-INT-REG-037	Daftar - kata sandi tanpa simbol ditolak. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1", konfirmasi "Password1", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan ditolak (400).
	TC-INT-REG-038	Daftar - kata sandi dengan spasi di dalam. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Pass word1!", konfirmasi "Pass word1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Berhasil (201) atau ditolak (400).
	TC-INT-REG-039	Daftar - peran dikirim dengan spasi. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa ", nama "Aidil", dan NIM "12345678".	Ditolak (400) atau berhasil.

Tabel 4.10 Test Case Registrasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-INT-REG-040	Daftar - email dengan karakter khusus. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user#name@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "12345678".	Permintaan ditolak (400).
	TC-INT-REG-041	Daftar - NIM 0 (nol) dianggap valid. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", dan NIM "00000000".	Permintaan berhasil (201).
	TC-INT-REG-042	Daftar - field peran huruf besar. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "MAHASISWA", nama "Aidil", dan NIM "12345678".	Berhasil (201) atau ditolak (400).
	TC-INT-REG-043	Daftar - foto base64 ditolak. Kondisi awal: Basis data bersih.	1. Siapkan data pendaftaran dengan email "user@mail.com", kata sandi "Password1!", konfirmasi "Password1!", peran "mahasiswa", nama "Aidil", NIM "12345678", dan foto berupa teks "data:image/png;base64,abc".	Permintaan ditolak (400).

Tabel 4.10 menyajikan kumpulan *test case* untuk fitur *registrasi* pada pengguna dengan peran sebagai mahasiswa. Skenario pengujian yang disusun mencakup kondisi *registrasi* berhasil serta berbagai kondisi validasi terhadap kesalahan input, seperti email yang tidak diisi, email yang telah terdaftar, maupun peran pengguna yang tidak sesuai. Setiap *test case* dilengkapi dengan langkah-langkah pengujian, data masukan yang digunakan, serta hasil yang diharapkan sebagai acuan keberhasilan pengujian.

Tabel 4. 11 Test Case Logout Mahasiswa

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-LOGOUT-01	Keluar tanpa kode akses. Kondisi awal: User sudah masuk.	1. Kirim permintaan <i>logout</i> tanpa token akses (Authorization header) dan tanpa cookie.	Permintaan ditolak (401 - tidak sah).
	TC-LOGOUT-02	Keluar dengan kode akses tidak valid. Kondisi awal: User sudah masuk.	1. Kirim permintaan <i>logout</i> dengan token akses "invalid" pada Authorization header.	Permintaan ditolak (401). Kode tidak dikenal.
	TC-LOGOUT-03	Keluar dengan kode akses kadaluwarsa. Kondisi awal: Token sengaja dibuat kadaluwarsa.	1. Buat token akses yang sudah kadaluwarsa. 2. Kirim permintaan <i>logout</i> dengan token kadaluwarsa pada Authorization header.	Permintaan ditolak (401). Kode kadaluwarsa.
	TC-LOGOUT-04	Keluar melalui kode akses berhasil. Kondisi awal: Sesi aktif ada di basis data.	1. <i>Login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" untuk mendapat token akses. 2. Kirim permintaan <i>logout</i> dengan token akses pada Authorization header. 3. Periksa data di basis data.	Permintaan berhasil (200). Sesi terhapus.
	TC-LOGOUT-05	Keluar melalui cookie berhasil. Kondisi awal: Sesi aktif ada di basis data.	1. <i>Login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" untuk mendapat cookie. 2. Kirim permintaan <i>logout</i> dengan cookie (tanpa Authorization header).	Permintaan berhasil (200). Sesi terhapus.
	TC-INT-LOGOUT-001	Keluar ketika sesi sudah tidak ada. Kondisi awal: Sesi dihapus manual dari basis data.	1. <i>Login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Hapus sesi manual dari basis data. 3. Kirim permintaan <i>logout</i> dengan token akses.	Permintaan ditolak (401). Sesi tidak ditemukan.
	TC-INT-LOGOUT-002	Keluar menghapus sesi dari basis data. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Catat ID sesi. 3. <i>Logout</i> dengan token akses. 4. Cari sesi di basis data.	Sesi tidak ditemukan.
	TC-INT-LOGOUT-003	Cookie dibersihkan setelah keluar.	1. <i>Login</i> dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!".	Cookie kode akses dan penyegar

Tabel 4.11 Test Case Logout Mahasiswa (Lanjut)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01		Kondisi awal: Sesi aktif.	2. Logout melalui cookie. 3. Periksa header respons Set-Cookie.	kosong.
	TC-INT-LOGOUT-004	Keluar dua kali dengan kode sama. Kondisi awal: Sesi aktif.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses (pertama). 3. Logout lagi dengan token akses yang sama (kedua).	Pertama berhasil (200). Kedua ditolak (401).
	TC-INT-LOGOUT-005	Kode akses tanpa data valid. Kondisi awal: Tidak ada sesi.	1. Buat token akses tanpa data payload (sid, uid). 2. Kirim permintaan <i>logout</i> dengan token tersebut.	Permintaan ditolak (401).
	TC-INT-LOGOUT-006	Kode akses tanpa ID sesi. Kondisi awal: Tidak ada sesi.	1. Buat token akses hanya dengan ID user (tanpa sid). 2. Kirim permintaan <i>logout</i> .	Permintaan ditolak (401).
	TC-INT-LOGOUT-007	Header kode akses tanpa awalan Bearer. Kondisi awal: Sesi aktif.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Kirim header Authorization dengan nilai token akses langsung (tanpa awalan "Bearer ").	Permintaan ditolak (401). Format salah.
	TC-INT-LOGOUT-008	User dihapus tetapi keluar tetap berhasil. Kondisi awal: Sesi aktif, user dihapus.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Hapus user dari basis data. 3. Kirim permintaan <i>logout</i> dengan token akses.	Permintaan berhasil (200).
	TC-INT-LOGOUT-009	Multiple sesi - hanya satu terhapus saat keluar. Kondisi awal: Dua sesi aktif untuk user sama.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!" dua kali untuk mendapat dua sesi. 2. Logout dengan token akses sesi pertama. 3. Periksa data di basis data.	Sesi pertama hilang, sesi kedua masih ada.
	TC-INT-LOGOUT-010	Cookie dan header kode akses dikirim bersamaan. Kondisi awal: Sesi aktif.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Kirim <i>logout</i> dengan cookie dan Authorization header bersamaan.	Permintaan berhasil (200).

Tabel 4.11 Test Case Logout Mahasiswa (Lanjut)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-01	TC-INT-LOGOUT-011	Keluar - status sukses. Kondisi awal: Sesi aktif.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses. 3. Periksa status.	Status "success".
	TC-INT-LOGOUT-012	Keluar - pesan sesuai. Kondisi awal: Sesi aktif.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses. 3. Periksa pesan.	Pesan "Logout successful".
	TC-INT-LOGOUT-013	Keluar tanpa masuk sebelumnya. Kondisi awal: Tidak ada sesi.	1. Kirim permintaan <i>logout</i> tanpa token akses.	Permintaan ditolak (401).
	TC-INT-LOGOUT-014	Keluar - kode akses tidak bisa dipakai lagi. Kondisi awal: Sesi aktif.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses. 3. Coba akses fitur dengan token akses lama.	Kode akses tidak valid.
	TC-INT-LOGOUT-015	Keluar - data user tidak berubah. Kondisi awal: Sesi aktif.	1. Login dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 2. Catat data user. 3. Logout dengan token akses. 4. Bandingkan data.	Data user tidak berubah.

Tabel 4.11 menyajikan *test case* untuk fitur *logout* pada pengguna dengan peran sebagai mahasiswa yang mencakup beberapa skenario pengujian. Skenario utama yang diuji meliputi proses *logout* yang berhasil ketika pengguna masih memiliki sesi aktif dan token autentikasi yang valid. Setiap skenario dilengkapi dengan langkah-langkah pengujian serta hasil yang diharapkan untuk memastikan bahwa proses *logout* dapat mengakhiri sesi pengguna dengan benar dan membatasi akses ke sistem setelah pengguna keluar dari akun.

Tabel 4. 12 Test Case Kelola Profil Mahasiswa

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-02	TC-MHS-UP-01	Ubah nama berhasil. Kondisi awal: User mahasiswa dengan nama "Old Name", kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan nama "New Name". 3. Periksa data di basis data.	Permintaan berhasil (200). Nama berubah jadi "New Name".
	TC-MHS-UP-02	Ubah NIM berhasil. Kondisi awal: User mahasiswa, kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan NIM "12345678". 3. Periksa data di basis data.	Permintaan berhasil (200). NIM berubah.
	TC-MHS-UP-03	Ubah foto melalui unggah file berhasil.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan foto. 3. Periksa data di basis data.	Permintaan berhasil (200). Foto terisi.
	TC-MHS-UP-04	Ubah nama dan NIM sekaligus. Kondisi awal: User mahasiswa, kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan nama "Updated" dan NIM "999". 3. Periksa data di basis data.	Permintaan berhasil (200). Kedua field berubah.
	TC-MHS-UP-05	Ubah dengan data kosong. Kondisi awal: User mahasiswa, kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan tanpa mengisi field apapun (body kosong).	Permintaan ditolak (400). Tidak ada field diubah.
	TC-MHS-UP-06	Ubah tanpa kode akses. Kondisi awal: User mahasiswa terdaftar.	1. Kirim data perubahan (nama "New Name") tanpa token akses.	Permintaan ditolak (401).
	TC-MHS-UP-07	Ubah dengan kode akses tidak valid. Kondisi awal: User mahasiswa terdaftar.	1. Kirim data perubahan (nama "New Name") dengan token akses "invalid".	Permintaan ditolak (401).

Tabel 4.12 Test Case Kelola Profil Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-02	TC-MHS-UP-08	Ubah oleh psikolog ditolak. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> menggunakan akun psikolog (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan profil mahasiswa (nama "New Name").	Permintaan ditolak (403).
	TC-MHS-UP-09	Ubah oleh admin ditolak. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> menggunakan akun admin (email "admin@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan profil mahasiswa (nama "New Name").	Permintaan ditolak (403).
	TC-MHS-UP-10	Ubah dengan nama kosong. Kondisi awal: User mahasiswa, kode akses valid.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan dengan nama "" (string kosong).	Permintaan ditolak (400).
	TC-MHS-UP-11	Ubah dengan NIM kosong - diabaikan. Kondisi awal: User mahasiswa, kode akses valid.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan dengan NIM ""	Permintaan berhasil (200). NIM tidak berubah.
	TC-MHS-UP-12	Ubah dengan field tidak dikenal. Kondisi awal: User mahasiswa, kode akses valid.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim field "unknown": "value".	Permintaan ditolak (400). Field tidak dikenal.
	TC-MHS-UP-13	Ubah dengan nama sangat panjang (200 karakter). Kondisi awal: User mahasiswa, kode akses valid.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim nama 200 karakter.	Permintaan berhasil (200).
	TC-MHS-UP-14	Ubah dua kali dengan data sama. Kondisi awal: User mahasiswa, kode akses valid.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Ubah nama "First" (pertama). 3. Ubah lagi nama "First" (kedua).	Kedua berhasil (200). Nama tetap.

Tabel 4.12 Test Case Kelola Profil Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-02	TC-MHS-UP-15	Ubah foto lama ke baru. Kondisi awal: User sudah punya foto.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Unggah foto lama. 3. Unggah foto baru. 4. Bandingkan alamat.	Alamat foto berbeda.
	TC-INT-UP-001	Ubah profil - user tidak ditemukan. Kondisi awal: User dihapus setelah masuk.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Hapus user dari basis data. 3. Kirim data perubahan nama "New Name".	Permintaan ditolak (401).
	TC-INT-UP-002	Ubah profil - data tetap setelah perubahan. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Ubah nama "New Name". 3. Periksa data di basis data.	Data sesuai perubahan.
	TC-INT-UP-003	Ubah profil - NIM bukan angka ditolak. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim NIM "abc".	Permintaan ditolak (400).
	TC-INT-UP-004	Ubah profil - NIM terlalu panjang. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim NIM 20 digit.	Permintaan ditolak (400).
	TC-INT-UP-005	Ubah profil - email tidak bisa diubah. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan email "new@mail.com".	Permintaan ditolak (400).
	TC-INT-UP-006	Ubah profil - nama dengan angka ditolak. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim nama "Nama123".	Permintaan ditolak (400).
	TC-INT-UP-007	Ubah profil - format respons. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan	Respons berisi data user baru.

Tabel 4.12 Test Case Kelola Profil Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-02			valid (nama "New Name"). 3. Periksa respons.	
	TC-INT-UP-008	Ubah profil - nama hanya spasi ditolak. Kondisi awal: User terdaftar.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim nama " ".	Permintaan ditolak (400).
	TC-INT-UP-009	Ubah profil - status sukses. Kondisi awal: User terdaftar.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan valid (nama "New Name").	Status "success".
	TC-INT-UP-010	Ubah profil - nama 1 karakter ditolak. Kondisi awal: User terdaftar.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim nama "A".	Permintaan ditolak (400).
	TC-INT-UP-011	Ubah profil - tidak bisa ubah user lain. Kondisi awal: Dua user mahasiswa (A dan B).	1. <i>Login</i> menggunakan akun mahasiswa A. 2. Coba ubah profil B dengan mengirim ID B.	Permintaan ditolak (403).
	TC-INT-UP-012	Ubah profil - file bukan gambar ditolak. Kondisi awal: User terdaftar.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Unggah file "test.txt".	Permintaan ditolak (400).
	TC-INT-UP-013	Ubah profil - foto terlalu besar. Kondisi awal: User terdaftar.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Unggah foto 10 MB.	Permintaan ditolak (400).
	TC-INT-UP-014	Ubah profil - field asing diabaikan. Kondisi awal: User terdaftar.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim field valid (nama "New Name") + field asing ("unknown": "value").	Field valid berubah, asing diabaikan.
	TC-INT-UP-015	Ubah profil - konsisten setelah perubahan. Kondisi awal: User terdaftar.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Ubah nama "New Name". 3. <i>Logout</i> .	Nama baru masih tersimpan.

Tabel 4.12 *Test Case* Kelola Profil Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-02			4. Login lagi dengan email "mahasiswa@mail.com" dan kata sandi "Valid123!". 5. Periksa profil.	
	TC-INT-UP-016	Ubah profil - NIM karakter khusus. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim NIM "1234-5678".	Permintaan ditolak (400).
	TC-INT-UP-017	Ubah profil - tanpa perubahan tetap berhasil. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan dengan nilai sama (nama sesuai data saat ini).	Permintaan berhasil (200).
	TC-INT-UP-018	Ubah profil - foto opsional. Kondisi awal: User terdaftar.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan (nama "New Name") tanpa menyertakan file foto.	Permintaan berhasil (200). Foto tidak berubah.

Tabel 4.12 menyajikan *test case* untuk fitur kelola profil pada peran mahasiswa yang mencakup skenario pembaruan data profil seperti nama lengkap serta validasi kesalahan seperti penyimpanan data kosong dan percobaan akses oleh peran yang tidak sesuai.

Tabel 4. 13 *Test Case* Konsultasi Mahasiswa

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-03	CONS-MHS-01	Ajukan konsultasi tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan konsultasi (dengan psychologistId valid dan message "Saya ingin konsultasi") tanpa token akses.	Permintaan ditolak (401). Tidak ada konsultasi dibuat.
	CONS-MHS-02	Ajukan konsultasi sebagai psikolog ditolak. Kondisi awal:	1. Masuk sebagai psikolog (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan	Permintaan ditolak (403). Hanya mahasiswa.

Tabel 4.13 Test Case Konsultasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-03		Kode akses psikolog valid.	konsultasi dengan psychologistId valid dan message "Saya ingin konsultasi".	
	CONS-MHS-03	Ajukan konsultasi tanpa ID psikolog. Kondisi awal: Kode akses mahasiswa valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim konsultasi tanpa menyertakan psychologistId, dengan message "Saya ingin konsultasi".	Permintaan ditolak (400). ID psikolog wajib.
	CONS-MHS-04	Ajukan dengan ID psikolog tidak valid. Kondisi awal: Kode akses mahasiswa valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim psychologistId "123" dan message "Saya ingin konsultasi".	Permintaan ditolak (400). Format ID salah.
	CONS-MHS-05	Ajukan dengan psikolog tidak ditemukan. Kondisi awal: Kode akses mahasiswa valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim psychologistId palsu (ObjectId valid yang tidak ada) dan message "Saya ingin konsultasi".	Permintaan ditolak (404). Psikolog tidak ada.
	CONS-MHS-06	Ajukan dengan pesan kosong. Kondisi awal: ID psikolog valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim psychologistId valid dan message "" (string kosong).	Permintaan ditolak (400).
	CONS-MHS-07	Ajukan dengan pesan terlalu panjang (1001 karakter). Kondisi awal: ID psikolog valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim psychologistId valid dan message 1001 karakter.	Permintaan ditolak (413).
	CONS-MHS-08	Ajukan duplikat (sudah ada yang menunggu). Kondisi awal: Konsultasi pending sudah ada.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim konsultasi ke psikolog yang sama dengan message "Saya ingin konsultasi".	Permintaan ditolak (409).

Tabel 4.13 Test Case Konsultasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-03	CONS-MHS-09	Ajukan konsultasi berhasil. Kondisi awal: Belum ada konsultasi, ID psikolog valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim psychologistId valid dan message "Saya ingin konsultasi". 3. Periksa basis data.	Permintaan berhasil (201). Konsultasi baru status "menunggu".
	CONS-MHS-10	Lihat riwayat tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan riwayat konsultasi tanpa token akses.	Permintaan ditolak (401).
	CONS-MHS-11	Lihat riwayat - hanya milik sendiri. Kondisi awal: Ada 2 konsultasi (milik user ini + user lain).	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan riwayat. 3. Hitung data.	Permintaan berhasil (200). Hanya 1 konsultasi.
	CONS-MHS-12	Riwayat kosong (belum pernah ajukan). Kondisi awal: Tidak ada konsultasi.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan riwayat.	Permintaan berhasil (200). Data kosong.
	CONS-MHS-13	Lihat detail konsultasi tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan detail konsultasi (dengan ID konsultasi valid) tanpa token akses.	Permintaan ditolak (401).
	CONS-MHS-14	Lihat detail dengan ID tidak valid. Kondisi awal: Kode akses valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim detail dengan ID "abc".	Permintaan ditolak (400).
	CONS-MHS-15	Lihat detail konsultasi tidak ditemukan. Kondisi awal: ID tidak ada di basis data.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim detail dengan ID palsu (ObjectId valid yang tidak ada).	Permintaan ditolak (404).
	CONS-MHS-16	Akses detail konsultasi milik user lain.	1. Masuk sebagai mahasiswa A (email "mahasiswa@mail.com",	Permintaan ditolak (403).

Tabel 4.13 Test Case Konsultasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-03		Kondisi awal: Konsultasi milik user lain ada.	kata sandi "Valid123!"). 2. Akses detail konsultasi milik B dengan mengirim ID konsultasi B.	
	CONS-MHS-17	Lihat detail milik sendiri berhasil. Kondisi awal: Konsultasi milik sendiri ada.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!") 2. Kirim detail dengan ID konsultasi milik sendiri. 3. Periksa respons..	Permintaan berhasil (200). Data sesuai.
	CONS-MHS-18	Ajukan - status konsultasi "menunggu". Kondisi awal: Kode akses valid.	1. Masuk sebagai mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!") 2. Ajukan konsultasi dengan psychologistId valid dan message "Saya ingin konsultasi". 3. Periksa status.	Status "menunggu" (pending).
	TC-INT-CONS-MHS-001	Riwayat - format respons. Kondisi awal: Ada konsultasi.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!") 2. Kirim riwayat. 3. Periksa format.	Setiap item punya: ID, psikolog, status, tanggal.
	TC-INT-CONS-MHS-002	Detail - data lengkap. Kondisi awal: Konsultasi milik user ada.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!") 2. Kirim detail dengan ID konsultasi milik sendiri. 3. Periksa data.	Data lengkap termasuk pesan dan status.
	TC-INT-CONS-MHS-003	Ajukan ke psikolog tidak tersedia. Kondisi awal: Psikolog sedang tidak tersedia.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!") 2. Kirim permintaan ke psikolog tersebut dengan message "Saya ingin konsultasi".	Berhasil (201) atau ditolak.
	TC-INT-CONS-MHS-004	Ajukan dengan pesan 1 karakter. Kondisi awal: Kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!") 2. Kirim psychologistId valid dan message "A".	Berhasil (201) atau ditolak (400).

Tabel 4.13 Test Case Konsultasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-03	TC-INT-CONS-MHS-005	Riwayat diurutkan berdasarkan tanggal. Kondisi awal: Konsultasi dengan tanggal berbeda.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim riwayat. 3. Periksa urutan.	Terbaru ke terlama.
	TC-INT-CONS-MHS-006	Detail setelah diterima psikolog. Kondisi awal: Konsultasi sudah diterima.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim detail dengan ID konsultasi. 3. Periksa status.	Status "diterima".
	TC-INT-CONS-MHS-007	Detail setelah ditolak psikolog. Kondisi awal: Konsultasi sudah ditolak.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim detail dengan ID konsultasi.	Status "ditolak".
	TC-INT-CONS-MHS-008	Ajukan ke psikolog berbeda - keduanya berhasil. Kondisi awal: Kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan ke psikolog A dengan message "Saya ingin konsultasi". 3. Kirim permintaan ke psikolog B dengan message "Saya ingin konsultasi".	Kedua berhasil (201).
	TC-INT-CONS-MHS-009	Riwayat - filter berdasarkan status. Kondisi awal: Konsultasi dengan berbagai status.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim riwayat dengan filter status. 3. Periksa data.	Data sesuai filter.
	TC-INT-CONS-MHS-010	Ajukan - pesan dengan karakter khusus. Kondisi awal: Kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim psychologistId valid dan message "<script>alert(1)</script>".	Berhasil (201) atau konten dibersihkan.
	TC-INT-CONS-MHS-011	Detail - tidak termasuk data sensitif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!").	Tidak ada kata sandi.

Tabel 4.13 Test Case Konsultasi Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-03		Kondisi awal: Konsultasi ada.	2. Kirim detail dengan ID konsultasi. 3. Periksa respons.	
	TC-INT-CONS-MHS-012	Ajukan - sesi kadaluwarsa. Kondisi awal: Sesi mahasiswa kadaluwarsa.	1. Tunggu hingga token akses kadaluwarsa. 2. Kirim konsultasi dengan psychologistId valid dan message "Saya ingin konsultasi".	Permintaan ditolak (401).
	TC-INT-CONS-MHS-013	Riwayat - pagination. Kondisi awal: Banyak konsultasi.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim riwayat dengan parameter batasan (limit). 3. Periksa jumlah.	Jumlah sesuai batasan.
	TC-INT-CONS-MHS-014	Ajukan - data konsultasi lengkap. Kondisi awal: Berhasil mengajukan.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Ajukan dengan psychologistId valid dan message "Saya ingin konsultasi". 3. Periksa data di basis data.	Ada ID mahasiswa, ID psikolog, pesan, status.
	TC-INT-CONS-MHS-015	Riwayat - akses psikolog ditolak. Kondisi awal: Psikolog login.	1. Login menggunakan akun psikolog (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim riwayat konsultasi mahasiswa.	Permintaan ditolak (403).
	TC-INT-CONS-MHS-016	Ajukan - psikolog sama, konsultasi sebelumnya selesai. Kondisi awal: Konsultasi sebelumnya selesai.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan ke psikolog yang sama dengan message "Saya ingin konsultasi".	Permintaan berhasil (201).
	TC-INT-CONS-MHS-017	Riwayat - data hanya milik user login. Kondisi awal: Konsultasi milik berbagai user.	1. Login menggunakan akun mahasiswa A (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim riwayat. 3. Periksa semua data.	Hanya milik A yang muncul.

Tabel 4.13 menyajikan *test case* untuk fitur konsultasi pada peran mahasiswa yang mencakup skenario pengajuan konsultasi berhasil serta berbagai validasi kesalahan seperti akses tanpa *login*, penggunaan peran yang tidak sesuai, dan pengajuan tanpa memilih psikolog.

Tabel 4. 14 Test Case Chat Mahasiswa

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-04	TC-MHS-01	Daftar ruang obrolan tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan daftar ruang obrolan tanpa token akses.	Permintaan ditolak (401).
	TC-MHS-02	Daftar ruang obrolan berhasil. Kondisi awal: Ruang milik user ada.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan daftar ruang obrolan.	Permintaan berhasil (200). Daftar ruang.
	TC-MHS-03	Daftar ruang untuk user baru (belum punya ruang). Kondisi awal: User baru tanpa ruang.	1. Daftar user baru dengan email "newstudent@mail.com", kata sandi "Password1!", peran "mahasiswa", nama "Baru", NIM "87654321". 2. <i>Login</i> menggunakan akun baru. 3. Kirim daftar ruang obrolan.	Permintaan berhasil (200). Daftar kosong.
	TC-MHS-04	Hanya ruang milik user yang dikembalikan. Kondisi awal: Ruang milik user lain ada.	1. <i>Login</i> menggunakan akun mahasiswa A (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim daftar ruang obrolan.	Ruang milik B tidak muncul.
	TC-MHS-05	Dapatkan pesan tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan pesan (dengan roomId valid) tanpa token akses.	Permintaan ditolak (401).
	TC-MHS-06	Dapatkan pesan dari ruang tidak ditemukan. Kondisi awal:	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan dengan	Permintaan ditolak (404).

Tabel 4.14 Test Case Chat Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-04		Kode akses valid.	ID ruang palsu (ObjectId valid yang tidak ada).	
	TC-MHS-07	Dapatkan pesan berhasil. Kondisi awal: Ruang milik user ada dengan pesan.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan dengan ID ruang valid.	Permintaan berhasil (200). Daftar pesan.
	TC-MHS-08	Akses ruang bukan milik user. Kondisi awal: Ruang milik user lain.	1. Login menggunakan akun mahasiswa A (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Akses ruang milik B dengan mengirim ID ruang B.	Permintaan ditolak (403).
	TC-MHS-09	Kirim pesan tanpa field wajib. Kondisi awal: Kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan body kosong (tanpa roomId, senderId, content).	Permintaan ditolak (400).
	TC-MHS-10	Kirim pesan berhasil. Kondisi awal: Ruang aktif milik user.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user login, content "Test message". 3. Periksa respons.	Permintaan berhasil (201). Isi pesan "Test message".
	TC-MHS-11	Pesan tidak disimpan ke basis data. Kondisi awal: Ruang ada.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content "Test". 3. Periksa data di basis data.	Pesan tidak tersimpan.
	TC-MHS-12	Tutup ruang tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan tutup ruang obrolan (dengan roomId valid) tanpa token akses.	Permintaan ditolak (401).
	TC-MHS-13	Tutup ruang tidak ditemukan. Kondisi awal: Kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!").	Permintaan ditolak (404).

Tabel 4.14 Test Case Chat Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-04			2. Tutup ruang obrolan dengan ID palsu (ObjectId valid yang tidak ada).	
	TC-MHS-14	Tutup ruang berhasil. Kondisi awal: Ruang aktif milik user.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Tutup ruang obrolan dengan roomId valid. 3. Periksa data di basis data.	Permintaan berhasil (200). Status jadi "tidak aktif".
	TC-MHS-15	Tutup ruang dua kali. Kondisi awal: Ruang sudah tidak aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Tutup ruang obrolan dengan roomId valid (pertama). 3. Tutup lagi dengan roomId yang sama (kedua).	Kedua berhasil (200). Status tetap.
	TC-MHS-16	Dapatkan pesan setelah ruang ditutup. Kondisi awal: Ruang tidak aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Tutup ruang obrolan. 3. Kirim permintaan pesan dengan roomId yang sama.	Permintaan berhasil (200). Masih bisa baca.
	TC-MHS-17	Kirim pesan ke ruang yang sudah ditutup. Kondisi awal: Ruang tidak aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Tutup ruang obrolan. 3. Kirim pesan dengan roomId yang sama, senderId sesuai ID user, content "Test".	Berhasil (201) atau ditolak (400).
	TC-MHS-18	Kirim pesan terlalu panjang (1001 karakter). Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content 1001 karakter.	Permintaan ditolak (413).
	TC-MHS-19	Kirim pesan mengandung skrip HTML. Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan	Ditolak (400) atau dibersihkan.

Tabel 4.14 Test Case Chat Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-04			roomId valid, senderId sesuai ID user, content "<script>alert(1)</script>".	
	TC-MHS-20	Kirim pesan - senderId tidak cocok. Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa A (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim dengan roomId valid, senderId ID user B, content "Test".	Ditolak (403) atau senderId dipaksa ke A.
	TC-MHS-21	Daftar ruang dengan status berbeda. Kondisi awal: Ruang aktif dan tidak aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim daftar ruang obrolan. 3. Periksa status.	Status setiap ruang tercantum.
	TC-MHS-22	Kirim pesan hanya spasi. Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content " ".	Permintaan ditolak (400).
	TC-MHS-23	Kirim pesan - roomId tidak valid. Kondisi awal: Kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim roomId "abc", senderId sesuai ID user, content "Test".	Permintaan ditolak (400).
	TC-MHS-24	Dapatkan pesan - ruang tanpa pesan. Kondisi awal: Ruang baru tanpa pesan.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan pesan dengan roomId valid.	Daftar kosong.
	TC-MHS-25	Kirim pesan gambar (URL). Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content URL gambar.	Permintaan berhasil (201).
	TC-MHS-26	Daftar ruang - hanya milik role	1. Login menggunakan akun mahasiswa (email	Ruang milik psikolog tidak

Tabel 4.14 Test Case Chat Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-04		sendiri. Kondisi awal: Kode akses mahasiswa valid.	"mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim daftar ruang obrolan.	muncul.
	TC-MHS-27	Kirim pesan sukses. Kondisi awal: Ruang aktif.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content "Test".	Permintaan berhasil (201).
	TC-INT-CHAT-MHS-001	Daftar ruang - format respons. Kondisi awal: Ada ruang.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim daftar ruang obrolan. 3. Periksa format.	Setiap ruang: ID, partisipan, status.
	TC-INT-CHAT-MHS-002	Dapatkan pesan - format respons. Kondisi awal: Ada pesan.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan pesan dengan roomId valid. 3. Periksa format.	Setiap pesan: pengirim, isi, waktu.
	TC-INT-CHAT-MHS-003	Kirim pesan - senderId dipaksa dari token. Kondisi awal: Ruang aktif.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim dengan roomId valid, senderId berbeda dari ID user <i>login</i> , content "Test". 3. Periksa respons.	senderId sesuai user <i>login</i> .
	TC-INT-CHAT-MHS-004	Tutup ruang - bukan pemilik. Kondisi awal: Dua user berbeda.	1. <i>Login</i> menggunakan akun mahasiswa A (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Tutup ruang obrolan milik B dengan mengirim roomId milik B.	Permintaan ditolak (403).
	TC-INT-CHAT-MHS-005	Daftar ruang - setelah ruang ditutup. Kondisi awal: Ruang sudah ditutup.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim daftar ruang obrolan. 3. Periksa status.	Ruang ditutup tetap muncul.

Tabel 4.14 Test Case Chat Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-04	TC-INT-CHAT-MHS-006	Kirim pesan dengan emoji. Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content "😊".	Permintaan berhasil (201).
	TC-INT-CHAT-MHS-007	Dapatkan pesan - urutan lama ke baru. Kondisi awal: Beberapa pesan.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan pesan dengan roomId valid. 3. Periksa urutan.	Dari paling lama ke baru.
	TC-INT-CHAT-MHS-008	Kirim pesan hanya angka. Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content "12345".	Permintaan berhasil (201).
	TC-INT-CHAT-MHS-009	Daftar ruang - hanya role mahasiswa. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim daftar ruang obrolan.	Hanya ruang di mana user partisipan.
	TC-INT-CHAT-MHS-010	Kirim pesan tanpa roomId. Kondisi awal: Kode akses valid.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan tanpa menyertakan roomId, dengan senderId sesuai ID user dan content "Test".	Permintaan ditolak (400).
	TC-INT-CHAT-MHS-011	Dapatkan pesan - pengirim psikolog. Kondisi awal: Ada pesan dari psikolog.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan pesan dengan roomId valid. 3. Periksa data.	Pengirim adalah psikolog.
	TC-INT-CHAT-MHS-012	Kirim pesan - batas panjang. Kondisi awal: Ruang aktif.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai	Permintaan berhasil (201).

Tabel 4.14 Test Case Chat Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-04			ID user, content tepat batas maksimal karakter.	
	TC-INT-CHAT-MHS-013	Daftar ruang - tanpa ruang sama sekali. Kondisi awal: Tidak ada ruang.	1. <i>Login</i> menggunakan akun mahasiswa baru (tanpa ruang). 2. Kirim daftar ruang obrolan.	Data kosong (200).
	TC-INT-CHAT-MHS-014	Kirim pesan - sesi kadaluwarsa. Kondisi awal: Sesi kadaluwarsa.	1. Tunggu hingga token akses kadaluwarsa. 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content "Test".	Permintaan ditolak (401).
	TC-INT-CHAT-MHS-015	Dapatkan pesan - setelah psikolog kirim. Kondisi awal: Psikolog baru kirim pesan.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan pesan dengan roomId valid. 3. Periksa pesan baru.	Pesan psikolog muncul.
	TC-INT-CHAT-MHS-016	Tutup ruang - hanya peserta ruang. Kondisi awal: Ruang milik A dan B.	1. <i>Login</i> menggunakan akun mahasiswa C (bukan peserta ruang). 2. Coba tutup ruang milik A dan B dengan mengirim roomId.	Permintaan ditolak (403).
	TC-INT-CHAT-MHS-017	Kirim pesan mengandung link. Kondisi awal: Ruang aktif.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan roomId valid, senderId sesuai ID user, content "https://example.com".	Permintaan berhasil (201).
	TC-INT-CHAT-MHS-018	Daftar ruang - urutan aktivitas terbaru. Kondisi awal: Beberapa ruang.	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim daftar ruang obrolan. 3. Periksa urutan.	Aktivitas terbaru di atas.

Tabel 4.14 menyajikan *test case* untuk fitur *chat* pada peran mahasiswa yang mencakup skenario melihat daftar ruang percakapan, akses tanpa *login*, pembatasan akses ke ruang percakapan milik pengguna lain, serta validasi bahwa hanya ruang

percakapan milik sendiri yang ditampilkan, lengkap dengan kondisi awal, langkah pengujian, dan hasil yang diharapkan.

Tabel 4. 15 Test Case Chatbot Mahasiswa

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-05	TC-CHATBOT-SVC-01	Kirim <i>chat</i> tanpa pesan. Kondisi awal: Tidak ada data khusus.	1. Siapkan data <i>chat</i> dengan messages [] (array kosong). 2. Kirim <i>chat</i> ke sistem.	Permintaan ditolak (400). Pesan "Messages is required".
	TC-CHATBOT-SVC-02	Kirim <i>chat</i> dengan 1 pesan user. Kondisi awal: Layanan Groq diatur respons "Test reply".	1. Siapkan data <i>chat</i> dengan messages [{ "role": "user", "content": "Halo" }]. 2. Kirim <i>chat</i> ke sistem.	Permintaan berhasil (200). Respons peran "assistant", isi "Test reply".
	TC-CHATBOT-SVC-03	Layanan <i>chatbot</i> error internal. Kondisi awal: Layanan Groq gagal (error 500).	1. Siapkan data <i>chat</i> dengan messages [{ "role": "user", "content": "Halo" }]. 2. Kirim <i>chat</i> ke sistem.	Permintaan gagal (500). Pesan "Chatbot service failed".
	TC-CHATBOT-SVC-04	Format respons mengandung status dan data. Kondisi awal: Layanan normal.	1. Kirim <i>chat</i> ke sistem dengan messages [{ "role": "user", "content": "Halo" }]. 2. Periksa struktur.	JSON dengan status: "success", data: peran + isi.
	TC-CHATBOT-SVC-05	Kirim banyak pesan (riwayat percakapan). Kondisi awal: Layanan normal.	1. Kirim messages [{ "role": "user", "content": "A" }, { "role": "assistant", "content": "B" }, { "role": "user", "content": "C" }]. 2. Periksa respons.	Permintaan berhasil (200). Peran "assistant".
	TC-CHATBOT-SVC-06	Layanan kembali isi kosong - pakai cadangan. Kondisi awal: Groq kembali kosong.	1. Kirim <i>chat</i> ke sistem dengan messages [{ "role": "user", "content": "Halo" }]. 2. Periksa respons.	Permintaan berhasil (200). Isi "Maaf, tidak ada respons dari model."
	TC-CHATBOT-SVC-07	Kirim <i>chat</i> tanpa kode akses - tetap bisa. Kondisi awal: Tidak ada sesi.	1. Kirim <i>chat</i> ke sistem dengan messages [{ "role": "user", "content": "Halo" }] tanpa token akses.	Permintaan berhasil (200). Fitur publik.

Tabel 4.15 Test Case Chatbot Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-05	TC-CHATBOT-SVC-08	Kirim <i>chat</i> dengan format messages salah. Kondisi awal: Tidak ada data khusus.	1. Siapkan data <i>chat</i> dengan messages berupa teks "invalid" (bukan array). 2. Kirim <i>chat</i> ke sistem.	Permintaan ditolak (400).
	TC-CHATBOT-SVC-09	Kirim <i>chat</i> - messages kosong. Kondisi awal: Tidak ada data khusus.	1. Siapkan data <i>chat</i> dengan messages [] (array kosong). 2. Kirim <i>chat</i> ke sistem.	Permintaan ditolak (400).
	TC-CHATBOT-SVC-10	Kirim <i>chat</i> - peran tidak dikenal. Kondisi awal: Tidak ada data khusus.	1. Kirim messages [{"role": "unknown", "content": "Test"}].	Ditolak (400) atau diterima.
	TC-CHATBOT-SVC-11	Kirim <i>chat</i> - isi terlalu panjang. Kondisi awal: Tidak ada data khusus.	1. Kirim messages [{"role": "user", "content": "isi 10000 karakter"}].	Ditolak (413) atau dipotong.
	TC-CHATBOT-SVC-12	Kirim <i>chat</i> - banyak pesan. Kondisi awal: Layanan normal.	1. Kirim 50 pesan dalam satu permintaan (messages array dengan 50 objek).	Berhasil (200) atau ditolak (400).
	TC-CHATBOT-SVC-13	Kirim <i>chat</i> - karakter khusus. Kondisi awal: Layanan normal.	1. Kirim messages [{"role": "user", "content": "@#\$%^&*()"}].	Permintaan berhasil (200).
	TC-CHATBOT-SVC-14	Kirim <i>chat</i> - bahasa Indonesia. Kondisi awal: Layanan normal.	1. Kirim messages [{"role": "user", "content": "Saya merasa cemas hari ini"}].	Permintaan berhasil (200). Respons bahasa Indonesia.
	TC-CHATBOT-INT-01	Waktu respons cepat. Kondisi awal: Layanan normal.	1. Kirim <i>chat</i> ke sistem dengan messages [{"role": "user", "content": "Halo"}]. 2. Catat waktu.	Waktu di bawah batas.

Tabel 4.15 Test Case Chatbot Mahasiswa (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-MHS-05	TC-CHATBOT-INT-02	Layanan Groq tidak tersedia. Kondisi awal: Layanan Groq mati.	1. Kirim <i>chat</i> ke sistem dengan messages [{ "role": "user", "content": "Halo" }].	Permintaan gagal (503).
	TC-CHATBOT-INT-03	Kunci API tidak valid. Kondisi awal: Kunci API Groq salah.	1. Kirim <i>chat</i> ke sistem dengan messages [{ "role": "user", "content": "Halo" }].	Permintaan gagal (401 atau 500).
	TC-CHATBOT-INT-04	Banyak permintaan dalam waktu singkat. Kondisi awal: Layanan normal.	1. Kirim 10 <i>chat</i> berturut-turut dengan messages [{ "role": "user", "content": "Halo" }]. 2. Periksa respons.	Semua berhasil (200) atau dibatasi (429).
	TC-CHATBOT-INT-05	Konten tidak pantas. Kondisi awal: Layanan normal.	1. Kirim messages [{ "role": "user", "content": "konten tidak pantas" }].	Ditolak (400) atau difilter.
	TC-CHATBOT-INT-06	Pesan sangat pendek (1 karakter). Kondisi awal: Layanan normal.	1. Kirim messages [{ "role": "user", "content": "A" }].	Permintaan berhasil (200).

Tabel 4.15 menunjukkan bahwa seluruh skenario pengujian pada fitur *chatbot* mahasiswa telah dilaksanakan secara menyeluruh. Pengujian mencakup kondisi penggunaan normal, dan validasi terhadap berbagai jenis *input* pengguna.

4. Requirement Traceability Matrix (RTM) Mahasiswa

Requirement Traceability Matrix (RTM) Mahasiswa digunakan untuk menunjukkan keterlacakan antara setiap kebutuhan fungsional yang telah didefinisikan dengan skenario pengujian yang dilakukan.

Tabel 4. 16 RTM Mahasiswa

Scenario ID	Requirements Description	Test Case ID
TS-MHS-01 (Autentikasi)	Pengguna dapat mendaftar akun mahasiswa	TC-AUTH-01 s.d. TC-AUTH-30, TC-INT-REG-001 s.d. TC-INT-REG-043
TS-MHS-01 (Autentikasi)	Pengguna dapat <i>login</i> sebagai mahasiswa	TC-LOGIN-01 s.d. TC-LOGIN-10, TC-INT-LOGIN-001 s.d. TC-INT-LOGIN-024
TS-MHS-01 (Autentikasi)	Pengguna dapat <i>logout</i> dari akun mahasiswa	TC-LOGOUT-01 s.d. TC-LOGOUT-05, TC-INT-LOGOUT-001 s.d. TC-INT-LOGOUT-015
TS-MHS-02 (Kelola Profil)	Pengguna dapat mengubah profil (nama, NIM, foto)	TC-MHS-UP-01 s.d. TC-MHS-UP-15, TC-INT-UP-001 s.d. TC-INT-UP-018
TS-MHS-03 (Konsultasi)	Pengguna dapat mengajukan konsultasi ke psikolog	CONS-MHS-01 s.d. CONS-MHS-09, CONS-MHS-18, TC-INT-CONS-MHS-003, TC-INT-CONS-MHS-004, TC-INT-CONS-MHS-008, TC-INT-CONS-MHS-010, TC-INT-CONS-MHS-012, TC-INT-CONS-MHS-014, TC-INT-CONS-MHS-016
TS-MHS-03 (Konsultasi)	Pengguna dapat melihat riwayat dan detail konsultasi	CONS-MHS-10 s.d. CONS-MHS-17, TC-INT-CONS-MHS-001, TC-INT-CONS-MHS-002, TC-INT-CONS-MHS-005 s.d. TC-INT-CONS-MHS-007, TC-INT-CONS-MHS-009, TC-INT-CONS-MHS-011, TC-INT-CONS-MHS-013, TC-INT-CONS-MHS-015, TC-INT-CONS-MHS-017
TS-MHS-04 (Chat)	Pengguna dapat melihat daftar ruang <i>chat</i>	TC-MHS-01 s.d. TC-MHS-04, TC-MHS-21, TC-MHS-26, TC-INT-CHAT-MHS-001, TC-INT-CHAT-MHS-005, TC-INT-CHAT-MHS-009, TC-INT-CHAT-MHS-013, TC-INT-CHAT-MHS-018
TS-MHS-04 (Chat)	Pengguna dapat menutup ruang <i>chat</i>	TC-MHS-12 s.d. TC-MHS-15, TC-INT-CHAT-MHS-004, TC-INT-CHAT-MHS-016
TS-MHS-05 (Chatbot)	Pengguna dapat menggunakan <i>chatbot</i> AI	TC-CHATBOT-SVC-01 s.d. TC-CHATBOT-SVC-14, TC-CHATBOT-INT-01 s.d. TC-CHATBOT-INT-06

Tabel 4.16 menunjukkan bahwa seluruh kebutuhan fungsional pada modul mahasiswa telah dipetakan dengan *test case* yang sesuai, meliputi fitur autentikasi, pengelolaan profil, konsultasi, *chat*, dan *chatbot* AI.

B. Pengujian Role Psikolog

1. Main Map Fitur Psikolog

Berikut adalah seluruh fitur yang dapat diakses oleh pengguna dengan peran Psikolog:

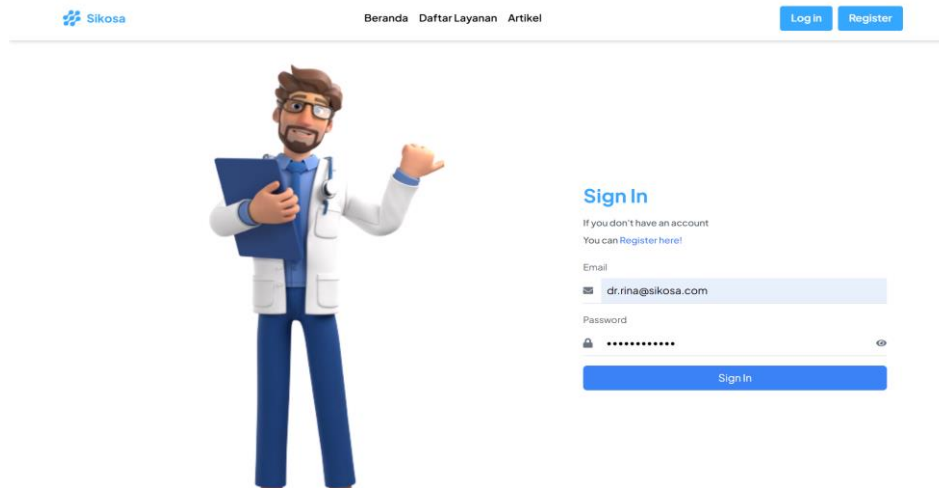
Tabel 4. 17 Main Map Fitur Psikolog

No	Fitur	Deskripsi
1	Login	Masuk ke sistem menggunakan akun psikolog
2	Kelola Profil	Memperbarui nama, deskripsi, spesialisasi, pendidikan, dan foto profil
3	Konsultasi	Menerima atau menolak pengajuan sesi konsultasi dari mahasiswa
4	Chat	Mengirim dan membaca pesan dalam sesi konsultasi aktif
5	Kelola Artikel	Membuat, mengubah, menghapus, dan melihat artikel
6	Logout	Keluar dari sistem

Tabel 4.17 menjabarkan peta utama (*main map*) dari tujuh fitur yang tersedia untuk hak akses Psikolog, yang mencakup fungsi autentikasi, manajemen profil dan artikel, serta pengelolaan konsultasi dan komunikasi pesan (*chat*).

a) Fitur Login Psikolog

Fitur *login* psikolog merupakan mekanisme autentikasi yang digunakan oleh pengguna dengan peran sebagai psikolog untuk memperoleh akses ke dalam sistem SiKosa. Proses autentikasi dilakukan dengan memasukkan kredensial yang valid, berupa alamat email dan kata sandi (*password*) yang telah terdaftar pada sistem.

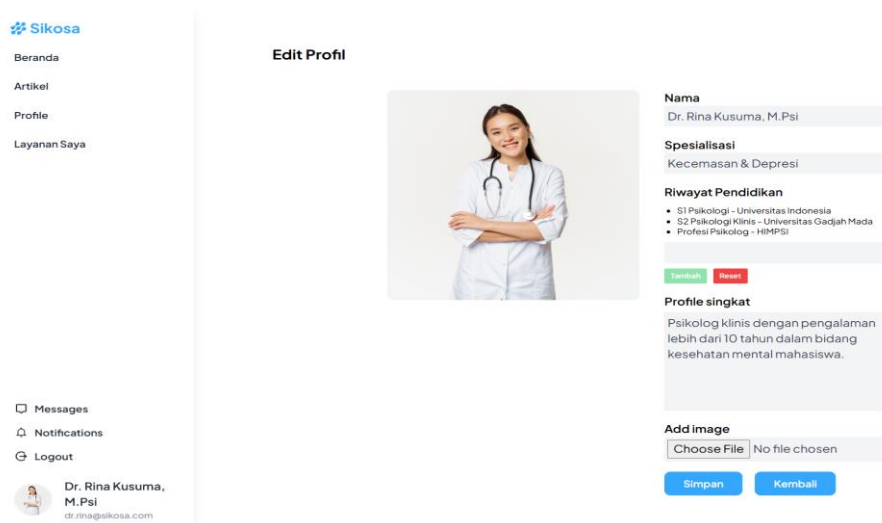


Gambar 4. 11 Halaman *login* psikolog

Gambar 4.11 menampilkan halaman *login* yang digunakan oleh psikolog untuk mengakses sistem. Pada halaman ini tersedia *form* autentikasi yang terdiri dari *input* email dan *password*. Psikolog harus memasukkan kredensial yang valid untuk dapat masuk ke dalam sistem.

b) Fitur Kelola Profil (Psikolog)

Fitur kelola profil psikolog digunakan untuk mengelola dan memperbarui informasi akun psikolog di dalam sistem yang relevan dengan identitas pengguna.

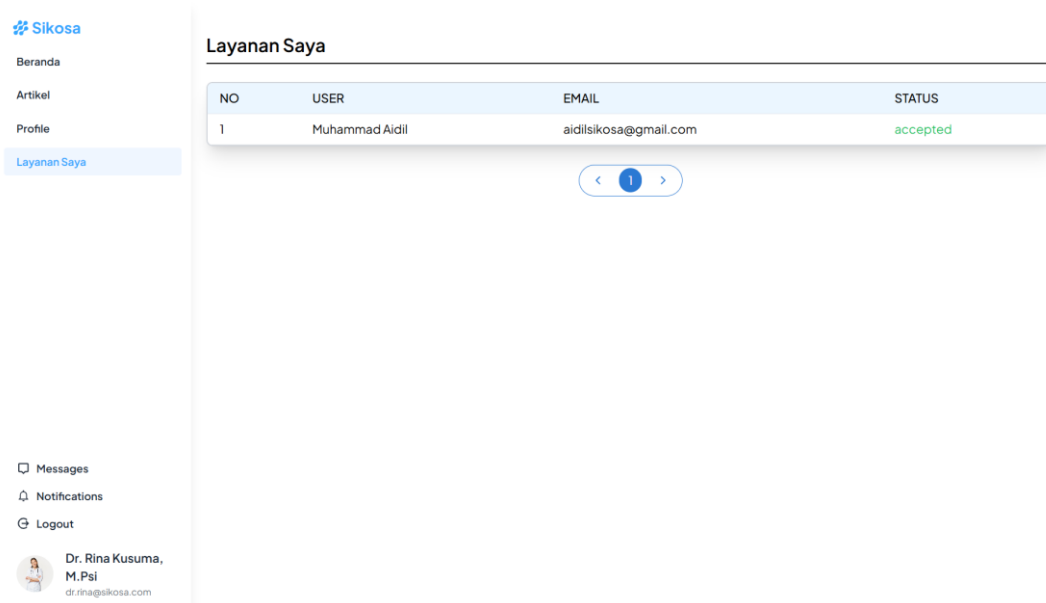


Gambar 4. 12 Halaman kelola profil psikolog

Gambar 4.12 menampilkan halaman kelola profil psikolog yang memungkinkan pengguna untuk melihat dan memperbarui informasi akun secara langsung. Pada halaman ini, psikolog dapat melakukan pengeditan data profil kemudian menyimpannya agar perubahan dapat diterapkan pada sistem.

c) Fitur Konsultasi (Psikolog)

Fitur konsultasi psikolog merupakan fitur utama yang digunakan untuk menangani proses komunikasi antara psikolog dan pengguna (mahasiswa).



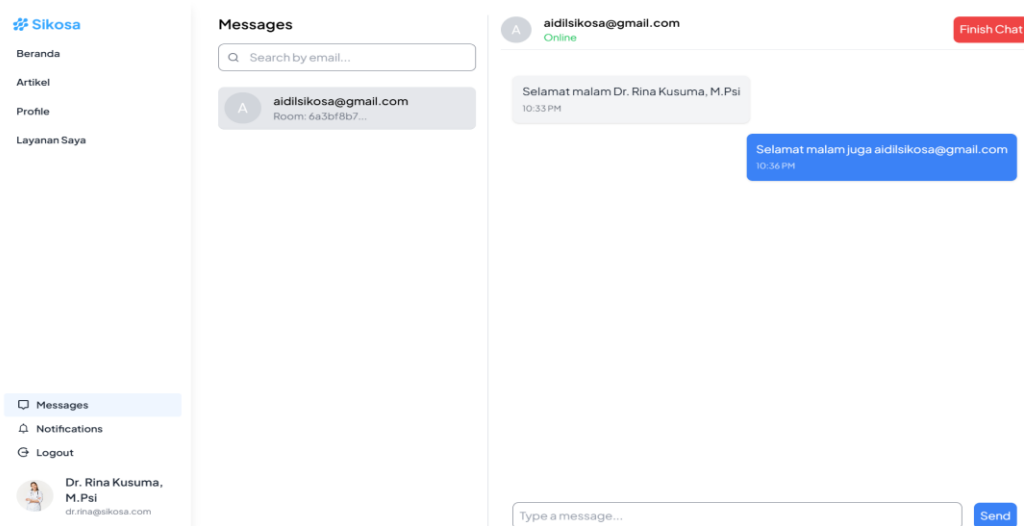
Gambar 4. 13 Halaman konsultasi (psikolog)

Gambar 4.13 menampilkan halaman konsultasi psikolog yang berisi daftar sesi konsultasi antara psikolog dan pengguna. Pada halaman ini, psikolog dapat mengelola status konsultasi sesuai kebutuhan layanan.

d) Fitur Chat (Psikolog)

Fitur chat (psikolog) digunakan sebagai media komunikasi *real-time* antara psikolog dan pengguna dalam proses konsultasi. Melalui fitur ini, psikolog dapat

mengirim dan menerima pesan secara langsung, sehingga interaksi dapat berlangsung lebih cepat dan efektif.

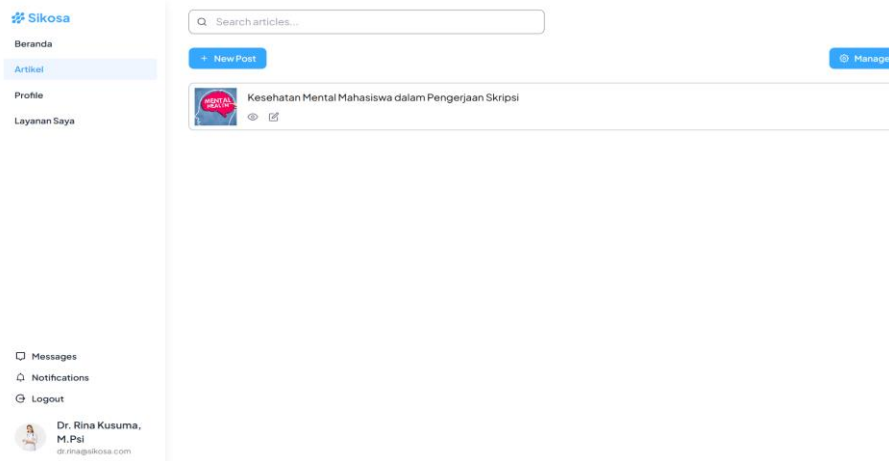


Gambar 4. 14 Halaman *chat* (psikolog)

Gambar 4.14 menampilkan halaman *chat* pada sisi psikolog yang digunakan sebagai media komunikasi antara psikolog dan mahasiswa selama proses konsultasi berlangsung. Halaman ini menyediakan ruang percakapan yang menampilkan riwayat pesan secara real-time, sehingga setiap pesan yang dikirim maupun diterima dapat langsung ditampilkan tanpa perlu melakukan penyegaran halaman. Melalui fitur ini, psikolog dapat memberikan tanggapan, arahan, maupun saran kepada mahasiswa secara interaktif.

e) Fitur Kelola Artikel

Fitur kelola artikel digunakan oleh psikolog untuk membuat, mengedit, dan menghapus artikel yang akan dipublikasikan pada sistem. Fitur ini bertujuan untuk menyediakan informasi dan edukasi terkait kesehatan mental kepada pengguna.

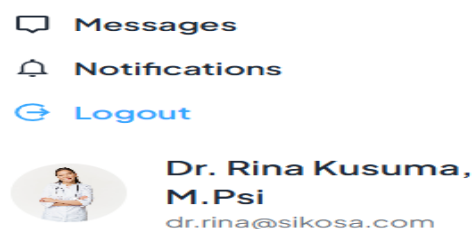


Gambar 4. 15 Halaman kelola artikel

Gambar 4.15 menampilkan halaman kelola artikel yang digunakan oleh psikolog untuk mengelola konten artikel. Pada halaman ini, psikolog dapat menambahkan artikel baru, melakukan perubahan pada artikel yang sudah ada, serta menghapus artikel yang tidak diperlukan sehingga konten yang tersedia tetap relevan dan terstruktur.

f) Fitur *Logout* (Psikolog)

Fitur *Logout* (Psikolog) digunakan untuk mengakhiri sesi *login* yang sedang aktif pada akun psikolog. Dengan menggunakan fitur ini, sistem akan menghapus sesi autentikasi pengguna sehingga akses ke halaman psikolog tidak lagi tersedia tanpa proses *login* ulang.



Gambar 4. 16 Fitur logout

Gambar 4.16 menampilkan proses *logout* yang dilakukan oleh psikolog pada sistem. Setelah pengguna memilih menu *logout*, sistem akan mengakhiri sesi aktif dan mengarahkan kembali ke halaman *login* sebagai bentuk validasi ulang akses pengguna.

2. Test Scenario Psikolog

Penyusunan *test scenario* didasarkan pada masing-masing fitur sistem guna membentuk landasan acuan.

Tabel 4. 18 Test Scenario Psikolog

ID Skenario	Skenario Pengujian
TS-PSI-01	Autentikasi: Proses masuk ke sistem dan keluar dari sistem
TS-PSI-02	Kelola Profil: Mengubah data diri seperti nama, deskripsi, spesialisasi, latar belakang pendidikan, dan foto profil
TS-PSI-03	Konsultasi: Melihat notifikasi konsultasi masuk serta menerima atau menolak permintaan konsultasi dari mahasiswa
TS-PSI-04	Chat: Berkomunikasi melalui pesan teks dengan mahasiswa yang konsultasinya telah diterima
TS-PSI-05	Kelola Artikel: Membuat, mengubah, melihat, dan menghapus artikel yang ditulis

Tabel 4.18 menyajikan skenario pengujian pada modul psikolog yang mencakup seluruh fungsi utama sistem, yaitu autentikasi, pengelolaan profil, konsultasi, *chat*, dan pengelolaan artikel.

3. Test Case Development (Psikolog)

Bagian ini menyajikan *test case* untuk pengguna dengan hak akses psikolog sebagai penjabaran teknis yang lebih rinci dari *test scenario* yang telah disusun pada tahap sebelumnya. Setiap *test case* memuat kondisi pengujian, langkah-langkah yang dilakukan, data masukan yang digunakan, serta hasil yang diharapkan.

Tabel 4. 19 Test Case Login Psikolog

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-01	TC-LOGIN-01	Login dengan email kosong. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengosongkan kolom email dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Email wajib diisi.
	TC-LOGIN-02	Login dengan kata sandi kosong. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "psikolog@mail.com" dan mengosongkan kolom kata sandi. 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Kata sandi wajib.
	TC-LOGIN-03	Login dengan format email salah. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "invalid" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Format salah.
	TC-LOGIN-04	Login dengan kata sandi terlalu pendek. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "psikolog@mail.com" dan mengisi kolom kata sandi dengan "123". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Minimal 8 karakter.
	TC-LOGIN-05	Login dengan email terlalu panjang. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan teks 269 karakter dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Melebihi batas.
	TC-LOGIN-06	Login dengan email tidak terdaftar. Kondisi awal: Email tidak ada di basis data.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "notfound@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (404).
	TC-LOGIN-07	Login dengan kata sandi salah. Kondisi awal: Akun psikolog terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "psikolog@mail.com" dan mengisi kolom kata sandi dengan "Wrong123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (401).

Tabel 4.19 Test Case Login Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-01	TC-LOGIN-08	Login psikolog berhasil. Kondisi awal: Akun psikolog dengan email "psikolog@mail.com", sandi "Valid123!" terdaftar.	1. Siapkan data login dengan mengisi kolom email dengan "psikolog@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data login ke sistem. 3. Periksa data.	Permintaan berhasil (200). Data dengan peran psikolog.
	TC-LOGIN-09	Login psikolog - ada kode akses. Kondisi awal: Akun terdaftar.	1. Kirim data login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Periksa respons.	Ada "accessToken" dan "refreshToken".
	TC-LOGIN-10	Login psikolog - cookie tersimpan. Kondisi awal: Akun terdaftar.	1. Kirim data login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Periksa data pada cookie.	Cookie berisi "accessToken" dan "refreshToken".
	TC-INT-LOGIN-001	Login - session tercatat. Kondisi awal: Akun terdaftar, belum ada sesi.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Periksa data di basis data sesi.	Sesi baru tercatat.
	TC-INT-LOGIN-002	Login ganda. Kondisi awal: Akun terdaftar.	1. Login dua kali dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Periksa sesi.	Dua sesi.
	TC-INT-LOGIN-003	Body kosong. Kondisi awal: Tidak ada data.	1. Kosongkan seluruh body (tanpa email dan kata sandi). 2. Kirim data ke sistem.	Permintaan ditolak (400).
	TC-INT-LOGIN-004	Tanpa email. Kondisi awal: Tidak ada data.	1. Kirim hanya kata sandi "Valid123!" tanpa menyertakan email.	Permintaan ditolak (400).
	TC-INT-LOGIN-005	Tanpa kata sandi. Kondisi awal: Tidak ada data.	1. Kirim hanya email "psikolog@mail.com" tanpa menyertakan kata sandi.	Permintaan ditolak (400).
	TC-INT-LOGIN-006	User belum verifikasi. Kondisi awal: Akun dengan "verified": false.	1. Login dengan email akun psikolog yang belum diverifikasi dan kata sandi akun tersebut.	Permintaan berhasil (200).

Tabel 4.19 Test Case Login Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-01	TC-INT-LOGIN-007	Akun dinonaktifkan. Kondisi awal: Akun dinonaktifkan admin.	1. Login dengan email akun psikolog yang dinonaktifkan dan kata sandi akun tersebut.	Permintaan ditolak (403).
	TC-INT-LOGIN-008	Akun dihapus. Kondisi awal: Akun dihapus.	1. Login dengan email akun psikolog yang sudah dihapus dan kata sandi akun tersebut.	Permintaan ditolak (404).
	TC-INT-LOGIN-009	Email dengan spasi. Kondisi awal: Akun terdaftar.	1. Kirim email "psikolog@mail.com " dan kata sandi "Valid123!".	Permintaan berhasil (200).
	TC-INT-LOGIN-010	Kata sandi dengan spasi. Kondisi awal: Akun terdaftar.	1. Kirim email "psikolog@mail.com" dan kata sandi " Valid123! ".	Permintaan berhasil (200).
	TC-INT-LOGIN-011	Login setelah register. Kondisi awal: Akun baru dibuat.	1. Login dengan email dan kata sandi akun psikolog yang baru dibuat.	Permintaan berhasil (200).
	TC-INT-LOGIN-012	Login berkali-kali. Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!" 5 kali.	Semua berhasil (200).
	TC-INT-LOGIN-013	Email huruf besar. Kondisi awal: Akun dengan email campuran kapital.	1. Login dengan email "Psikolog@Mail.com" dan kata sandi "Valid123!".	Permintaan berhasil (200).
	TC-INT-LOGIN-014	Login setelah logout. Kondisi awal: Sesi sebelumnya dihapus.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!" setelah logout.	Permintaan berhasil (200). Sesi baru.
	TC-INT-LOGIN-015	Data palsu. Kondisi awal: Tidak ada akun.	1. Kirim email "fake@mail.com" dan kata sandi "Fake123!".	Permintaan ditolak (401).
	TC-INT-LOGIN-016	Status sukses. Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Periksa status.	Status "success".
	TC-INT-LOGIN-017	Data user sesuai. Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Periksa data.	Data sesuai basis data.

Tabel 4.19 Test Case Login Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-01	TC-INT-LOGIN-018	Tidak mengubah data. Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Bandingkan data.	Data tidak berubah.
	TC-INT-LOGIN-019	Token akses valid. Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Gunakan token.	Token valid.
	TC-INT-LOGIN-020	Cookie dibersihkan saat <i>logout</i> . Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Logout. 3. Periksa data pada cookie.	Cookie dibersihkan.
	TC-INT-LOGIN-021	Header JSON. Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Periksa header.	Content-Type: "application/json".
	TC-INT-LOGIN-022	Kata sandi minimal. Kondisi awal: Akun dengan sandi 8 karakter.	1. Login dengan email akun psikolog yang memiliki kata sandi 8 karakter dan kata sandi tersebut.	Permintaan berhasil (200).
	TC-INT-LOGIN-023	Data palsu - tidak buat sesi. Kondisi awal: Akun palsu.	1. Kirim email "fake@mail.com" dan kata sandi "Fake123!". 2. Periksa sesi.	Tidak ada sesi baru.
	TC-INT-LOGIN-024	Login setelah <i>logout</i> lalu login lagi - cookie baru. Kondisi awal: Akun terdaftar.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Logout. 3. Login lagi.	Cookie baru dibuat.

Tabel 4.19 menyajikan *test case login* pada psikolog yang dilakukan untuk memastikan proses autentikasi dapat berjalan dengan benar sesuai alur yang telah dirancang. Pengujian ini mencakup validasi kredensial, penanganan berbagai kondisi *input*, serta respons sistem terhadap skenario berhasil maupun gagal *login*. Dengan demikian, fungsi autentikasi dapat dipastikan mampu memberikan respons yang sesuai dengan kebutuhan sistem dan menjaga keamanan akses pengguna.

Tabel 4. 20 Test Case Logout Psikolog

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-01	TC-LOGOUT-01	Keluar tanpa kode akses. Kondisi awal: User sudah masuk.	1. Kirim permintaan logout tanpa token akses (Authorization header) dan tanpa cookie.	TC-LOGOUT-01
	TC-LOGOUT-02	Keluar dengan kode akses tidak valid. Kondisi awal: User sudah masuk.	1. Kirim permintaan logout dengan token akses "invalid" pada Authorization header.	TC-LOGOUT-02
	TC-LOGOUT-03	Keluar dengan kode akses kadaluwarsa. Kondisi awal: Token sengaja dibuat kadaluwarsa.	1. Buat token akses yang sudah kadaluwarsa. 2. Kirim permintaan logout dengan token kadaluwarsa pada Authorization header.	TC-LOGOUT-03
	TC-LOGOUT-04	Keluar melalui kode akses berhasil. Kondisi awal: Sesi aktif ada di basis data.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!" untuk mendapat token akses. 2. Kirim permintaan logout dengan token akses pada Authorization header. 3. Periksa data di basis data.	TC-LOGOUT-04
	TC-LOGOUT-05	Keluar melalui cookie berhasil. Kondisi awal: Sesi aktif ada di basis data.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!" untuk mendapat cookie. 2. Kirim permintaan logout dengan cookie (tanpa Authorization header).	TC-LOGOUT-05
	TC-INT-LOGOUT-001	Keluar ketika sesi sudah tidak ada. Kondisi awal: Sesi dihapus manual dari basis data.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Hapus sesi manual dari basis data. 3. Kirim permintaan logout dengan token akses.	Permintaan ditolak (401). Sesi tidak ditemukan.
	TC-INT-LOGOUT-002	Keluar menghapus sesi dari basis data. Kondisi awal: Sesi aktif.	1. Login dengan "psikolog@mail.com" dan kata sandi "Valid123!". 2. Catat ID sesi. 3. Logout 4. Cari sesi di basis data.	Sesi tidak ditemukan.

Tabel 4.20 Test Case Logout Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-01	TC-INT-LOGOUT-003	Cookie dibersihkan setelah keluar. Kondisi awal: Sesi aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Logout melalui cookie. 3. Periksa header respons Set-Cookie.	Cookie kode akses dan penyegar kosong.
	TC-INT-LOGOUT-004	Keluar dua kali dengan kode sama. Kondisi awal: Sesi aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses (pertama). 3. Logout lagi dengan token akses yang sama (kedua).	Pertama berhasil (200). Kedua ditolak (401).
	TC-INT-LOGOUT-005	Kode akses tanpa data valid. Kondisi awal: Tidak ada sesi.	1. Buat token akses tanpa data payload (sid, uid). 2. Kirim permintaan logout dengan token tersebut.	Permintaan ditolak (401).
	TC-INT-LOGOUT-006	Kode akses tanpa ID sesi. Kondisi awal: Tidak ada sesi.	1. Buat token akses hanya dengan ID user (tanpa sid). 2. Kirim permintaan logout.	Permintaan ditolak (401).
	TC-INT-LOGOUT-007	Header kode akses tanpa awalan Bearer. Kondisi awal: Sesi aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim header Authorization dengan nilai token akses langsung (tanpa awalan "Bearer ").	Permintaan ditolak (401). Format salah.
	TC-INT-LOGOUT-008	User dihapus tetapi keluar tetap berhasil. Kondisi awal: Sesi aktif, user dihapus.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Hapus user dari basis data. 3. Kirim permintaan logout dengan token akses.	Permintaan berhasil (200).
	TC-INT-LOGOUT-009	Multiple sesi - hanya satu terhapus saat keluar. Kondisi awal: Dua sesi aktif untuk user sama.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!" dua kali untuk mendapat dua sesi. 2. Logout dengan token akses sesi pertama. 3. Periksa data di basis data.	Sesi pertama hilang, sesi kedua masih ada.
	TC-INT-LOGOUT-010	Cookie dan header kode akses dikirim bersamaan.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim logout dengan	Permintaan berhasil (200).

Tabel 4.20 Test Case Logout Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-01		Kondisi awal: Sesi aktif.	cookie dan Authorization header bersamaan.	
	TC-INT-LOGOUT-011	Keluar - status sukses. Kondisi awal: Sesi aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses. 3. Periksa status.	Status "success".
	TC-INT-LOGOUT-012	Keluar - pesan sesuai. Kondisi awal: Sesi aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses. 3. Periksa pesan.	Pesan "Logout successful".
	TC-INT-LOGOUT-013	Keluar tanpa masuk sebelumnya. Kondisi awal: Tidak ada sesi.	1. Kirim permintaan logout tanpa token akses.	Permintaan ditolak (401).
	TC-INT-LOGOUT-014	Keluar - kode akses tidak bisa dipakai lagi. Kondisi awal: Sesi aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses. 3. Coba akses fitur dengan token akses lama.	Kode akses tidak valid.
	TC-INT-LOGOUT-015	Keluar - data user tidak berubah. Kondisi awal: Sesi aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Catat data user. 3. Logout dengan token akses. 4. Bandingkan data.	Data user tidak berubah.

Tabel 4.20 menyajikan *test case* untuk fitur logout pada pengguna dengan hak akses psikolog. Pengujian yang dilakukan berfokus pada pengelolaan sesi (*session management*) serta validitas proses autentikasi ketika pengguna mengakhiri akses ke sistem. Setiap skenario dirancang untuk memastikan bahwa proses logout dapat berjalan dengan benar, sesi pengguna berhasil diakhiri, dan token autentikasi tidak lagi dapat digunakan untuk mengakses fitur yang memerlukan proses autentikasi.

Tabel 4. 21 Test Case Kelola Profil Psikolog

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-02	TC-PSI-UP-01	Ubah profil tanpa kode akses. Kondisi awal: User psikolog terdaftar.	1. Kirim data perubahan (nama "New Name", description "New desc", specialization "Anxiety", educationBackground ["S1"]) tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-PSI-UP-02	Ubah profil dengan kode tidak valid. Kondisi awal: User psikolog terdaftar.	1. Kirim data perubahan (nama "New Name") dengan token akses "invalid" pada Authorization header.	Permintaan ditolak (401).
	TC-PSI-UP-03	Ubah profil sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan (nama "New Name", description "New desc") untuk profil psikolog.	Permintaan ditolak (403).
	TC-PSI-UP-04	Ubah profil sebagai admin ditolak. Kondisi awal: Admin login.	1. Login menggunakan akun admin (email "admin@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan (nama "New Name", description "New desc") untuk profil psikolog.	Permintaan ditolak (403).
	TC-PSI-UP-05	Ubah profil - user sudah dihapus. Kondisi awal: User psikolog dihapus setelah login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Hapus user psikolog dari basis data. 3. Kirim data perubahan (nama "New Name").	Permintaan ditolak (401).
	TC-PSI-UP-06	Ubah dengan data kosong. Kondisi awal: Psikolog login, kode valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kosongkan seluruh data body (tanpa nama, description, specialization, educationBackground). 3. Kirim data ke sistem.	Permintaan ditolak (400).
	TC-PSI-UP-07	Ubah dengan educationBackground bukan array. Kondisi awal:	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim educationBackground:	Permintaan ditolak (400).

Tabel 4.21 Test Case Kelola Profil Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-02		Psikolog login.	"invalid" (string, bukan array).	
	TC-PSI-UP-08	Ubah dengan field tidak dikenal. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim body {"unknown": "field"} tanpa field valid lainnya.	Permintaan ditolak (400).
	TC-PSI-UP-09	Ubah nama kosong - diabaikan. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "" (string kosong), description "New desc".	Permintaan berhasil (200). Nama tidak berubah.
	TC-PSI-UP-10	Ubah description kosong - berhasil. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim description "" (string kosong).	Permintaan berhasil (200). Description kosong.
	TC-PSI-UP-11	Ubah specialization kosong - berhasil. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim specialization "" (string kosong).	Permintaan berhasil (200).
	TC-PSI-UP-12	Ubah educationBackground array kosong. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim educationBackground [].	Permintaan berhasil (200).
	TC-PSI-UP-13	Ubah nama berhasil. Kondisi awal: Psikolog dengan nama "Old Name", kode valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "New Name". 3. Periksa data di basis data.	Permintaan berhasil (200). Nama berubah.
	TC-PSI-UP-14	Ubah description berhasil. Kondisi awal: Psikolog login, kode valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim description "Desc". 3. Periksa data di basis data.	Permintaan berhasil (200). Description berubah.
	TC-PSI-UP-15	Ubah specialization berhasil. Kondisi awal:	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim specialization	Permintaan berhasil (200). Specialization berubah.

Tabel 4.21 Test Case Kelola Profil Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-02		Psikolog <i>login</i> , kode valid.	"Anxiety". 3. Periksa data di basis data.	
	TC-INT-PSI-UP-001	Ubah educationBackground berhasil. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim educationBackground ["S1"]. 3. Periksa data di basis data.	Permintaan berhasil (200).
	TC-INT-PSI-UP-002	Ubah foto melalui unggah file. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Unggah file "test.png" sebagai foto profil.	Permintaan berhasil (200). Path foto ada.
	TC-INT-PSI-UP-003	Ganti foto - path berubah. Kondisi awal: Psikolog sudah punya foto.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Unggah foto lama "old.png". 3. Unggah foto baru "new.png". 4. Bandingkan path.	Path berbeda.
	TC-INT-PSI-UP-004	Ubah beberapa field sekaligus. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "New Name", description "New desc", specialization "Anxiety", educationBackground ["S1 Psikologi"]. 3. Periksa data di basis data.	Semua field berubah.
	TC-INT-PSI-UP-005	Ubah dua kali dengan data sama (idempotent). Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "Same" (pertama). 3. Kirim nama "Same" lagi (kedua).	Kedua berhasil (200). Nama tetap.
	TC-INT-PSI-UP-006	Ubah - file bukan gambar ditolak. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Unggah file "test.txt" (bukan gambar).	Permintaan ditolak (400).

Tabel 4.21 Test Case Kelola Profil Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-02	TC-INT-PSI-UP-007	Ubah - foto terlalu besar. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Unggah foto berukuran 10 MB.	Permintaan ditolak (400).
	TC-INT-PSI-UP-008	Ubah - format respons sesuai. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "New Name". 3. Periksa respons.	Data terupdate dalam respons.
	TC-INT-PSI-UP-009	Ubah - hanya mengirim field yang bisa diubah. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim email baru "new@mail.com" (field yang tidak bisa diubah).	Permintaan ditolak (400).
	TC-INT-PSI-UP-010	Ubah - status sukses. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "New Name". 3. Periksa status.	Status "success".
	TC-INT-PSI-UP-011	Ubah - data tetap setelah perubahan. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Ubah nama jadi "New Name". 3. Periksa data di basis data.	Data sesuai perubahan.
	TC-INT-PSI-UP-012	Ubah - field asing diabaikan. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "New Name" dan field asing "unknown": "field".	Field valid (nama) berubah.
	TC-INT-PSI-UP-013	Ubah - konsisten setelah <i>login</i> ulang. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Ubah nama jadi "New Name". 3. <i>Logout</i> . 4. <i>Login</i> lagi dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 5. Periksa profil.	Nama baru masih ada.

Tabel 4.21 Test Case Kelola Profil Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-02	TC-INT-PSI-UP-014	Ubah - description dengan karakter panjang. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim description 500 karakter.	Permintaan berhasil (200).
	TC-INT-PSI-UP-015	Ubah - specialization dengan karakter khusus. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim specialization "Anxiety & Stress".	Permintaan berhasil (200).
	TC-INT-PSI-UP-016	Ubah - educationBackground dengan beberapa item. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim educationBackground ["S1 Psikologi", "S2 Profesi"].	Permintaan berhasil (200).
	TC-INT-PSI-UP-017	Ubah - tanpa token (401). Kondisi awal: User psikolog terdaftar.	1. Kirim data perubahan (nama "New Name") tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-INT-PSI-UP-018	Ubah - role mahasiswa (403). Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data perubahan (nama "New Name") untuk profil psikolog.	Permintaan ditolak (403).
	TC-INT-PSI-UP-019	Ubah - nama hanya spasi ditolak. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama " " (hanya spasi).	Permintaan ditolak (400).
	TC-INT-PSI-UP-020	Ubah - foto opsional. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim nama "New Name" tanpa menyertakan foto.	Permintaan berhasil (200). Foto tidak berubah.

Berdasarkan Tabel 4.21, skenario pengujian pada modul kelola profil

dirancang secara khusus untuk memverifikasi bahwa seluruh fungsi yang berkaitan

dengan perubahan data profil hanya dapat diakses dan dijalankan oleh aktor psikolog sesuai dengan hak akses yang telah ditetapkan. Pengujian mencakup proses pembaruan informasi profil, seperti nama, deskripsi, spesialisasi, latar belakang pendidikan, dan foto profil, serta validasi terhadap berbagai kondisi *input* yang diberikan.

Tabel 4. 22 Test Case Konsultasi Psikolog

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-03	CONS-PSI-01	Lihat notifikasi tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan notifikasi tanpa token akses (Authorization header).	Permintaan ditolak (401).
	CONS-PSI-02	Lihat notifikasi sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan notifikasi konsultasi psikolog.	Permintaan ditolak (403).
	CONS-PSI-03	Tidak ada konsultasi - notifikasi kosong. Kondisi awal: Psikolog login, tidak ada konsultasi.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan notifikasi konsultasi.	Permintaan berhasil (200). Data kosong.
	CONS-PSI-05	Hanya data milik psikolog login. Kondisi awal: 2 konsultasi (1 milik psikolog A, 1 milik psikolog B).	1. Login sebagai psikolog A (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan notifikasi konsultasi.	Hanya 1 konsultasi (milik A).
	CONS-PSI-06	Format notifikasi sesuai. Kondisi awal: 1 konsultasi.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan notifikasi konsultasi. 3. Periksa format.	Setiap item: consultationId, user, message, status.
	CONS-PSI-07	Update status tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim update status konsultasi (ID valid, status "accepted") tanpa token akses (Authorization header).	Permintaan ditolak (401).

Tabel 4.22 Test Case Konsultasi Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-03	CONS-PSI-08	Update status sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa <i>login</i> .	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim update konsultasi	Permintaan ditolak (403).
	CONS-PSI-09	Update dengan ID tidak valid. Kondisi awal: Psikolog <i>login</i> .	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update dengan consultationId "123" (format salah) dan status "accepted".	Permintaan ditolak (400).
	CONS-PSI-10	Update konsultasi tidak ditemukan. Kondisi awal: ID tidak ada.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update dengan consultationId palsu (ObjectId valid tapi tidak ada) dan status "accepted".	Permintaan ditolak (404).
	CONS-PSI-11	Update konsultasi milik psikolog lain. Kondisi awal: Konsultasi milik psikolog B.	1. <i>Login</i> sebagai psikolog A (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim update dengan consultationId milik B dan status "accepted".	Permintaan ditolak (403).
	CONS-PSI-12	Update dengan status tidak valid ("pending"). Kondisi awal: Status konsultasi pending.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim status "pending" (bukan "accepted" atau "rejected").	Permintaan ditolak (400).
	CONS-PSI-13	Update konsultasi yang bukan pending (sudah diterima). Kondisi awal: Status konsultasi "accepted".	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim status "rejected" pada konsultasi yang sudah "accepted".	Permintaan ditolak (400).
	CONS-PSI-14	Accept konsultasi berhasil. Kondisi awal: Konsultasi pending, room inactive.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim status "accepted". 3. Periksa basis data.	Permintaan berhasil (200). Status jadi "accepted", room jadi "active".
	CONS-PSI-15	Reject konsultasi berhasil.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!".	Permintaan berhasil (200). Status jadi

Tabel 4.22 Test Case Konsultasi Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-03		Kondisi awal: Konsultasi pending, room inactive.	2. Kirim status "rejected". 3. Periksa basis data.	"rejected", room tetap "inactive".
	CONS-PSI-16	Accept tanpa chatRoom - tidak crash. Kondisi awal: Konsultasi pending tanpa chatRoom.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim status "accepted".	Permintaan berhasil (200). Sistem tidak error.
	CONS-PSI-17	Reject - tidak ada perubahan di room. Kondisi awal: Konsultasi pending, room inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim status "rejected". 3. Periksa room.	Room tetap inactive.
	CONS-PSI-18	Accept - room dibuat jika belum ada. Kondisi awal: Konsultasi tanpa room.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim status "accepted". 3. Periksa room.	Room baru dibuat dengan status active.
	TC-INT-CONS-PSI-001	Lihat notifikasi tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan notifikasi konsultasi tanpa token akses.	Permintaan ditolak (401).
	TC-INT-CONS-PSI-002	Lihat notifikasi sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan notifikasi konsultasi dengan token mahasiswa.	Permintaan ditolak (403).
	TC-INT-CONS-PSI-003	Tidak ada konsultasi - notifikasi kosong. Kondisi awal: Psikolog login, tidak ada konsultasi.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan notifikasi konsultasi.	Permintaan berhasil (200). Data = [].
	TC-INT-CONS-PSI-004	Ada konsultasi - data sesuai DB. Kondisi awal: 3 konsultasi di DB.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan notifikasi konsultasi.	Permintaan berhasil (200). jumlah data konsultasi sesuai dengan basis data.
	TC-INT-CONS-PSI-005	Data hanya milik psikolog login. Kondisi awal: 2	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!".	Permintaan berhasil (200). hanya data milik

Tabel 4.22 Test Case Konsultasi Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-03		konsultasi (1 milik psikolog ini, 1 milik lain).	2. Kirim permintaan notifikasi konsultasi.	psikolog yang login yang ditampilkan.
	TC-INT-CONS-PSI-006	Format notifikasi sesuai. Kondisi awal: 1 konsultasi di DB.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan notifikasi konsultasi. 3. Periksa format.	Item punya consultationId, user_id, user.email, message, status.
	TC-INT-CONS-PSI-007	Update status tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim update status konsultasi tanpa token akses.	Permintaan ditolak (401).
	TC-INT-CONS-PSI-008	Update status sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim update status konsultasi dengan token mahasiswa.	Permintaan ditolak (403).
	TC-INT-CONS-PSI-009	Update dengan ID tidak valid. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi.	Permintaan ditolak (400).
	TC-INT-CONS-PSI-010	Update konsultasi tidak ditemukan. Kondisi awal: ID tidak ada di DB.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi.	Permintaan ditolak (404).
	TC-INT-CONS-PSI-011	Update konsultasi milik psikolog lain. Kondisi awal: Konsultasi milik psikolog lain.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi dengan consultationId milik psikolog lain.	Permintaan ditolak (403).
	TC-INT-CONS-PSI-012	Update dengan status tidak valid. Kondisi awal: Konsultasi pending.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi dengan status "pending".	Permintaan ditolak (400).
	TC-INT-CONS-PSI-013	Update konsultasi yang bukan pending. Kondisi awal: Status	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status	Permintaan ditolak (400).

Tabel 4.22 *Test Case* Konsultasi Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-03		konsultasi "accepted".	konsultasi dengan status "rejected".	
	TC-INT-CONS-PSI-014	Accept konsultasi berhasil. Kondisi awal: Konsultasi pending, room inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi dengan status "accepted". 3. Periksa basis data.	Permintaan berhasil (200). Status jadi "accepted", room jadi "active".
	TC-INT-CONS-PSI-015	Reject konsultasi berhasil. Kondisi awal: Konsultasi pending, room inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi dengan status "rejected". 3. Periksa basis data.	Permintaan berhasil (200). Status jadi "rejected", room tetap "inactive".
	TC-INT-CONS-PSI-016	Accept tanpa chatRoom - tidak crash. Kondisi awal: Konsultasi pending tanpa chatRoom.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi dengan status "accepted".	Permintaan berhasil (200). Sistem tidak error.
	TC-INT-CONS-PSI-017	Response structure sesuai. Kondisi awal: Konsultasi pending.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim update status konsultasi dengan status "accepted".	Permintaan berhasil (200). respons berisi pesan konfirmasi dan ID konsultasi.
	TC-INT-CONS-PSI-018	Race condition - hanya satu berhasil. Kondisi awal: Konsultasi pending.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim 2 request accept bersamaan.	satu permintaan berhasil (200), satu permintaan ditolak (400).

Tabel 4.22 menyajikan *test case* untuk fitur konsultasi pada pengguna dengan hak akses psikolog. Pengujian yang dilakukan difokuskan pada validasi dua aspek utama, yaitu kebenaran logika bisnis dalam proses menerima maupun menolak pengajuan konsultasi dari mahasiswa serta terhadap upaya akses yang tidak sah oleh pengguna yang tidak memiliki autentikasi atau hak akses yang sesuai. Setiap *test case* disusun untuk memastikan bahwa sistem mampu memproses setiap tindakan sesuai dengan aturan bisnis yang telah ditetapkan.

Tabel 4. 23 Test Case Chat Psikolog

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-04	TC-PSI-01	Daftar ruang tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan daftar chat room tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-PSI-02	Daftar ruang berhasil. Kondisi awal: Room milik psikolog ada.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar chat room.	Permintaan berhasil (200). Daftar ruang.
	TC-PSI-03	Daftar ruang untuk psikolog baru. Kondisi awal: Psikolog baru tanpa room.	1. Daftar psikolog baru (email "psikolog2@mail.com", password "Valid123!"). 2. Login dengan email "psikolog2@mail.com" dan kata sandi "Valid123!". 3. Kirim permintaan daftar chat room.	Permintaan berhasil (200). Kosong.
	TC-PSI-04	Hanya room milik psikolog ini. Kondisi awal: Room milik psikolog lain ada.	1. Login sebagai psikolog A (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan daftar chat room.	Room psikolog B tidak muncul.
	TC-PSI-05	Dapatkan pesan tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan pesan (roomId valid) tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-PSI-06	Dapatkan pesan - room tidak ditemukan. Kondisi awal: Kode valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan pesan dengan roomId palsu (ObjectId valid tapi tidak ada).	Permintaan ditolak (404).
	TC-PSI-07	Dapatkan pesan berhasil. Kondisi awal: Room milik psikolog dengan pesan.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan pesan dengan roomId valid milik psikolog.	Permintaan berhasil (200). Daftar pesan.
	TC-PSI-08	Akses room bukan milik psikolog. Kondisi awal: Room	1. Login sebagai psikolog A (email "psikolog@mail.com", kata sandi "Valid123!").	Permintaan ditolak (403).

Tabel 4.23 Test Case Chat Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-04		milik psikolog lain.	2. Akses <i>chat</i> room milik psikolog B.	
	TC-PSI-09	Kirim pesan tanpa field wajib. Kondisi awal: Kode valid.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kosongkan seluruh data body (tanpa roomId, senderId, message). 3. Kirim data ke sistem.	Permintaan ditolak (400).
	TC-PSI-10	Kirim pesan berhasil. Kondisi awal: Room aktif milik psikolog.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId valid, senderId psikolog, message "Reply from psikolog".	Permintaan berhasil (201).
	TC-PSI-11	Pesan tidak disimpan ke basis data. Kondisi awal: Room ada.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId valid, senderId psikolog, message "Test". 3. Periksa data di basis data.	Pesan tidak tersimpan.
	TC-PSI-12	Kirim pesan terlalu panjang (1001 karakter). Kondisi awal: Room aktif.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId valid, senderId psikolog, message 1001 karakter.	Permintaan ditolak (413).
	TC-PSI-13	Kirim pesan mengandung skrip HTML. Kondisi awal: Room aktif.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId valid, senderId psikolog, message "<script>alert(1)</script>".	Ditolak (400) atau dibersihkan.
	TC-PSI-14	Tutup room tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan tutup <i>chat</i> room (roomId valid) tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-PSI-15	Tutup room tidak ditemukan. Kondisi awal: Kode valid.	1. <i>Login</i> dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup dengan roomId palsu	Permintaan ditolak (404).

Tabel 4.23 Test Case Chat Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-04			(ObjectId valid tapi tidak ada).	
	TC-PSI-16	Tutup room berhasil. Kondisi awal: Room aktif milik psikolog.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup chat room. 3. Periksa data di basis data.	Permintaan berhasil (200). Status "inactive".
	TC-PSI-17	Tutup room dua kali. Kondisi awal: Room sudah inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup (pertama). 3. Kirim permintaan tutup lagi (kedua).	Kedua berhasil (200). Status tetap.
	TC-PSI-18	Dapatkan pesan setelah room ditutup. Kondisi awal: Room inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Tutup chat room. 3. Kirim permintaan pesan dengan roomId yang sama.	Permintaan berhasil (200). Masih bisa baca.
	TC-PSI-19	Kirim pesan ke room inactive. Kondisi awal: Room inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Tutup chat room. 3. Kirim pesan dengan roomId yang sama, senderId psikolog, message "Test".	Berhasil (201) atau ditolak (400).
	TC-PSI-20	Kirim pesan - senderId tidak cocok. Kondisi awal: Room aktif.	1. Login sebagai psikolog A (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim pesan dengan senderId milik psikolog B (bukan A).	Ditolak (403) atau dipaksa ke A.
	TC-PSI-21	Daftar ruang - urutan aktivitas. Kondisi awal: Beberapa room.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar chat room. 3. Periksa urutan.	Terbaru di atas.
	TC-PSI-22	Kirim pesan hanya spasi. Kondisi awal: Room aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId valid, senderId	Permintaan ditolak (400).

Tabel 4.23 Test Case Chat Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-04			psikolog, message " " (hanya spasi).	
	TC-PSI-23	Kirim pesan - roomId tidak valid. Kondisi awal: Kode valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId "abc" (format salah), senderId psikolog, message "Test".	Permintaan ditolak (400).
	TC-PSI-24	Dapatkan pesan - room tanpa pesan. Kondisi awal: Room baru.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan pesan dengan roomId valid milik psikolog (room baru tanpa pesan).	Daftar kosong.
	TC-PSI-25	Kirim pesan gambar (URL). Kondisi awal: Room aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId valid, senderId psikolog, message "https://example.com/image.png".	Permintaan berhasil (201).
	TC-PSI-26	Kirim pesan dengan emoji. Kondisi awal: Room aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan dengan roomId valid, senderId psikolog, message "😊".	Permintaan berhasil (201).
	TC-PSI-27	Kirim pesan - sesi kadaluwarsa. Kondisi awal: Sesi kadaluwarsa.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Tunggu hingga token akses kadaluwarsa. 3. Kirim pesan dengan roomId valid, senderId psikolog, message "Test".	Permintaan ditolak (401).
	TC-PSI-28	Tutup room - bukan peserta. Kondisi awal: Room milik psikolog A dan mahasiswa.	1. Login sebagai psikolog C (bukan peserta room) dengan email "psikolog2@mail.com" dan kata sandi "Valid123!". 2. Coba tutup room milik A.	Permintaan ditolak (403).
	TC-INT-CHAT-PSI-001	Daftar ruang tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan daftar ruang chat tanpa token akses.	Permintaan ditolak (401).

Tabel 4.23 Test Case Chat Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-04	TC-INT-CHAT-PSI-002	Daftar ruang berhasil. Kondisi awal: Room milik psikolog ada.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar ruang chat dengan token psikolog.	Permintaan berhasil (200). Array tidak kosong.
	TC-INT-CHAT-PSI-003	Daftar ruang untuk psikolog baru. Kondisi awal: Psikolog baru tanpa room.	1. Daftar psikolog baru (email "psikolog2@mail.com", password "Valid123!"). 2. Login dengan email "psikolog2@mail.com" dan kata sandi "Valid123!". 3. Kirim permintaan daftar ruang chat.	Permintaan berhasil (200). Array kosong.
	TC-INT-CHAT-PSI-004	Hanya room milik psikolog ini. Kondisi awal: Room milik psikolog lain ada.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar ruang chat.	Room milik psikolog lain tidak muncul.
	TC-INT-CHAT-PSI-005	Dapatkan pesan tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan pesan chat tanpa token akses.	Permintaan ditolak (401).
	TC-INT-CHAT-PSI-006	Dapatkan pesan - room tidak ditemukan. Kondisi awal: Token psikolog valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan pesan chat.	Permintaan ditolak (404).
	TC-INT-CHAT-PSI-007	Dapatkan pesan berhasil. Kondisi awal: Room milik psikolog dengan pesan.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan pesan chat dengan roomId valid.	Permintaan berhasil (200). Array pesan.
	TC-INT-CHAT-PSI-008	Akses room bukan milik psikolog. Kondisi awal: Room milik psikolog lain.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan pesan chat dengan roomId milik psikolog lain.	Permintaan ditolak (403).

Tabel 4.23 Test Case Chat Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-04	TC-INT-CHAT-PSI-009	Kirim pesan tanpa field wajib. Kondisi awal: Token psikolog valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan chat dengan body {} (tanpa roomId, senderId, message).	Permintaan ditolak (400).
	TC-INT-CHAT-PSI-010	Kirim pesan berhasil. Kondisi awal: Room aktif milik psikolog.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan chat dengan roomId valid, message "Test".	Permintaan berhasil (201).
	TC-INT-CHAT-PSI-011	Pesan tidak disimpan ke basis data. Kondisi awal: Room ada.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan chat dengan message valid. 3. Periksa basis data.	Pesan tidak ada di room.messages di DB.
	TC-INT-CHAT-PSI-012	Kirim pesan terlalu panjang. Kondisi awal: Room aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan chat dengan message 1001 karakter.	Permintaan ditolak (413).
	TC-INT-CHAT-PSI-013	Kirim pesan mengandung skrip HTML. Kondisi awal: Room aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan chat dengan message "<script>alert(1)</script>".	Permintaan ditolak (400).
	TC-INT-CHAT-PSI-014	Tutup room tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan tutup ruang chat tanpa token akses.	Permintaan ditolak (401).
	TC-INT-CHAT-PSI-015	Tutup room tidak ditemukan. Kondisi awal: Token psikolog valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup ruang chat.	Permintaan ditolak (404).
	TC-INT-CHAT-PSI-016	Tutup room berhasil. Kondisi awal: Room aktif milik psikolog.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup ruang chat dengan roomId valid. 3. Periksa basis data.	Permintaan berhasil (200). Status room jadi "inactive".

Tabel 4.23 Test Case Chat Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-04	TC-INT-CHAT-PSI-017	Tutup room dua kali (idempotent). Kondisi awal: Room sudah inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup ruang chat dua kali.	Kedua berhasil (200). Status tetap "inactive".
	TC-INT-CHAT-PSI-018	Dapatkan pesan setelah room ditutup. Kondisi awal: Room inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup ruang chat. 3. Kirim permintaan pesan chat dengan roomId yang sama.	Permintaan berhasil (200). Masih bisa baca.
	TC-INT-CHAT-PSI-019	Kirim pesan ke room yang sudah ditutup. Kondisi awal: Room inactive.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan tutup ruang chat. 3. Kirim pesan chat ke room inactive.	Berhasil (201) atau ditolak (400).
	TC-INT-CHAT-PSI-020	Kirim pesan - senderId tidak cocok. Kondisi awal: Room aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan chat dengan senderId berbeda dari user login.	Ditolak (403) atau senderId dipaksa sesuai token.
	TC-INT-CHAT-PSI-021	Daftar ruang - urutan aktivitas terbaru. Kondisi awal: Beberapa room.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar ruang chat. 3. Periksa urutan.	Aktivitas terbaru di atas.
	TC-INT-CHAT-PSI-022	Kirim pesan hanya spasi. Kondisi awal: Room aktif.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pesan chat dengan message " " (hanya spasi).	Permintaan ditolak (400).

Tabel 4.23 menyajikan rincian *test case* untuk fitur layanan pesan (*chat*) pada hak akses psikolog. Pengujian difokuskan pada verifikasi proses pemuatan daftar percakapan, pembukaan ruang *chat*, serta pengiriman dan penerimaan pesan antara psikolog dan mahasiswa.

Tabel 4. 24 Test Case Kelola Artikel Psikolog

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-05	TC-PSI-ART-01	Buat artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim data artikel baru (judul "Test", konten "Test content") tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-PSI-ART-02	Update artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim data artikel (judul "Updated", konten "Updated content") tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-PSI-ART-03	Hapus artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan hapus artikel (articleId valid) tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-PSI-ART-04	Buat artikel tanpa judul. Kondisi awal: Psikolog login, kode valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim data artikel dengan konten "Test content" tanpa menyertakan judul.	Permintaan ditolak (400). Judul wajib.
	TC-PSI-ART-05	Buat artikel tanpa konten. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim data artikel dengan judul "Test" tanpa menyertakan konten.	Permintaan ditolak (400). Konten wajib.
	TC-PSI-ART-06	Buat artikel dengan data valid. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Hello", konten "World".	Permintaan berhasil (201). Artikel tersimpan.
	TC-PSI-ART-07	Update artikel milik psikolog lain. Kondisi awal: Artikel milik psikolog B, psikolog A login.	1. Login sebagai psikolog A (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Kirim pembaruan (judul "Hacked", konten "Hacked content") untuk artikel milik psikolog B.	Permintaan ditolak (403).
	TC-PSI-ART-08	Hapus artikel milik psikolog lain. Kondisi awal: Artikel milik psikolog B, psikolog A login.	1. Login sebagai psikolog A (email "psikolog@mail.com", kata sandi "Valid123!"). 2. Hapus artikel milik psikolog B.	Permintaan ditolak (403).

Tabel 4.24 Test Case Kelola Artikel Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-05	TC-PSI-ART-09	Update artikel tidak ditemukan. Kondisi awal: Psikolog login, ID palsu.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pembaruan (judul "New") dengan articleId palsu (ObjectId valid tapi tidak ada).	Permintaan ditolak (404).
	TC-PSI-ART-10	Update artikel milik sendiri berhasil. Kondisi awal: Artikel milik psikolog ini ada.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul baru "New".	Permintaan berhasil (200). Judul berubah.
	TC-PSI-ART-11	Update judul - slug berubah. Kondisi awal: Artikel dengan slug "old-title".	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "New Title". 3. Periksa respons.	Slug jadi "new-title".
	TC-PSI-ART-12	Update konten saja. Kondisi awal: Artikel milik sendiri.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim konten "New Content".	Permintaan berhasil (200). Konten berubah.
	TC-PSI-ART-13	Hapus artikel milik sendiri berhasil. Kondisi awal: Artikel milik sendiri.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Hapus artikel dengan articleId valid milik sendiri.	Permintaan berhasil (200). Artikel terhapus.
	TC-PSI-ART-14	Slug otomatis saat buat. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Hello World", konten "Content". 3. Periksa respons.	Slug jadi "hello-world".
	TC-PSI-ART-15	Slug berubah saat judul diupdate. Kondisi awal: Artikel milik sendiri.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim pembaruan judul "Updated Title". 3. Periksa slug.	Slug jadi "updated-title".
	TC-PSI-ART-16	Format respons valid. Kondisi awal:	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Hello", konten	Ada field: id, title, slug, writer.

Tabel 4.24 Test Case Kelola Artikel Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-05		Psikolog login.	"World". 3. Periksa respons.	
	TC-PSI-ART-17	Buat artikel - judul dan konten minimal. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "A", konten "B".	Permintaan berhasil (201).
	TC-PSI-ART-18	Update artikel - data kosong. Kondisi awal: Artikel milik sendiri.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kosongkan seluruh data body (tanpa judul dan konten). 3. Kirim data ke sistem.	Permintaan berhasil (200). Data tidak berubah.
	TC-PSI-ART-19	Hapus artikel dua kali. Kondisi awal: Artikel ada lalu dihapus.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Hapus artikel milik sendiri (pertama). 3. Hapus artikel yang sama lagi (kedua).	Pertama berhasil (200). Kedua ditolak (404).
	TC-PSI-ART-20	Buat artikel - tanpa psikolog login (token mahasiswa). Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim judul "Test", konten "Test content".	Permintaan ditolak (403).
	TC-PSI-ART-21	Buat artikel - judul sangat panjang. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul 200 karakter, konten "Test content".	Permintaan berhasil (201) atau ditolak (400).
	TC-PSI-ART-22	Update artikel - judul sangat panjang. Kondisi awal: Artikel milik sendiri.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim judul 200 karakter.	Berhasil (200) atau ditolak (400).
	TC-INT-PSI-01	Update artikel milik psikolog lain ditolak. Kondisi awal: Artikel milik psikolog lain.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan articleId milik psikolog lain.	Permintaan ditolak (401).

Tabel 4.24 Test Case Kelola Artikel Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-05	TC-INT-PSI-02	Hapus artikel milik psikolog lain ditolak. Kondisi awal: Artikel milik psikolog lain.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus artikel dengan articleId milik psikolog lain.	Permintaan ditolak (401).
	TC-INT-PSI-03	Update artikel tidak ditemukan. Kondisi awal: Token psikolog valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel.	Permintaan ditolak (401 atau 404).
	TC-INT-PSI-04	Hapus artikel tidak ditemukan. Kondisi awal: Token psikolog valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus artikel.	Permintaan ditolak (401 atau 404).
	TC-INT-PSI-05	Buat artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan buat artikel tanpa token akses.	Permintaan ditolak (401).
	TC-INT-PSI-06	Update artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan update artikel tanpa token akses.	Permintaan ditolak (401).
	TC-INT-PSI-07	Hapus artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan hapus artikel tanpa token akses.	Permintaan ditolak (401).
	TC-INT-PSI-08	Buat artikel tanpa judul. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel dengan content saja (tanpa title).	Permintaan ditolak (400).
	TC-INT-PSI-09	Buat artikel tanpa konten. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel dengan title saja (tanpa content).	Permintaan ditolak (400).
	TC-INT-PSI-10	Update artikel dengan ID tidak valid.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update	Permintaan ditolak (400).

Tabel 4.24 Test Case Kelola Artikel Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-05		Kondisi awal: Psikolog login.	artikel.	
	TC-INT-PSI-11	Hapus artikel dua kali. Kondisi awal: Artikel milik psikolog ada.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus artikel (pertama). 3. Kirim permintaan hapus artikel lagi (kedua).	Pertama berhasil (200). Kedua ditolak (401 atau 404).
	TC-INT-PSI-12	Lihat artikel by ID tidak ditemukan. Kondisi awal: Token admin valid.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat artikel.	Permintaan ditolak (404).
	TC-INT-PSI-14	Update artikel milik sendiri berhasil. Kondisi awal: Artikel milik psikolog ini.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan title baru.	Permintaan berhasil (200).
	TC-INT-PSI-15	Update judul - slug berubah. Kondisi awal: Artikel milik psikolog ini.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan title "New Title".	Permintaan berhasil (200). slug artikel berubah menjadi new-title.
	TC-INT-PSI-16	Update konten saja. Kondisi awal: Artikel milik psikolog ini.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan content baru.	Permintaan berhasil (200).
	TC-INT-PSI-17	Hapus artikel milik sendiri berhasil. Kondisi awal: Artikel milik psikolog ini.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus artikel.	Permintaan berhasil (200).
	TC-INT-PSI-18	Lihat artikel by ID valid. Kondisi awal: Artikel ada di DB.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat artikel dengan articleId valid.	Permintaan berhasil (200).

Tabel 4.24 *Test Case* Kelola Artikel Psikolog (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-PSI-05	TC-INT-PSI-19	Lihat semua artikel. Kondisi awal: Artikel ada di DB.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar artikel.	Permintaan berhasil (200). data adalah array.
	TC-INT-PSI-20	Slug otomatis saat buat. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel dengan title "Hello World".	slug artikel terbentuk menjadi hello-world.
	TC-INT-PSI-21	Slug berubah saat judul diupdate. Kondisi awal: Artikel milik psikolog ada.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan title "Updated Title".	slug artikel berubah menjadi updated-title.
	TC-INT-PSI-22	Format respons valid. Kondisi awal: Psikolog login.	1. Login dengan email "psikolog@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel dengan title dan content. 3. Periksa format.	Respons punya id, title, slug, writer.

Tabel 4.24 menguraikan *test case* untuk modul pengelolaan artikel oleh aktor Psikolog. Pengujian ini berfokus pada validasi keberhasilan penambahan data, penanganan integritas kelengkapan *input*, serta proteksi kepemilikan artikel dari upaya modifikasi oleh pihak yang tidak berwenang.

4. Requirement Traceability Matrix (RTM) Psikolog

Requirement Traceability Matrix (RTM) Psikolog digunakan untuk menunjukkan keterlacakan antara setiap kebutuhan fungsional yang telah didefinisikan dengan skenario pengujian yang dilakukan.

Tabel 4. 25 RTM Psikolog

Scenario ID	Requirements Description	Test Case ID
TS-PSI-01 (Autentikasi)	Psikolog dapat login ke sistem	TC-LOGIN-01 s.d. TC-LOGIN-10, TC-INT-LOGIN-001 s.d. TC-INT-LOGIN-024
TS-PSI-01	Psikolog dapat logout dari sistem	TC-LOGOUT-01 s.d. TC-

Tabel 4.25 RTM Psikolog (Lanjutan)

<i>Scenario ID</i>	<i>Requirements Description</i>	<i>Test Case ID</i>
(Autentikasi)		LOGOUT-05, TC-INT-LOGOUT-001 s.d. TC-INT-LOGOUT-015
TS-PSI-02 (Kelola Profil)	Psikolog dapat mengubah profil (nama, deskripsi, spesialisasi, pendidikan, foto)	TC-PSI-UP-01 s.d. TC-PSI-UP-15, TC-INT-PSI-UP-001 s.d. TC-INT-PSI-UP-020
TS-PSI-03 (Konsultasi)	Psikolog dapat melihat notifikasi konsultasi masuk	CONS-PSI-01 s.d. CONS-PSI-06
TS-PSI-03 (Konsultasi)	Psikolog dapat menerima atau menolak konsultasi	CONS-PSI-07 s.d. CONS-PSI-18, TC-INT-CONS-PSI-001 s.d. TC-INT-CONS-PSI-018
TS-PSI-04 (Chat)	Psikolog dapat melihat daftar ruang <i>chat</i>	TC-PSI-01 s.d. TC-PSI-04, TC-PSI-21
TS-PSI-04 (Chat)	Psikolog dapat mengirim dan menerima pesan <i>chat</i>	TC-PSI-05 s.d. TC-PSI-13, TC-PSI-18 s.d. TC-PSI-20, TC-PSI-22 s.d. TC-PSI-27
TS-PSI-04 (Chat)	Psikolog dapat menutup ruang <i>chat</i>	TC-PSI-14 s.d. TC-PSI-17, TC-PSI-28
TS-PSI-05 (Kelola Artikel)	Psikolog dapat membuat artikel	TC-PSI-ART-01, TC-PSI-ART-04 s.d. TC-PSI-ART-06, TC-PSI-ART-14, TC-PSI-ART-17, TC-PSI-ART-20, TC-PSI-ART-21
TS-PSI-05 (Kelola Artikel)	Psikolog dapat memperbaiki artikel milik sendiri	TC-PSI-ART-02, TC-PSI-ART-07, TC-PSI-ART-09 s.d. TC-PSI-ART-12, TC-PSI-ART-15, TC-PSI-ART-18, TC-PSI-ART-22
TS-PSI-05 (Kelola Artikel)	Psikolog dapat menghapus artikel milik sendiri	TC-PSI-ART-03, TC-PSI-ART-08, TC-PSI-ART-13, TC-PSI-ART-19

Tabel 4.25 menyajikan *Requirement Traceability Matrix* (RTM) untuk modul psikolog yang memetakan setiap kebutuhan fungsional dengan *test case* yang digunakan dalam proses pengujian. Pemetaan tersebut mencakup fitur autentikasi, pengelolaan profil, konsultasi, *chat*, serta pengelolaan artikel.

C. Pengujian Role Admin

1. Main Map Fitur Mahasiswa

Berikut adalah seluruh fitur yang dapat diakses dan digunakan oleh pengguna dengan peran admin dalam sistem SiKosa.

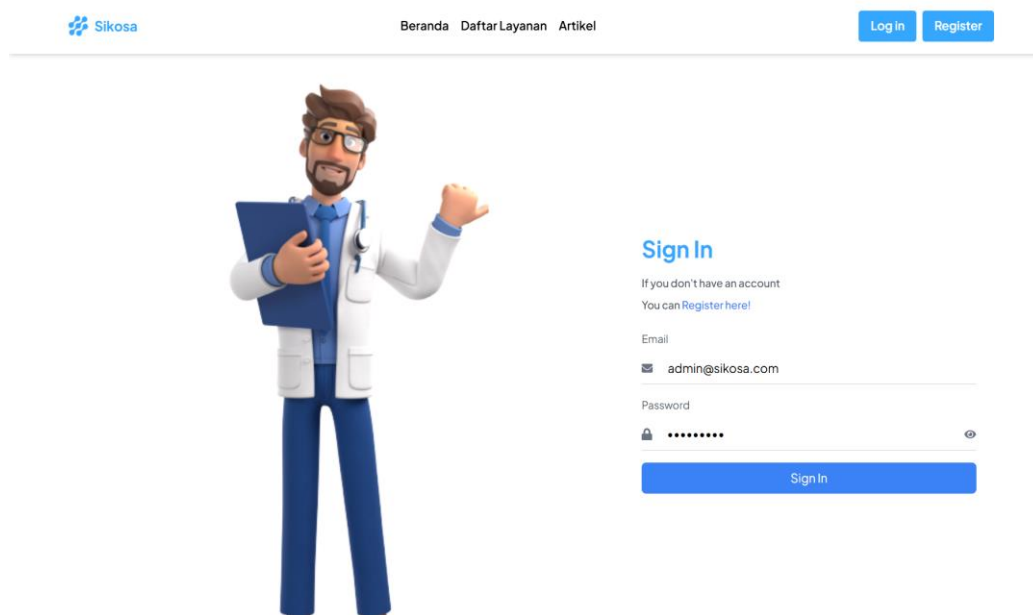
Tabel 4. 26 Main Map Fitur Admin

No	Fitur	Deskripsi
1	Login	Masuk ke sistem menggunakan akun admin
2	Kelola User	Menambah, mengubah, menghapus, dan melihat data seluruh pengguna
3	Kelola Artikel	Membuat, mengubah, menghapus, dan melihat artikel
4	Logout	Keluar dari sistem

Tabel 4.26 menyajikan pemetaan fitur utama (*main map*) untuk aktor Admin. Tabel ini mendeskripsikan fitur yang mendukung keseluruhan tugas administratif dan pengelolaan konten di dalam sistem.

a) Fitur *Login* (Admin)

Fitur *login* (admin) merupakan mekanisme autentikasi yang digunakan oleh pengguna dengan peran admin untuk mengakses sistem.

**Gambar 4. 17 Halaman *login* (admin)**

Gambar 4.17 menampilkan halaman *login* yang digunakan oleh admin untuk masuk ke dalam sistem. Pada halaman ini terdapat *form input* email dan

password yang harus diisi sebelum admin dapat mengakses fitur-fitur yang tersedia di dalam sistem.

b) **Fitur Kelola User (admin)**

Fitur kelola *user* digunakan oleh admin untuk mengelola data pengguna yang terdaftar dalam sistem. Melalui fitur ini, admin dapat melihat daftar *user*, menambahkan *user* baru, mengubah data *user*, serta menghapus akun *user* yang sudah tidak aktif. Fitur ini memastikan data pengguna tetap terorganisir dan sesuai dengan kebutuhan sistem.

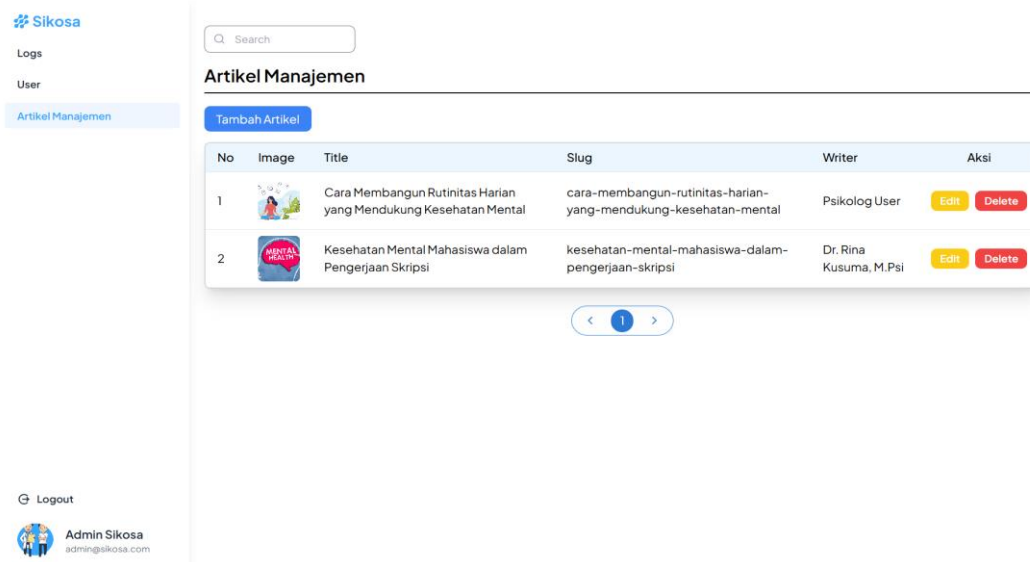
No	Nama	Role	Aksi
1	Mahasiswa User	mahasiswa	Edit Delete
2	Psikolog User	psikolog	Edit Delete
3	Admin User	admin	Edit Delete
4	Muhammad Aidil	mahasiswa	Edit Delete
5	Admin Sikosa	admin	Edit Delete
6	Dr. Rina Kusuma, M.Psi	psikolog	Edit Delete
7	Budi Santoso, M.Psi	psikolog	Edit Delete
8	Andi Pratama	mahasiswa	Edit Delete
9	Sari Dewi	mahasiswa	Edit Delete
10	Rizki Ramadhan	mahasiswa	Edit Delete

Gambar 4. 18 Halaman kelola *user*

Gambar 4.18 menampilkan halaman kelola *user* yang digunakan admin untuk mengelola data pengguna. Pada halaman ini, admin dapat melakukan proses penambahan, pengubahan, dan penghapusan data *user* sesuai dengan hak akses yang telah ditentukan.

c) Fitur Kelola Artikel (Admin)

Fitur kelola artikel (admin) digunakan untuk mengelola seluruh konten artikel yang tersedia dalam sistem. Admin memiliki hak akses untuk menambahkan, mengedit, serta menghapus artikel yang dipublikasikan kepada pengguna.



Gambar 4. 19 Halaman kelola artikel (admin)

Gambar 4.19 menampilkan halaman kelola artikel pada sisi admin yang digunakan untuk mengatur konten artikel. Pada halaman ini admin dapat melakukan pengelolaan artikel secara menyeluruh agar informasi yang tersedia tetap terstruktur dan relevan.

d) Fitur Logout (Admin)

Fitur Logout (Admin) digunakan untuk mengakhiri sesi login admin yang sedang aktif dalam sistem. Sistem akan menghapus sesi autentikasi sehingga akses ke halaman admin tidak dapat dilakukan tanpa login ulang.

 Logout

Admin Sikosa
admin@sikosa.com

Gambar 4. 20 Fitur logout

Gambar 4.20 menampilkan proses *logout* yang dilakukan oleh admin.

Setelah tombol *logout* dipilih, sistem akan mengakhiri sesi pengguna dan mengarahkan kembali ke halaman *login* sebagai langkah keamanan sistem.

2. Test Scenario Admin

Rincian *test scenario* di bawah ini disusun untuk memastikan bahwa seluruh modul pengelola yang dioperasikan oleh aktor Admin dapat berfungsi dengan baik dan sesuai dengan spesifikasi yang diharapkan

Tabel 4. 27 Test Scenario Admin

ID Skenario	Skenario Pengujian
TS-ADM-01	Autentikasi: Proses masuk ke sistem dan keluar dari sistem
TS-ADM-02	Kelola User: Melihat, menambah, mengubah, dan menghapus data pengguna (mahasiswa, psikolog, admin)
TS-ADM-03	Kelola Artikel: Melihat, menambah, mengubah, dan menghapus artikel

Tabel 4.27 memaparkan rincian *test scenario* yang dirancang untuk menguji seluruh fungsionalitas pada modul aktor admin. Skenario pengujian disusun berdasarkan kebutuhan fungsional sistem dan mencakup setiap fitur yang menjadi tanggung jawab admin.

3. Test Case Development Admin

Berikut adalah jabaran *test case* terperinci yang digunakan sebagai pedoman evaluasi fungsional menyeluruh untuk pengguna dengan peran Admin.

Tabel 4. 28 Test Case Login Admin

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-01	TC-LOGIN-01	Login dengan email kosong. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengosongkan kolom email dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Email wajib.
	TC-LOGIN-02	Login dengan kata sandi kosong. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "admin@mail.com" dan mengosongkan kolom kata sandi. 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400). Sandi wajib.
	TC-LOGIN-03	Login dengan format email salah. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "invalid" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400).
	TC-LOGIN-04	Login dengan kata sandi terlalu pendek. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "admin@mail.com" dan mengisi kolom kata sandi dengan "123". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400).
	TC-LOGIN-05	Login dengan email terlalu panjang. Kondisi awal: Belum ada sesi.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan teks 269 karakter dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (400).
	TC-LOGIN-06	Login dengan email tidak terdaftar. Kondisi awal: Email tidak ada.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "notfound@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (404).
	TC-LOGIN-07	Login dengan kata sandi salah. Kondisi awal: Akun admin terdaftar.	1. Siapkan data <i>login</i> dengan mengisi kolom email dengan "admin@mail.com" dan mengisi kolom kata sandi dengan "Wrong123!". 2. Kirim data <i>login</i> ke sistem.	Permintaan ditolak (401).

Tabel 4.28 Test Case Login Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-01	TC-LOGIN-08	Login admin berhasil. Kondisi awal: Akun admin dengan email "admin@mail.com", sandi "Valid123!" terdaftar.	1. Siapkan data login dengan mengisi kolom email dengan "admin@mail.com" dan mengisi kolom kata sandi dengan "Valid123!". 2. Kirim data login ke sistem. 3. Periksa data.	Permintaan berhasil (200). Data dengan peran admin.
	TC-LOGIN-09	Login admin - ada kode akses. Kondisi awal: Akun terdaftar.	1. Kirim data login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Periksa respons.	Ada "accessToken" dan "refreshToken".
	TC-LOGIN-10	Login admin - cookie tersimpan. Kondisi awal: Akun terdaftar.	1. Kirim data login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Periksa data pada cookie.	Cookie berisi token.
	TC-INT-LOGIN-001	Login - session tercatat. Kondisi awal: Akun terdaftar, belum ada sesi.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Periksa sesi.	Sesi baru.
	TC-INT-LOGIN-002	Login ganda. Kondisi awal: Akun terdaftar.	1. Login dua kali dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Periksa sesi.	Dua sesi.
	TC-INT-LOGIN-003	Body kosong. Kondisi awal: Tidak ada data.	1. Kosongkan seluruh body (tanpa email dan kata sandi). 2. Kirim data ke sistem.	Permintaan ditolak (400).
	TC-INT-LOGIN-004	Tanpa email. Kondisi awal: Tidak ada data.	1. Kirim hanya kata sandi "Valid123!" tanpa menyertakan email.	Permintaan ditolak (400).
	TC-INT-LOGIN-005	Tanpa kata sandi. Kondisi awal: Tidak ada data.	1. Kirim hanya email "admin@mail.com" tanpa menyertakan kata sandi.	Permintaan ditolak (400).
	TC-INT-LOGIN-006	User belum verifikasi. Kondisi awal: Akun dengan "verified": false.	1. Login dengan email akun admin yang belum diverifikasi dan kata sandi akun tersebut.	Permintaan berhasil (200).

Tabel 4.28 Test Case Login Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-01	TC-INT-LOGIN-007	Akun dinonaktifkan. Kondisi awal: Akun dinonaktifkan.	1. Login dengan email akun admin yang dinonaktifkan dan kata sandi akun tersebut.	Permintaan ditolak (403).
	TC-INT-LOGIN-008	Akun dihapus. Kondisi awal: Akun dihapus.	1. Login dengan email akun admin yang sudah dihapus dan kata sandi akun tersebut.	Permintaan ditolak (404).
	TC-INT-LOGIN-009	Email dengan spasi. Kondisi awal: Akun terdaftar.	1. Kirim email "admin@mail.com " dan kata sandi "Valid123!".	Permintaan berhasil (200).
	TC-INT-LOGIN-010	Kata sandi dengan spasi. Kondisi awal: Akun terdaftar.	1. Kirim email "admin@mail.com" dan kata sandi " Valid123! ".	Permintaan berhasil (200).
	TC-INT-LOGIN-011	Login setelah register. Kondisi awal: Admin baru dibuat.	1. Login dengan email dan kata sandi akun admin yang baru dibuat.	Permintaan berhasil (200).
	TC-INT-LOGIN-012	Login berkali-kali. Kondisi awal: Akun terdaftar.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!" 5 kali.	Semua berhasil (200).
	TC-INT-LOGIN-013	Email huruf besar. Kondisi awal: Akun dengan email campuran.	1. Login dengan email "Admin@Mail.com" dan kata sandi "Valid123!".	Permintaan berhasil (200).
	TC-INT-LOGIN-014	Login setelah logout. Kondisi awal: Sesi sebelumnya dihapus.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!" setelah logout.	Permintaan berhasil (200).
	TC-INT-LOGIN-015	Data palsu. Kondisi awal: Tidak ada akun.	1. Kirim email "fake@mail.com" dan kata sandi "Fake123!".	Permintaan ditolak (401).
	TC-INT-LOGIN-016	Status sukses. Kondisi awal: Akun terdaftar.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Periksa status.	"success".

Tabel 4.28 Test Case Login Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-01	TC-INT-LOGIN-017	Data user sesuai. Kondisi awal: Akun terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Periksa data.	Sesuai basis data.
	TC-INT-LOGIN-018	Tidak mengubah data. Kondisi awal: Akun terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Bandingkan data.	Tidak berubah.
	TC-INT-LOGIN-019	Token akses valid. Kondisi awal: Akun terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Gunakan token.	Token valid.
	TC-INT-LOGIN-020	Cookie dibersihkan saat <i>logout</i> . Kondisi awal: Akun terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. <i>Logout</i> . 3. Periksa data pada cookie.	Cookie dibersihkan.
	TC-INT-LOGIN-021	Header JSON. Kondisi awal: Akun terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Periksa header.	Content-Type: "application/json".
	TC-INT-LOGIN-022	Kata sandi minimal. Kondisi awal: Akun dengan sandi 8 karakter.	1. <i>Login</i> dengan email akun admin yang memiliki kata sandi 8 karakter dan kata sandi tersebut.	Permintaan berhasil (200).
	TC-INT-LOGIN-023	Data palsu - tidak buat sesi. Kondisi awal: Akun palsu.	1. Kirim email "fake@mail.com" dan kata sandi "Fake123!". 2. Periksa sesi.	Tidak ada sesi baru.
	TC-INT-LOGIN-024	<i>Login</i> - cookie baru setelah <i>logout</i> lalu <i>login</i> . Kondisi awal: Akun terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. <i>Logout</i> . 3. <i>Login</i> lagi dengan email "admin@mail.com" dan kata sandi "Valid123!".	Cookie baru.

Tabel 4.28 memaparkan rincian *test case* untuk modul *login* pada aktor admin, pengujian ini berfokus pada validasi proses autentikasi. Skenario yang diuji

difokuskan untuk memastikan bahwa sistem tangguh dalam mencegah akses yang tidak valid.

Tabel 4. 29 Test Case Logout Admin

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-01	TC-LOGOUT-01	Keluar tanpa kode akses. Kondisi awal: Admin sudah masuk.	1. Kirim permintaan <i>logout</i> tanpa token akses (Authorization header) dan tanpa cookie.	Permintaan ditolak (401).
	TC-LOGOUT-02	Keluar dengan kode tidak valid. Kondisi awal: Admin sudah masuk.	1. Kirim permintaan <i>logout</i> dengan token akses "invalid" pada Authorization header.	Permintaan ditolak (401).
	TC-LOGOUT-03	Keluar dengan kode kadaluwarsa. Kondisi awal: Token kadaluwarsa.	1. Buat token akses yang sudah kadaluwarsa. 2. Kirim permintaan <i>logout</i> dengan token kadaluwarsa pada Authorization header.	Permintaan ditolak (401).
	TC-LOGOUT-04	Keluar melalui kode akses berhasil. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!" untuk mendapat token akses. 2. Kirim permintaan <i>logout</i> dengan token akses pada Authorization header. 3. Periksa data di basis data.	Permintaan berhasil (200). Sesi terhapus.
	TC-LOGOUT-05	Keluar melalui cookie berhasil. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!" untuk mendapat cookie. 2. Kirim permintaan <i>logout</i> dengan cookie (tanpa Authorization header).	Permintaan berhasil (200). Sesi terhapus.
	TC-INT-LOGOUT-001	Keluar ketika sesi sudah tidak ada. Kondisi awal: Sesi dihapus manual dari basis data.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus sesi manual dari basis data. 3. Kirim permintaan <i>logout</i> dengan token akses.	Permintaan ditolak (401). Sesi tidak ditemukan.
	TC-INT-LOGOUT-002	Keluar menghapus sesi dari basis data.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Catat ID sesi.	Sesi tidak ditemukan.

Tabel 4.29 Test Case Logout Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-01		Kondisi awal: Sesi aktif.	3. Logout dengan token akses. 4. Cari sesi di basis data.	
	TC-INT-LOGOUT-003	Cookie dibersihkan setelah keluar. Kondisi awal: Sesi aktif.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Logout melalui cookie. 3. Periksa header respons Set-Cookie.	Cookie kode akses dan penyegar kosong.
	TC-INT-LOGOUT-004	Keluar dua kali dengan kode sama. Kondisi awal: Sesi aktif.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Logout dengan token akses (pertama). 3. Logout lagi dengan token akses yang sama (kedua).	Pertama berhasil (200). Kedua ditolak (401).
	TC-INT-LOGOUT-005	Kode akses tanpa data valid. Kondisi awal: Tidak ada sesi.	1. Buat token akses tanpa data payload (sid, uid). 2. Kirim permintaan logout dengan token tersebut.	Permintaan ditolak (401).
	TC-INT-LOGOUT-006	Kode akses tanpa ID sesi. Kondisi awal: Tidak ada sesi.	1. Buat token akses hanya dengan ID user (tanpa sid). 2. Kirim permintaan logout.	Permintaan ditolak (401).
	TC-INT-LOGOUT-007	Header kode akses tanpa awalan Bearer. Kondisi awal: Sesi aktif.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim header Authorization dengan nilai token akses langsung (tanpa awalan "Bearer ").	Permintaan ditolak (401). Format salah.
	TC-INT-LOGOUT-008	User dihapus tetapi keluar tetap berhasil. Kondisi awal: Sesi aktif, user dihapus.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus user dari basis data. 3. Kirim permintaan logout dengan token akses.	Permintaan berhasil (200).
	TC-INT-LOGOUT-009	Multiple sesi - hanya satu terhapus saat keluar. Kondisi awal: Dua sesi aktif untuk user sama.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!" dua kali untuk mendapat dua sesi. 2. Logout dengan token akses sesi pertama. 3. Periksa data di basis data.	Sesi pertama hilang, sesi kedua masih ada.

Tabel 4.29 Test Case Logout Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-01	TC-INT-LOGOUT-010	Cookie dan header kode akses dikirim bersamaan. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim <i>logout</i> dengan cookie dan Authorization header bersamaan.	Permintaan berhasil (200).
	TC-INT-LOGOUT-011	Keluar - status sukses. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. <i>Logout</i> dengan token akses. 3. Periksa status.	Status "success".
	TC-INT-LOGOUT-012	Keluar - pesan sesuai. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. <i>Logout</i> dengan token akses. 3. Periksa pesan.	Pesan "Logout successful".
	TC-INT-LOGOUT-013	Keluar tanpa masuk sebelumnya. Kondisi awal: Tidak ada sesi.	1. Kirim permintaan <i>logout</i> tanpa token akses.	Permintaan ditolak (401).
	TC-INT-LOGOUT-014	Keluar - kode akses tidak bisa dipakai lagi. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. <i>Logout</i> dengan token akses. 3. Coba akses fitur dengan token akses lama.	Kode akses tidak valid.
	TC-INT-LOGOUT-015	Keluar - data user tidak berubah. Kondisi awal: Sesi aktif.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Catat data user. 3. <i>Logout</i> dengan token akses. 4. Bandingkan data.	Data user tidak berubah.

Tabel 4.29 menyajikan rincian *test case* untuk fitur *logout* pada pengguna dengan hak akses admin. Pengujian ini difokuskan pada validasi mekanisme manajemen sesi berbasis token, dengan memastikan bahwa sistem mampu mengakhiri sesi pengguna secara menyeluruh ketika proses *logout* dilakukan. Selain itu, pengujian juga bertujuan untuk memverifikasi bahwa token autentikasi yang sebelumnya digunakan menjadi tidak valid sehingga tidak dapat dimanfaatkan kembali untuk mengakses fitur yang memerlukan autentikasi.

Tabel 4. 30 Test Case Kelola User Admin

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-02	TC-UM-001	Lihat semua user tanpa kode akses. Kondisi awal: Admin di basis data.	1. Kirim permintaan daftar user tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-UM-002	Lihat semua user dengan kode tidak valid. Kondisi awal: Admin di basis data.	1. Kirim permintaan daftar user dengan token akses "invalid" pada Authorization header.	Permintaan ditolak (401).
	TC-UM-003	Lihat semua user sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan daftar seluruh user.	Permintaan ditolak (403).
	TC-UM-004	Lihat semua user berhasil (admin). Kondisi awal: Admin login.	1. Login menggunakan akun admin (email "admin@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan daftar seluruh user.	Permintaan berhasil (200). Daftar user.
	TC-UM-005	Lihat user - hanya admin yang ada. Kondisi awal: Hanya admin di basis data.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar user.	1 user (admin).
	TC-UM-006	Lihat user by ID valid. Kondisi awal: User terdaftar.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan dengan userId valid.	Permintaan berhasil (200). User ditemukan.
	TC-UM-007	Lihat user by ID tidak ditemukan. Kondisi awal: User tidak ada.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan dengan userId palsu (ObjectId valid tapi tidak ada).	Permintaan ditolak (404).
	TC-UM-008	Lihat user dengan format ID invalid. Kondisi awal: -	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim userId "invalid-id" (bukan format ObjectId).	Permintaan gagal (400).

Tabel 4.30 Test Case Kelola User Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-02	TC-UM-009	Buat user baru tanpa foto. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Siapkan data user baru dengan email "newuser@mail.com", password "Password1!", role "mahasiswa", name "New User".	Permintaan berhasil (201).
	TC-UM-010	Buat user dengan email duplikat. Kondisi awal: Email sudah ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Siapkan data user baru dengan email "mahasiswa@mail.com" (sudah terdaftar), password "Password1!", role "mahasiswa".	Permintaan gagal (409).
	TC-UM-011	Buat user tanpa field wajib. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kosongkan seluruh data body (tanpa email, password, role). 3. Kirim data ke sistem.	Permintaan gagal (400).
	TC-UM-012	Update user berhasil. Kondisi awal: User terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim pembaruan email user menjadi "updated@mail.com".	Permintaan berhasil (200).
	TC-UM-013	Update password - terenkripsi. Kondisi awal: User terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim pembaruan password "NewPass123!". 3. Periksa data di basis data.	Password tersimpan tidak sama dengan aslinya.
	TC-UM-014	Update user tidak ditemukan. Kondisi awal: User tidak ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim pembaruan dengan userId palsu (ObjectId valid tapi tidak ada).	Permintaan ditolak (404).
	TC-UM-015	Hapus user berhasil. Kondisi awal: User terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus user dengan userId valid (bukan admin sendiri).	Permintaan berhasil (200). User terhapus.

Tabel 4.30 Test Case Kelola User Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-02	TC-UM-016	Hapus user tidak ditemukan. Kondisi awal: User tidak ada.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus dengan userId palsu (ObjectId valid tapi tidak ada).	Permintaan ditolak (404).
	TC-UM-017	Hapus user dengan format ID invalid. Kondisi awal: -	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus dengan userId "invalid-id" (bukan format ObjectId).	Permintaan gagal (400).
	TC-UM-018	Buat user dengan peran mahasiswa. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Siapkan data user baru dengan email "mhsbaru@mail.com", password "Password1!", role "mahasiswa", name "Mahasiswa Baru".	Permintaan berhasil (201).
	TC-UM-019	Buat user dengan peran psikolog. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Siapkan data user baru dengan email "psikologbaru@mail.com", password "Password1!", role "psikolog", name "Psikolog Baru".	Permintaan berhasil (201).
	TC-UM-020	Buat user dengan peran admin. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Siapkan data user baru dengan email "adminbaru@mail.com", password "Password1!", role "admin", name "Admin Baru".	Permintaan berhasil (201).
	TC-UM-021	Update user - email tidak boleh duplikat. Kondisi awal: Email sudah dipakai user lain.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim pembaruan user A (bukan admin) dengan email "mahasiswa@mail.com" (sudah dipakai user lain).	Permintaan gagal (409).

Tabel 4.30 Test Case Kelola User Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-02	TC-UM-022	Lihat semua user - data tidak sensitif. Kondisi awal: Ada beberapa user.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar seluruh user. 3. Periksa respons.	Respons tidak mengandung kata sandi.
	TC-UM-023	Hapus user - tidak bisa hapus diri sendiri. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus akun admin sendiri (userId admin yang <i>login</i>).	Permintaan ditolak (400).
	TC-UM-024	Buat user - email format valid. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Siapkan data user baru dengan email "valid@mail.com", password "Password1!", role "mahasiswa", name "Valid User".	Permintaan berhasil (201).
	TC-UM-025	Update user - field tidak dikenal diabaikan. Kondisi awal: User terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim pembaruan dengan email "updated@mail.com" (valid) dan field asing "unknown": "field".	Field valid (email) berubah.
	TC-INT-001	Lihat semua user tanpa kode akses. Kondisi awal: Admin di DB.	1. Kirim permintaan daftar seluruh user tanpa header Authorization.	Permintaan ditolak (401).
	TC-INT-002	Lihat semua user dengan token tidak valid. Kondisi awal: Admin di DB.	1. Kirim permintaan daftar seluruh user dengan token invalid.	Permintaan ditolak (401).
	TC-INT-003	Lihat semua user sebagai non-admin ditolak. Kondisi awal: Mahasiswa <i>login</i> .	1. <i>Login</i> menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan daftar seluruh user.	Permintaan ditolak (403).

Tabel 4.30 Test Case Kelola User Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-02	TC-INT-004	Lihat semua user berhasil (admin). Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar seluruh user.	Permintaan berhasil (200). data adalah array.
	TC-INT-005	Tidak ada user di DB (hanya admin). Kondisi awal: Hanya admin di DB.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar seluruh user.	Permintaan berhasil (200). Hanya satu user dengan peran admin yang ditampilkan.
	TC-INT-006	Lihat user by ID valid. Kondisi awal: User terdaftar.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat user dengan userId valid.	Permintaan berhasil (200). data defined.
	TC-INT-007	Lihat user by ID tidak ditemukan. Kondisi awal: User tidak ada.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat user.	Permintaan ditolak (404).
	TC-INT-008	Lihat user dengan format ID invalid. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat user.	Permintaan gagal (500).
	TC-INT-009	Buat user baru tanpa foto. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat user dengan email, password, role.	Permintaan berhasil (201). email user baru sesuai dengan yang dimasukkan.
	TC-INT-010	Buat user dengan email duplikat. Kondisi awal: Email sudah dipakai.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan user email yang sudah terdaftar.	Permintaan gagal (500).
	TC-INT-011	Buat user tanpa field wajib. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat user dengan body {}.	Permintaan gagal (500).
	TC-INT-012	Update user berhasil. Kondisi awal: User terdaftar.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update user dengan email baru.	Permintaan berhasil (200). data defined.

Tabel 4.30 Test Case Kelola User Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-02	TC-INT-013	Update <i>password</i> - terenkripsi. Kondisi awal: User terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update user dengan <i>password</i> baru. 3. Periksa basis data.	<i>Password</i> tersimpan tidak sama dengan aslinya.
	TC-INT-014	Update user tidak ditemukan. Kondisi awal: User tidak ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update user.	Permintaan ditolak (404).
	TC-INT-015	Hapus user berhasil. Kondisi awal: User terdaftar.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus user dengan <i>userId</i> valid (bukan admin sendiri).	Permintaan berhasil (200).
	TC-INT-016	Hapus user tidak ditemukan. Kondisi awal: User tidak ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus user.	Permintaan ditolak (404).
	TC-INT-017	Hapus user dengan format ID invalid. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus user.	Permintaan gagal (500).
	TC-INT-018	Buat user dengan peran mahasiswa. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat user dengan role "mahasiswa".	Permintaan berhasil (201).
	TC-INT-019	Buat user dengan peran psikolog. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat user dengan role "psikolog".	Permintaan berhasil (201).
	TC-INT-020	Buat user dengan peran admin. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat user dengan role "admin".	Permintaan berhasil (201).
	TC-INT-021	Update user - email tidak boleh duplikat. Kondisi awal: Email sudah dipakai user lain.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update user dengan email yang sudah dipakai.	Permintaan gagal (409).

Tabel 4.30 Test Case Kelola User Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-02	TC-INT-022	Lihat semua user - data tidak sensitif. Kondisi awal: Ada beberapa user.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar seluruh user. 3. Periksa respons.	Respons tidak mengandung kata sandi.
	TC-INT-023	Hapus user - tidak bisa hapus diri sendiri. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus user dengan userId admin sendiri.	Permintaan ditolak (400).

Tabel 4.30 menyajikan rincian *test case* untuk fitur kelola user pada pengguna dengan hak akses admin. Pengujian ini difokuskan pada validasi kemampuan admin dalam mengelola data pengguna yang terdaftar di dalam sistem. Skenario yang diuji mencakup berbagai proses pengelolaan, seperti melihat, menambahkan, mengubah, maupun menghapus data pengguna sesuai dengan hak akses yang dimiliki admin.

Tabel 4. 31 Test Case Kelola Artikel (Admin)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-03	TC-ADM-ART-01	Buat artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim data artikel (judul "Test", konten "Test content") tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-ADM-ART-02	Buat artikel sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim data artikel (judul "Test", konten "Test content").	Permintaan ditolak (403).
	TC-ADM-ART-03	Buat artikel tanpa judul. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim data artikel dengan konten "Test content" tanpa menyertakan judul.	Permintaan ditolak (400).

Tabel 4.31 Test Case Kelola Artikel Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-03	TC-ADM-ART-04	Buat artikel tanpa konten. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim data artikel dengan judul "Test" tanpa menyertakan konten.	Permintaan ditolak (400).
	TC-ADM-ART-05	Buat artikel dengan data valid. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Judul Artikel", konten "Konten artikel". 3. Periksa respons.	Permintaan berhasil (201). Artikel tersimpan.
	TC-ADM-ART-06	Lihat semua artikel. Kondisi awal: Admin <i>login</i> , ada artikel.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar artikel.	Permintaan berhasil (200). Daftar artikel.
	TC-ADM-ART-07	Lihat artikel by ID valid. Kondisi awal: Artikel ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan dengan articleId valid.	Permintaan berhasil (200).
	TC-ADM-ART-08	Lihat artikel by ID tidak ditemukan. Kondisi awal: ID palsu.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan dengan articleId palsu (ObjectId valid tapi tidak ada).	Permintaan ditolak (404).
	TC-ADM-ART-09	Lihat artikel dengan ID format invalid. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim articleId "invalid-id" (bukan format ObjectId).	Permintaan ditolak (400).
	TC-ADM-ART-10	Update artikel valid. Kondisi awal: Artikel ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Updated Title", konten "Updated content".	Permintaan berhasil (200).
	TC-ADM-ART-11	Update hanya judul - slug berubah. Kondisi awal: Artikel dengan slug "old-title".	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "New Title".	Slug jadi "new-title".

Tabel 4.31 Test Case Kelola Artikel Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-03	TC-ADM-ART-12	Update hanya konten. Kondisi awal: Artikel ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim konten "Updated content".	Permintaan berhasil (200).
	TC-ADM-ART-13	Update tanpa field - data tetap. Kondisi awal: Artikel ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kosongkan seluruh data body (tanpa judul dan konten). 3. Kirim data ke sistem.	Permintaan berhasil (200). Data tidak berubah.
	TC-ADM-ART-14	Hapus artikel valid. Kondisi awal: Artikel ada.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus artikel dengan articleId valid.	Permintaan berhasil (200).
	TC-ADM-ART-15	Hapus artikel tidak ditemukan. Kondisi awal: ID palsu.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus dengan articleId palsu (ObjectId valid tapi tidak ada).	Permintaan ditolak (404).
	TC-ADM-ART-16	Hapus artikel tanpa kode akses. Kondisi awal: Artikel ada.	1. Kirim permintaan hapus (articleId valid) tanpa token akses (Authorization header).	Permintaan ditolak (401).
	TC-ADM-ART-17	Hapus artikel dua kali. Kondisi awal: Artikel ada lalu dihapus.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Hapus artikel dengan articleId valid (pertama). 3. Hapus artikel yang sama lagi (kedua).	Pertama berhasil (200). Kedua ditolak (404).
	TC-ADM-ART-18	Slug otomatis saat buat. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Hello World", konten "Content". 3. Periksa slug.	Slug jadi "hello-world".
	TC-ADM-ART-19	Buat artikel - judul dan konten minimal. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "A", konten "B".	Permintaan berhasil (201).

Tabel 4.31 Test Case Kelola Artikel Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-03	TC-ADM-ART-20	Update artikel - slug berubah saat judul diupdate. Kondisi awal: Artikel ada.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Updated Title". 3. Periksa slug.	Slug jadi "updated-title".
	TC-ADM-ART-21	Format respons artikel. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim judul "Hello", konten "World". 3. Periksa format.	Ada field: id, title, slug, content, writer.
	TC-INT-ADM-01	Update artikel ID tidak ditemukan. Kondisi awal: Token admin valid.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel.	Permintaan ditolak (404).
	TC-INT-ADM-02	Hapus artikel ID tidak ditemukan. Kondisi awal: Token admin valid.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus artikel.	Permintaan ditolak (404).
	TC-INT-ADM-03	Lihat artikel by ID tidak ditemukan. Kondisi awal: Token admin valid.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat artikel.	Permintaan ditolak (404).
	TC-INT-ADM-04	Lihat artikel dengan ID format invalid. Kondisi awal: Token admin valid.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat artikel.	Permintaan ditolak (400).
	TC-INT-ADM-05	Buat artikel tanpa kode akses. Kondisi awal: Belum ada sesi.	1. Kirim permintaan buat artikel tanpa token akses.	Permintaan ditolak (401).
	TC-INT-ADM-06	Buat artikel sebagai mahasiswa ditolak. Kondisi awal: Mahasiswa login.	1. Login menggunakan akun mahasiswa (email "mahasiswa@mail.com", kata sandi "Valid123!"). 2. Kirim permintaan buat artikel.	Permintaan ditolak (403).

Tabel 4.31 Test Case Kelola Artikel Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-03	TC-INT-ADM-07	Buat artikel tanpa judul. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel tanpa field title.	Permintaan ditolak (400).
	TC-INT-ADM-08	Buat artikel tanpa konten. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel tanpa field content.	Permintaan ditolak (400).
	TC-INT-ADM-09	Buat artikel dengan data valid. Kondisi awal: Admin login.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel dengan title dan content.	Permintaan berhasil (201). data defined.
	TC-INT-ADM-10	Lihat semua artikel. Kondisi awal: Admin login, ada artikel.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar artikel.	Permintaan berhasil (200). data adalah array.
	TC-INT-ADM-11	Lihat artikel by ID valid. Kondisi awal: Artikel ada di DB.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan lihat artikel dengan articleId valid.	Permintaan berhasil (200). data defined.
	TC-INT-ADM-12	Update artikel tanpa field - data tetap. Kondisi awal: Artikel ada di DB.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan body {}.	Permintaan berhasil (200). Data tidak berubah.
	TC-INT-ADM-13	Hapus artikel tanpa kode akses. Kondisi awal: Artikel ada.	1. Kirim permintaan hapus artikel tanpa token akses.	Permintaan ditolak (401).
	TC-INT-ADM-14	Hapus artikel dua kali. Kondisi awal: Artikel ada lalu dihapus.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus artikel (pertama). 3. Kirim permintaan hapus artikel lagi (kedua).	Pertama berhasil (200). Kedua ditolak (404).
	TC-INT-ADM-15	Update artikel valid.	1. Login dengan email "admin@mail.com" dan kata sandi "Valid123!".	Permintaan berhasil (200).

Tabel 4.31 Test Case Kelola Artikel Admin (Lanjutan)

Scenario ID	Test Case ID	Scenario Description	Test Steps	Expected Result
TS-ADM-03		Kondisi awal: Artikel ada di DB.	2. Kirim permintaan update artikel dengan title dan content baru.	
	TC-INT-ADM-16	Update hanya judul - slug ikut berubah. Kondisi awal: Artikel ada di DB.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan title baru.	Permintaan berhasil (200).
	TC-INT-ADM-17	Update hanya konten. Kondisi awal: Artikel ada di DB.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan content baru.	Permintaan berhasil (200).
	TC-INT-ADM-18	Hapus artikel valid. Kondisi awal: Artikel ada di DB.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan hapus artikel.	Permintaan berhasil (200).
	TC-INT-ADM-19	Slug otomatis terbentuk saat buat. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel dengan title valid.	slug artikel terbentuk secara otomatis.
	TC-INT-ADM-20	Slug berubah saat judul diupdate. Kondisi awal: Artikel ada di DB.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan update artikel dengan title baru.	Permintaan berhasil (200).
	TC-INT-ADM-21	Format respons sesuai struktur. Kondisi awal: Artikel ada di DB.	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan daftar artikel.	setiap artikel dalam daftar memiliki judul dan slug.
	TC-INT-ADM-22	Buat artikel - judul dan konten minimal. Kondisi awal: Admin <i>login</i> .	1. <i>Login</i> dengan email "admin@mail.com" dan kata sandi "Valid123!". 2. Kirim permintaan buat artikel dengan judul "A" dan konten "B".	Permintaan berhasil (201).

Tabel 4.31 menguraikan *test case* untuk modul kelola artikel pada hak akses admin. Pengujian ini berfokus pada validasi keberhasilan pembuatan konten baru,

serta memverifikasi ketahanan sistem dalam menangani *anomali input* maupun upaya akses ilegal dari pengguna tanpa autentikasi.

4. Requirement Traceability Matrix (RTM) Admin

Requirement Traceability Matrix (RTM) Admin digunakan untuk menunjukkan keterlacakan antara setiap kebutuhan fungsional yang telah didefinisikan dengan skenario pengujian yang dilakukan.

Tabel 4. 32 RTM Admin

Scenario ID	Requirements Description	Test Case ID
TS-ADM-01 (Autentikasi)	Admin dapat <i>login</i> ke sistem	TC-LOGIN-01 s.d. TC-LOGIN-10, TC-INT-LOGIN-001 s.d. TC-INT-LOGIN-024
TS-ADM-01 (Autentikasi)	Admin dapat <i>logout</i> dari sistem	TC-LOGOUT-01 s.d. TC-LOGOUT-05, TC-INT-LOGOUT-001 s.d. TC-INT-LOGOUT-015
TS-ADM-02 (Kelola User)	Admin dapat melihat daftar semua user	TC-UM-001 s.d. TC-UM-005, TC-UM-022
TS-ADM-02 (Kelola User)	Admin dapat membuat user baru (semua role)	TC-UM-009 s.d. TC-UM-011, TC-UM-018 s.d. TC-UM-020, TC-UM-024
TS-ADM-02 (Kelola User)	Admin dapat memperbarui data user	TC-UM-012 s.d. TC-UM-014, TC-UM-021, TC-UM-025
TS-ADM-02 (Kelola User)	Admin dapat menghapus user	TC-UM-015 s.d. TC-UM-017, TC-UM-023
TS-ADM-03 (Kelola Artikel)	Admin dapat membuat artikel	TC-ADM-ART-01 s.d. TC-ADM-ART-05, TC-ADM-ART-18, TC-ADM-ART-19, TC-ADM-ART-21
TS-ADM-03 (Kelola Artikel)	Admin dapat melihat daftar dan detail artikel	TC-ADM-ART-06 s.d. TC-ADM-ART-09
TS-ADM-03 (Kelola Artikel)	Admin dapat memperbarui artikel	TC-ADM-ART-10 s.d. TC-ADM-ART-13, TC-ADM-ART-20
TS-ADM-03 (Kelola Artikel)	Admin dapat menghapus artikel	TC-ADM-ART-14 s.d. TC-ADM-ART-17

Tabel 4.32 menyajikan pemetaan antara setiap kebutuhan fungsional pada modul admin dengan *test case* yang relevan. Pemetaan tersebut mencakup fitur autentikasi, pengelolaan pengguna, serta pengelolaan artikel. Melalui RTM ini,

dapat dipastikan bahwa setiap *requirement* telah memiliki cakupan pengujian yang memadai, baik pada pengujian *unit* maupun integrasi, sehingga keterlacakan

D. Rekapitulasi *Test Case* yang Dikembangkan

Pemaparan berikut menyajikan rekapitulasi komprehensif seluruh *test case* yang telah dirancang dan diimplementasikan sebagai instrumen pengujian pada sistem SiKosa. Rekapitulasi ini memberikan gambaran mengenai cakupan pengujian yang dilakukan terhadap setiap modul dan kebutuhan fungsional sistem, sehingga dapat menunjukkan bahwa seluruh fitur telah melalui proses verifikasi sesuai dengan skenario pengujian yang telah ditetapkan.

Tabel 4. 33 Rekapitulasi Total *Test Case* per Modul

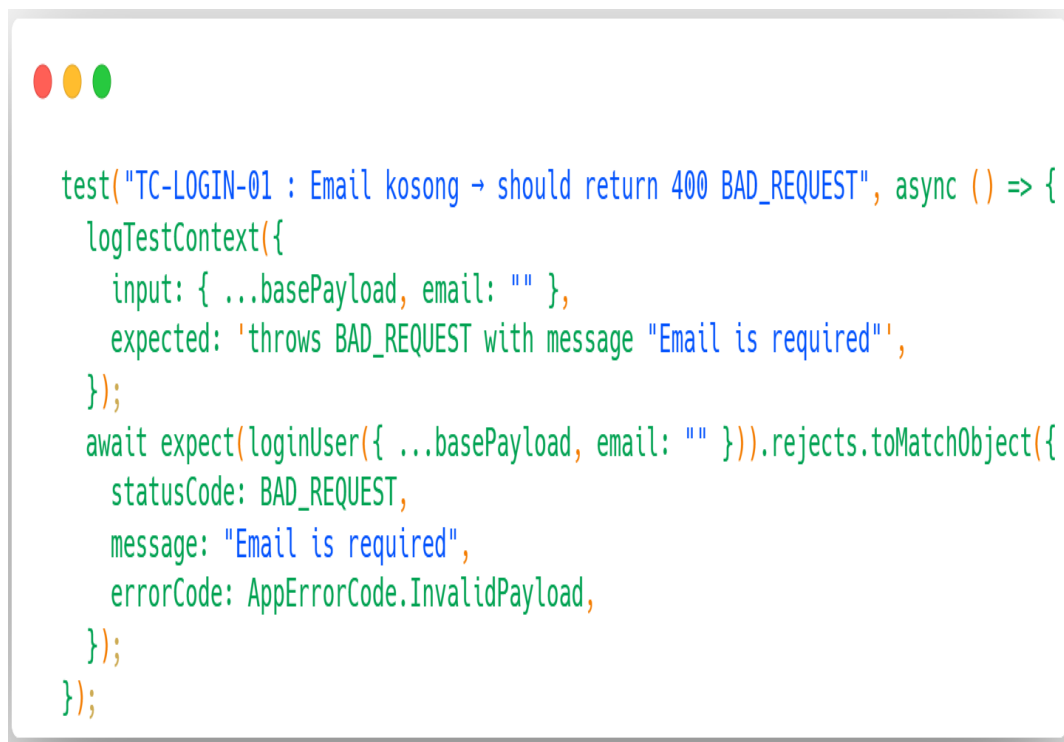
Modul	Unit Testing	Integration Testing	Total
Autentikasi	45	82	127
Manajemen Profil & User	55	55	110
Konsultasi	36	35	71
Chat	55	37	92
Chatbot	14	6	20
Manajemen Artikel	43	43	86
TOTAL	248	258	506

Tabel 4.33 menyajikan rekapitulasi kuantitatif dari seluruh pengujian yang telah dieksekusi pada sistem SiKosa. Secara keseluruhan, terdapat 506 *test case* yang digunakan untuk memverifikasi berbagai fungsionalitas sistem, yang terdiri atas 248 pengujian unit (*unit testing*) dan 258 pengujian integrasi (*integration testing*). Rekapitulasi ini memberikan gambaran mengenai cakupan pengujian yang telah dilakukan, sekaligus menunjukkan bahwa proses pengujian tidak hanya berfokus pada validasi setiap komponen secara individual, tetapi juga pada interaksi

101 antar komponen untuk memastikan sistem dapat berfungsi secara menyeluruh sesuai dengan kebutuhan dan spesifikasi.

E. Contoh Implementasi Kode *Test Case*

Untuk memberikan gambaran konkret bagaimana rancangan *test case* di atas diterjemahkan menjadi kode pengujian, berikut disajikan dua contoh implementasi nyata, satu pada level *unit testing* (modul *Login*) dan satu pada level *integration testing* (modul konsultasi).



```
test("TC-LOGIN-01 : Email kosong -> should return 400 BAD_REQUEST", async () => {
  logTestContext({
    input: { ...basePayload, email: "" },
    expected: 'throws BAD_REQUEST with message "Email is required"',
  });
  await expect(loginUser({ ...basePayload, email: "" })).rejects.toMatchObject({
    statusCode: BAD_REQUEST,
    message: "Email is required",
    errorCode: AppErrorCode.InvalidPayload,
  });
});
```

Gambar 4. 21 Cuplikan kode *unit test service login*

7 Gambar 4.21 menampilkan cuplikan kode *unit test* pada *service login* yang digunakan untuk memverifikasi autentikasi pengguna. Pengujian ini mencakup validasi terhadap kredensial yang diberikan, serta memastikan bahwa proses autentikasi menghasilkan keluaran yang sesuai dengan skenario pengujian ditetapkan.

```
test("TC-INT-CONS-MHS-009 : apply success - should 201 & create DB", async () => {
  const res = await apiTest({
    id: "TC-INT-CONS-MHS-009",
    method: "POST",
    url: "/api/consultation/apply",
    headers: { Authorization: `Bearer ${mahasiswaToken}` },
    payload: { psychologistId: psikologId, message: "Halo" },
    expectedStatus: CREATED,
  });

  expect(res.status).toBe(CREATED);

  const cons = await ConsultationModel.find();
  const rooms = await chatRoom.find();

  expect(cons.length).toBe(1);
  expect(cons[0].status).toBe("pending");

  expect(rooms.length).toBe(1);
  expect(rooms[0].participants.length).toBe(2);
  expect(rooms[0].status).toBe("inactive");
});
```

Gambar 4. 22 Cuplikan kode *integration test* konsultasi

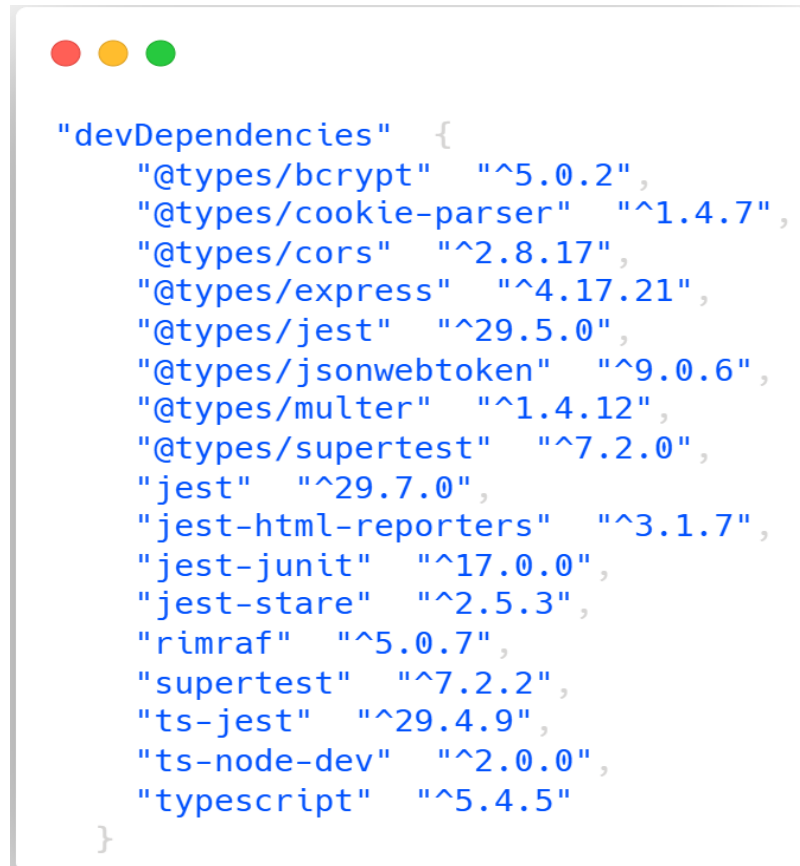
Gambar 4.22 menampilkan cuplikan kode *integration test* pada *chat* yang digunakan untuk memastikan komunikasi berjalan dengan baik. Pengujian ini mencakup validasi *request* dan *response*, serta memastikan alur pengiriman dan penerimaan pesan berfungsi sesuai dengan spesifikasi yang telah ditentukan.

4.2.4 Test Environment Setup

Pada tahap ini dilakukan persiapan lingkungan pengujian agar seluruh skenario yang telah dirancang dapat dijalankan secara optimal dan konsisten. Proses yang dilakukan mencakup empat langkah utama:

a) Instalasi *Tools* Pengujian

Seluruh *tools* pengujian diinstal sebagai *devDependencies* menggunakan *npm*, sehingga tidak mempengaruhi *bundle* produksi. Instalasi yang digunakan:



```
"devDependencies" {
  "@types/bcrypt"    "^5.0.2",
  "@types/cookie-parser"  "^1.4.7",
  "@types/cors"      "^2.8.17",
  "@types/express"    "^4.17.21",
  "@types/jest"       "^29.5.0",
  "@types/jsonwebtoken"  "^9.0.6",
  "@types/multer"     "^1.4.12",
  "@types/supertest"   "^7.2.0",
  "jest"              "^29.7.0",
  "jest-html-reporters"  "^3.1.7",
  "jest-junit"         "^17.0.0",
  "jest-stare"         "^2.5.3",
  "rimraf"             "^5.0.7",
  "supertest"          "^7.2.2",
  "ts-jest"             "^29.4.9",
  "ts-node-dev"        "^2.0.0",
  "typescript"         "^5.4.5"
}
```

Gambar 4. 23 Instalasi *devDependencies* pengujian

Gambar 4.23 menampilkan proses instalasi *devDependencies* yang digunakan untuk keperluan pengujian pada proyek. Dependensi ini diinstal sebagai kebutuhan *development environment*, seperti *unit testing* dan *integration testing*, sehingga mendukung proses validasi kode tanpa memengaruhi dependensi produksi aplikasi.

b) Konfigurasi Pengujian (*Jest.config.js*)

Berikut merupakan potongan isi berkas *Jest.config.js* yang digunakan sebagai konfigurasi *framework* pengujian pada proyek SiKosa. Berkas ini mendefinisikan pengaturan utama yang digunakan selama proses eksekusi pengujian, termasuk lingkungan pengujian, cakupan pengujian, serta konfigurasi lain yang mendukung pelaksanaan *test case* secara konsisten.

```

const { createDefaultPreset } = require("ts-jest");

const tsJestTransformCfg = createDefaultPreset() transform;

@type {import("jest") Config}
module exports {
  verbose true,
  preset "ts-jest",
  testEnvironment "node",
  transform {
    tsJestTransformCfg,
  },
  rootDir ".",
  moduleFileExtensions ["ts", "js", "json"],
  reporters ["default"],
  reporters [
    "<rootDir>/tests/reporters/verbose-reporter.js",

    [
      "jest-html-reporters",
      {
        publicPath "./reports",
        filename "integration-report.html",
        expand true,
      },
    ],
    [
      "jest-stare",
      {
        resultDir "./reports/jest-stare",
        reportTitle "SIKOSA Integration Test Report",
        coverageLink "../coverage/lcov-report/index.html",
      },
    ],
    [
      "jest-junit",
      {
        outputDirectory "./reports",
        outputName "junit.xml",
      },
    ],
  ],
  testMatch ["**/*.spec.ts", "**/*.integration.spec.ts"],
  collectCoverageFrom ["src/**/*.ts", "!src/**/*.spec.ts", "!src/**/*.integration.spec.ts",
"!src/**/*.d.ts"],
  coverageDirectory "coverage",
  detectOpenHandles true,
  forceExit true,
  clearMocks true,
  restoreMocks true,
  moduleNameMapper {
    "@(/(.*)$)" "<rootDir>/src/$1",
  },
  collectCoverage true,
  coverageProvider "v8",
  setupFiles ["<rootDir>/tests/setup/env.ts"],
  setupFilesAfterEnv ["<rootDir>/tests/setup/setupTest.ts"],
};

```

Gambar 4. 24 Konfigurasi Pengujian

Gambar 4.24 menampilkan konfigurasi pengujian yang digunakan untuk mengatur lingkungan *testing* pada proyek. Konfigurasi ini mencakup pengaturan *framework testing*, setup awal sebelum pengujian dijalankan, serta parameter pendukung lain agar proses *unit test* dan *integration test* dapat berjalan secara konsisten dan sesuai dengan struktur aplikasi.

c) Penyusunan Struktur Direktori Pengujian

Struktur folder pengujian disusun secara sistematis untuk memudahkan organisasi *test case* dan pemeliharaan di masa mendatang:



Gambar 4. 25 Struktur folder pengujian

Gambar 4.25 menampilkan struktur folder pengujian pada proyek yang disusun secara modular untuk memisahkan jenis pengujian, yaitu *unit test* dan *integration test*. Struktur ini juga mencakup folder pendukung seperti setup untuk konfigurasi *environment* pengujian, helpers untuk fungsi bantu, serta reporters untuk pengelolaan hasil laporan pengujian.

d) Persiapan Data Uji

Data uji berupa data simulasi (data *dummy*) disiapkan dalam setiap file *unit* dan *integration testing*.

Tabel 4. 34 Data Simulasi yang Digunakan

Tipe Data	Contoh Data	Digunakan di
Akun Mahasiswa	mahasiswa@mail.com	Auth, User, <i>Chat</i>
Akun Psikolog	psikolog@mail.com	Auth, Konsultasi
Akun Admin	admin@mail.com	Auth, Artikel
<i>Password</i>	Valid123! (di-hash)	Semua modul auth
Data Konsultasi	ObjectId generated	Konsultasi, <i>Chat</i>
Data Artikel	Title + Content dummy	Artikel

Tabel 4.34 menyajikan data simulasi yang digunakan sebagai bahan pengujian dalam sistem. Data ini berfungsi untuk merepresentasikan kondisi nyata yang akan diproses oleh aplikasi.

e) Implementasi *Custom Reporter*

Untuk memperjelas keluaran pengujian pada terminal, dibuat sebuah *custom reporter* (*verbose-reporter.js*) yang menampilkan progres eksekusi setiap *suite* dan *test case* secara terstruktur.

```

class VerboseProgressReporter {
  constructor(globalConfig, options) {
    this._globalConfig = globalConfig;
    this._options = options;
  }

  onStart() {
    writeStore({});
    process.stdout.write('\n' + '='.repeat(70) + '\n');
    process.stdout.write(' SIKOSA TEST RUN STARTED\n');
    process.stdout.write('='.repeat(70) + '\n');
  }
}

module.exports = VerboseProgressReporter;

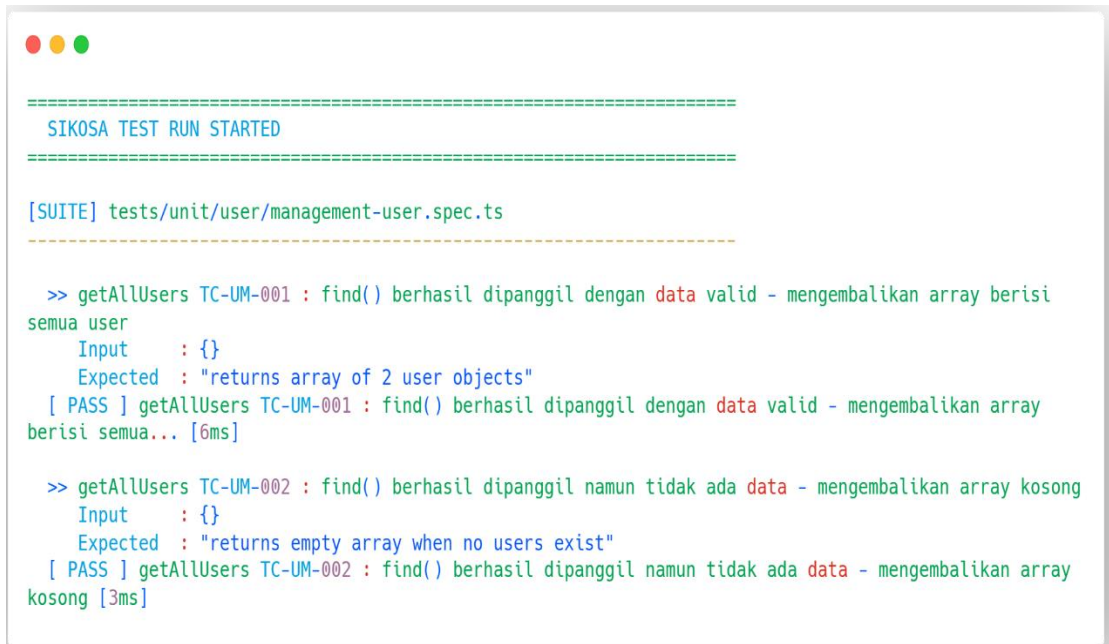
```

Gambar 4. 26 Konfigurasi *Custom Reporter*

Gambar 4.26 menampilkan konfigurasi *custom reporter* yang digunakan untuk mengelola dan menampilkan hasil pengujian secara lebih terstruktur. Konfigurasi ini memungkinkan output *test report* disesuaikan dengan kebutuhan proyek, seperti format laporan, detail hasil eksekusi *test*, serta informasi tambahan yang mendukung proses analisis hasil pengujian.

4.2.5 Test Execution

Pada tahap ini seluruh *test case* yang telah dirancang dieksekusi menggunakan *Jest* melalui *Command Line Interface (CLI)*. Pengujian dijalankan menggunakan perintah: `npm run test`, Setiap hasil eksekusi dicatat secara otomatis oleh *Jest*.



```
=====
SIKOSA TEST RUN STARTED
=====

[SUITE] tests/unit/user/management-user.spec.ts
-----

>> getAllUsers TC-UM-001 : find() berhasil dipanggil dengan data valid - mengembalikan array berisi semua user
  Input      : {}
  Expected   : "returns array of 2 user objects"
[ PASS ] getAllUsers TC-UM-001 : find() berhasil dipanggil dengan data valid - mengembalikan array berisi semua... [6ms]

>> getAllUsers TC-UM-002 : find() berhasil dipanggil namun tidak ada data - mengembalikan array kosong
  Input      : {}
  Expected   : "returns empty array when no users exist"
[ PASS ] getAllUsers TC-UM-002 : find() berhasil dipanggil namun tidak ada data - mengembalikan array kosong [3ms]
```

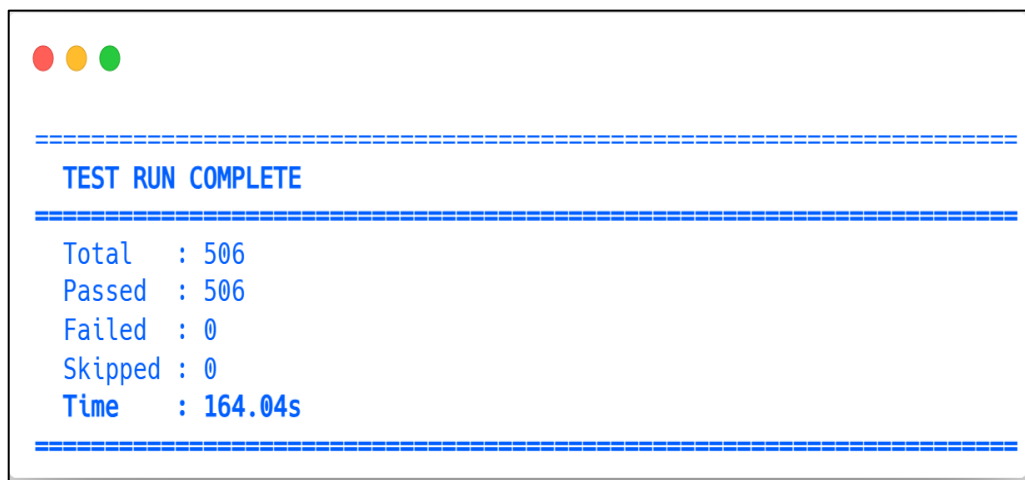
Gambar 4. 27 Proses eksekusi pengujian pada terminal

Gambar 4.27 menampilkan proses eksekusi pengujian pada terminal yang menunjukkan jalannya *unit test* dan *integration test* secara *real-time*. Pada tahap ini, sistem menjalankan seluruh *test case* yang telah didefinisikan, kemudian

menampilkan hasil berupa *status* berhasil atau gagal, termasuk informasi detail terkait *error* apabila ditemukan ketidaksesuaian pada fungsi yang diuji.

a) Hasil Eksekusi

Hasil eksekusi pengujian per modul menunjukkan bahwa seluruh komponen sistem telah diuji menggunakan *unit test* dan *integration test* dengan hasil yang konsisten.



Gambar 4. 28 Hasil eksekusi pengujian

Gambar 4.28 menampilkan hasil eksekusi pengujian yang memperlihatkan rangkuman dari seluruh proses *unit test* dan *integration test* yang telah dijalankan. Output yang ditampilkan menunjukkan status setiap *test case*, termasuk jumlah pengujian yang berhasil maupun gagal, sehingga memberikan gambaran menyeluruh mengenai kualitas dan stabilitas sistem yang diuji.

Tabel 4. 35 Hasil Eksekusi *Test Case* per Modul

Modul	Unit <i>Pass</i> /Total	Integrasi <i>Pass</i> /Total	Total	Status
Autentikasi	45/45	82/82	127	<i>PASS</i>
Manajemen Profil & User	55/55	55/55	110	<i>PASS</i>
Konsultasi	36/36	35/35	71	<i>PASS</i>

Tabel 4.35 Hasil Eksekusi *Test Case* per Modul (Lanjutan)

Modul	Unit <i>Pass</i> /Total	Integrasi <i>Pass</i> /Total	Total	Status
<i>Chat</i>	55/55	37/37	92	<i>PASS</i>
<i>Chatbot</i>	14/14	6/6	20	<i>PASS</i>
Manajemen Artikel	43/43	43/43	86	<i>PASS</i>
TOTAL	248/248	258/258	506	<i>PASS</i>

Tabel 4.35 menyajikan hasil eksekusi *test case* per modul yang mencakup pengujian *unit* dan integrasi pada seluruh komponen sistem. Berdasarkan hasil tersebut, seluruh modul memperoleh *status PASS*, yang menandakan bahwa seluruh fungsionalitas telah berjalan sesuai dengan spesifikasi yang ditetapkan tanpa ditemukan kegagalan pada proses pengujian.

b) Hasil Eksekusi per *Test Suite*

Hasil eksekusi per *test suite* menunjukkan bahwa setiap kelompok pengujian yang dikelompokkan berdasarkan fitur atau modul sistem telah dijalankan secara menyeluruh.

Tabel 4. 36 Rincian Hasil Eksekusi per File *Test Suite*

No	Test Suite	Total	<i>Pass</i>	<i>Fail</i>
1	unit/auth/register.service.spec.ts	30	30	0
2	unit/auth/login.service.spec.ts	10	10	0
3	unit/auth/logout.service.spec.ts	5	5	0
4	unit/artikel/artikel.admin.service.spec	21	21	0
5	unit/artikel/artikel.psikolog.service.sp	22	22	0
6	unit/chat/chat.mahasiswa.service.spec.ts	27	27	0
7	unit/chat/chat.psikolog.service.spec.ts	28	28	0
8	unit/chatbot/chatbot.service.spec.ts	14	14	0
9	unit/consultation/consultation.mhs.spec	18	18	0
10	unit/consultation/consultation.psi.spec	18	18	0
11	unit/user/user.mahasiswa.service.spec.ts	15	15	0
12	unit/user/user.psikolog.service.spec.ts	15	15	0

Tabel 4.36 Rincian Hasil Eksekusi per File *Test Suite* (Lanjutan)

No	Test Suite	Total	Pass	Fail
13	unit/user/management-user.spec.ts	25	25	0
14	integration/auth/register.integration	43	43	0
15	integration/auth/login.integration	24	24	0
16	integration/auth/logout.integration	15	15	0
17	integration/article/article.admin.integ	21	21	0
18	integration/article/article.psikolog.int	22	22	0
19	integration/chat/chat.mahasiswa.integrat	18	18	0
20	integration/chat/chat.psikolog.integrat	19	19	0
21	integration/chatbot/chatbot.integration	6	6	0
22	integration/consultation/cons.mhs.integ	17	17	0
23	integration/consultation/cons.psi.integ	18	18	0
24	integration/user/manajemen-user.integrat	17	17	0
25	integration/user/user.mahasiswa.integrat	18	18	0
26	integration/user/user.psikolog.integrat	20	20	0
	TOTAL	506	506	0

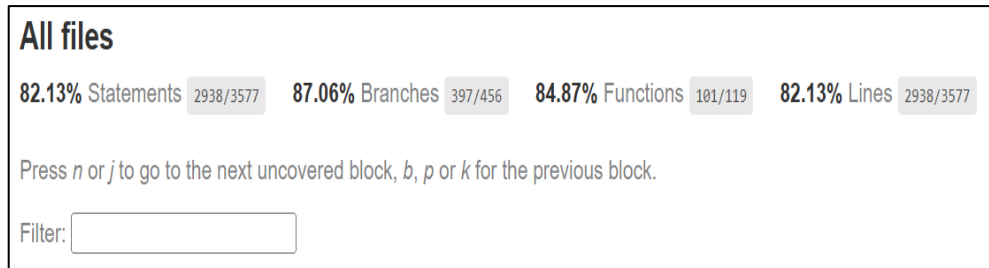
Tabel 4.36 menyajikan rincian hasil eksekusi pengujian berdasarkan file *test suite* yang mencakup *unit test* dan *integration test* pada seluruh modul sistem. Setiap file diuji untuk melihat jumlah *test case* yang dijalankan, jumlah yang berhasil (*pass*), serta jumlah yang gagal (*fail*).

4.2.6 Test Closure

Pada tahap *test closure*, dilakukan evaluasi secara menyeluruh terhadap seluruh hasil pengujian yang telah dilaksanakan untuk menilai tingkat keberhasilan sistem dalam memenuhi kebutuhan fungsional yang telah ditetapkan. Tahap ini bertujuan untuk memastikan bahwa seluruh proses pengujian telah selesai dilaksanakan sesuai dengan rencana serta mengidentifikasi apakah masih terdapat bagian sistem yang memerlukan perbaikan atau pengujian lanjutan.

a) Hasil *Code Coverage*

Hasil *code coverage* menunjukkan tingkat cakupan pengujian terhadap kode sumber aplikasi yang telah dieksekusi selama proses *unit test* dan *integration test*.



Gambar 4. 29 Hasil laporan *test coverage*

Gambar 4.29 menampilkan hasil laporan *test coverage* dari pengujian sistem yang mencakup metrik utama seperti *statement coverage*, *branch coverage*, *function coverage*, dan *line coverage*.

Tabel 4. 37 Hasil *Code Coverage* Pengujian

Jenis Coverage	Persentase	Tercover / Total
<i>Statement Coverage</i>	82.13%	2938 / 3577
<i>Branch Coverage</i>	87.06%	397 / 456
<i>Function Coverage</i>	84.87%	101 / 119
<i>Line Coverage</i>	82.13%	2938 / 32577

Tabel 4.37 menunjukkan hasil pengukuran *code coverage* dari proses pengujian sistem yang mencakup beberapa metrik utama, yaitu *statement coverage*, *branch coverage*, *function coverage*, dan *line coverage*.

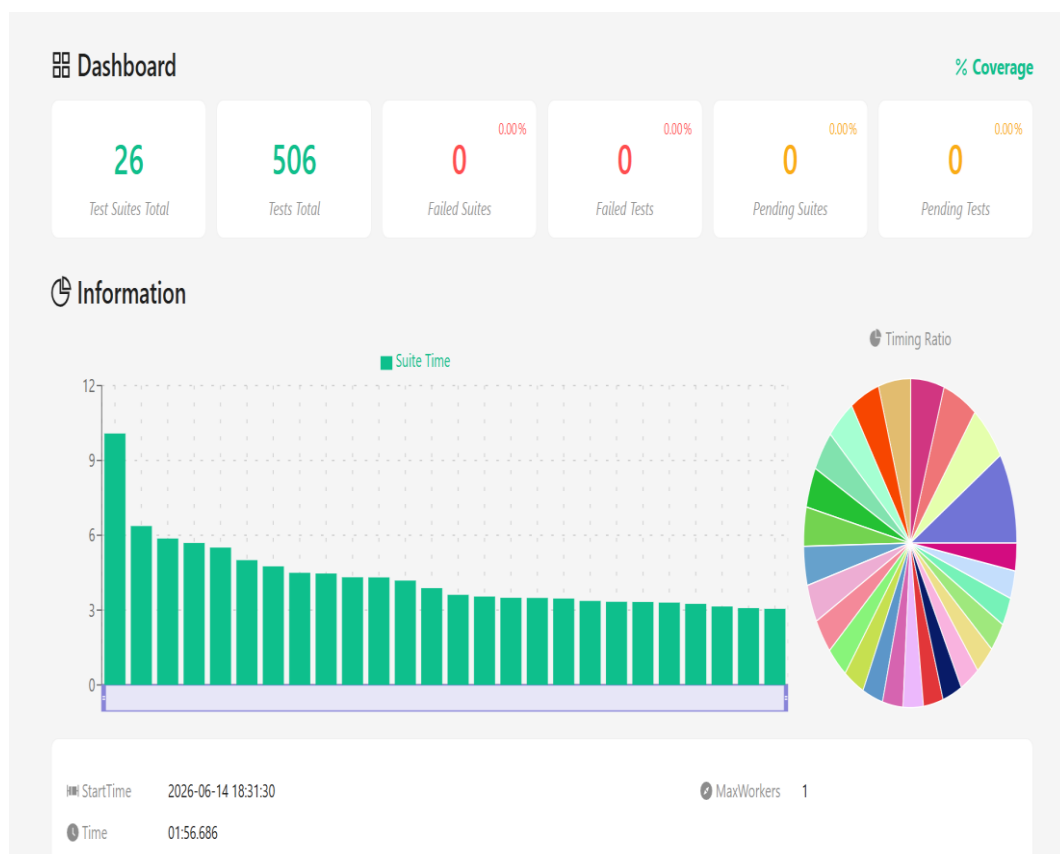
Hasil *code coverage* menunjukkan bahwa *statement coverage* mencapai 82,13% (2938 dari 3577 *statement*), *branch coverage* sebesar 87,06% (397 dari 456 percabangan), *function coverage* sebesar 84,87% (101 dari 119 fungsi), dan *line coverage* sebesar 82,13%. Secara keseluruhan, capaian tersebut menunjukkan

bahwa sebagian besar kode program telah berhasil diuji. Meskipun demikian, masih terdapat beberapa bagian kode yang belum tercakup sehingga diperlukan pengujian tambahan untuk meningkatkan cakupan secara lebih optimal.

b) Laporan Hasil Pengujian

Laporan hasil pengujian pada sistem dihasilkan secara otomatis oleh *Jest* melalui beberapa format laporan yang berbeda, sehingga memudahkan proses analisis hasil pengujian baik untuk kebutuhan debugging maupun integrasi ke pipeline CI/CD.

1. *HTML Report: reports/integration-report.html (Jest-html-reporters).*
2. *Dashboard: reports/Jest-stare/index.html (Jest-stare).*



Gambar 4. 30 Dashboard hasil pengujian (*Jest-stare*)

Gambar 4.30 menampilkan *dashboard* hasil pengujian yang dihasilkan oleh *Jest-stare* sebagai alat visualisasi laporan *testing*. Pada tampilan tersebut, seluruh *test suite* yang dijalankan berjumlah 26 *suite* dan semuanya berhasil (26/26 *pass*) tanpa adanya kegagalan.

Selain itu, pada tingkat *test case*, terlihat bahwa seluruh pengujian juga berjalan sukses dengan *total* 506 *test case* berhasil (506/506 *pass*). Tidak ditemukan adanya *test case* yang gagal (*fail*), yang mengindikasikan bahwa implementasi logika program telah sesuai dengan ekspektasi pada seluruh skenario pengujian yang diuji.

Secara keseluruhan, hasil pada *dashboard* ini menunjukkan bahwa sistem berada dalam kondisi stabil dan konsisten, karena seluruh *unit* pengujian yang dirancang berhasil dilewati tanpa *error*.

c) Ringkasan *Test Closure*

Ringkasan *test closure* ini merangkum hasil akhir dari seluruh aktivitas pengujian yang telah dilakukan pada sistem.

Tabel 4. 38 Ringkasan Hasil *Test Closure*

Metrik	Nilai
Total <i>Test Suite</i>	26
Total <i>Test Case</i>	506
<i>Test Case</i> Lulus (<i>Pass</i>)	506
<i>Test Case</i> Gagal (<i>Fail</i>)	0
Tingkat Keberhasilan	100%
<i>Statement Coverage</i>	82.13%
<i>Branch Coverage</i>	87.06%
<i>Function Coverage</i>	84.87%
<i>Line Coverage</i>	82.13%

Tabel 4.38 menunjukkan ringkasan hasil *test closure* yang merangkum seluruh aktivitas pengujian sistem, baik dari sisi eksekusi *test case* maupun tingkat

code coverage. Secara keseluruhan, pengujian mencakup 26 *test suite* dengan *total* 506 *test case* yang seluruhnya berhasil dijalankan tanpa adanya kegagalan, sehingga menghasilkan tingkat keberhasilan 100%.

4.3 Fase Akhir

4.3.1 Analisis Hasil Pengujian

Berdasarkan hasil eksekusi pengujian yang telah dilakukan pada fase pengujian, diperoleh beberapa temuan dan analisis sebagai berikut:

a) Tingkat Keberhasilan Pengujian (*Test Success Rate*)

Seluruh 506 *test case* yang dieksekusi berhasil dengan *status PASS*, menghasilkan tingkat keberhasilan sebesar 100%. Hal ini menunjukkan bahwa sistem SiKosa telah berjalan sesuai dengan spesifikasi yang ditetapkan pada tahap *requirement analysis*. Tidak ditemukan satu pun kegagalan (*failure*) dalam eksekusi pengujian, baik pada level *unit* maupun integrasi. Rumus perhitungan tingkat keberhasilan:

$$\text{Test Success Rate} = \frac{\text{Jumlah Test Case Lulus}}{\text{Total Test Case}} \times 100\% \quad (1)$$

Rumus (1) digunakan untuk menghitung nilai *Test Success Rate*, yaitu persentase keberhasilan pengujian berdasarkan jumlah *test case* yang berhasil dijalankan sesuai dengan hasil yang diharapkan dibandingkan dengan keseluruhan *test case* yang dieksekusi. Semakin tinggi nilai *Test Success Rate*, semakin besar tingkat keberhasilan sistem dalam memenuhi skenario pengujian yang telah ditetapkan. Keterangan:

a) *Test Success Rate* : Persentase keberhasilan pengujian (%).

b) Jumlah *Test Case* Lulus : Jumlah *test case* yang memperoleh hasil sesuai dengan yang diharapkan.

- c) **Total Test Case** : Jumlah seluruh test case yang dieksekusi selama proses pengujian.

Substitusi nilai:

$$\text{Test Success Rate} = \frac{506}{506} \times 100\% = 100\%$$

Berdasarkan hasil perhitungan menggunakan Rumus (1), diperoleh nilai *Test Success Rate* sebesar 100%. Hasil tersebut menunjukkan bahwa seluruh 506 test case yang telah dieksekusi berhasil lulus sesuai dengan hasil yang diharapkan tanpa ditemukan kegagalan pengujian. Dengan demikian, dapat disimpulkan bahwa seluruh fungsionalitas sistem yang menjadi ruang lingkup pengujian telah berjalan sesuai dengan skenario pengujian yang telah dirancang, sehingga implementasi pengujian otomatis menggunakan *Jest* berhasil memverifikasi fungsionalitas sistem SiKosa secara keseluruhan.

b) Analisis Code Coverage

Code coverage merupakan metrik yang menunjukkan persentase kode yang telah diuji dan dipakai sebagai indikator kecukupan serta efektivitas *test suite*. Hasil *coverage* dari pengujian ini adalah:

- Statement Coverage* sebesar 82,13%, yaitu seluruh 2938 dari 3577 pernyataan kode telah tercakup oleh pengujian.
- Branch Coverage* sebesar 87,06%, yaitu sebanyak 397 dari 456 percabangan logika telah berhasil diuji. Hal ini menunjukkan bahwa mayoritas alur keputusan dalam sistem, seperti kondisi percabangan (if-else atau switch case), telah tervalidasi dengan baik.

- c. *Function Coverage* sebesar 84,87%, yaitu sebanyak 101 dari 119 fungsi telah dipanggil selama proses pengujian. Hal ini mengindikasikan bahwa sebagian besar fungsi utama sistem telah diuji, sementara fungsi yang belum tercakup umumnya merupakan fungsi pendukung seperti inisialisasi sistem atau modul yang hanya berjalan pada kondisi tertentu.
- d. *Line Coverage* sebesar 82,13%, yang menunjukkan bahwa 2938 dari 3577 baris kode telah dieksekusi selama pengujian. Nilai ini konsisten dengan *statement coverage*, yang menandakan bahwa cakupan pengujian pada level baris dan pernyataan berjalan selaras tanpa perbedaan signifikan.

Secara keseluruhan, hasil *code coverage* menunjukkan bahwa seluruh metrik berada pada rentang 82%–87%, yang sudah melampaui batas umum praktik pengujian yang telah ditetapkan sebesar >80%.

c) Perbandingan Pengujian *Unit* dan Integrasi

Pengujian pada sistem dilakukan melalui dua pendekatan utama, yaitu *unit testing* dan *integration testing*, yang masing-masing memiliki tujuan dan karakteristik berbeda namun saling melengkapi dalam memastikan kualitas sistem secara keseluruhan.

Tabel 4. 39 Perbandingan Hasil *Unit Testing* dan *Integration Testing*

Aspek	Unit Testing	Integration Testing
Jumlah Test Case	248	258
Test Case Lulus	248	258
Tingkat Keberhasilan	100%	100%
Durasi Rata-rata	< 1 detik/suite	5-15 detik/suite
Dependensi Database	Tidak (mocked)	Ya (real MongoDB)
Cakupan	Logika service	End-to-end API flow

Tabel 4.39 menunjukkan perbandingan *unit testing* dan *integration testing* dari beberapa aspek utama. *Unit testing* memiliki 248 *test case*, sedangkan *integration testing* 258 *test case*, dengan tingkat keberhasilan keduanya 100%.

Dari sisi performa, *unit testing* lebih cepat (<1 detik/*suite*) karena hanya menguji logika internal menggunakan *mock database*, sementara *integration testing* lebih lambat (5–15 detik/*suite*) karena menggunakan MongoDB asli dan melibatkan komunikasi antar layanan.

Dari aspek cakupan, *unit testing* berfokus pada pengujian fungsi atau *service* secara individual, sedangkan *integration testing* mencakup alur API *end-to-end*. Secara keseluruhan, *unit testing* memastikan kebenaran logika dasar, sementara *integration testing* memvalidasi interaksi antar komponen sistem secara menyeluruh.

d) Temuan Pengujian

Selama proses pengujian sistem, ditemukan beberapa temuan penting yang menggambarkan kualitas dan karakteristik implementasi sistem yang diuji.

- 1) Pertama, validasi *input* berjalan konsisten, di mana seluruh *input* tidak valid berhasil ditolak dengan respons HTTP 400 (*Bad Request*). Hal ini menunjukkan bahwa mekanisme validasi telah diterapkan dengan baik pada sisi backend.
- 2) Kedua, manajemen sesi pengguna berjalan stabil, termasuk pada skenario *login* berulang yang menghasilkan multiple *session*, serta penghapusan sesi yang berjalan benar saat proses *logout* dilakukan.
- 3) Ketiga, otorisasi berbasis peran (*RBAC*) telah berfungsi sesuai ekspektasi, di mana pembatasan akses antar peran berhasil diterapkan dengan baik.

Mahasiswa tidak dapat mengakses *endpoint* psikolog, dan sebaliknya, yang menunjukkan kontrol akses berjalan efektif.

- 4) Keempat, konsistensi data antar komponen terjaga dengan baik pada *integration testing*. Data yang disimpan ke *database* konsisten dengan respons API, termasuk proses seperti hashing *password*, pembuatan slug artikel, serta pembuatan *session* saat *login*.
- 5) Kelima, masih terdapat kode yang belum tercakup dalam pengujian, yaitu sekitar 10,79% *statement coverage*, yang umumnya berada pada file entry point (*index.ts*) dan beberapa *edge-case error* handler. Kondisi ini tergolong wajar karena bagian tersebut biasanya hanya dieksekusi pada kondisi tertentu atau saat inisialisasi *server*.

Secara keseluruhan, temuan pengujian menunjukkan bahwa sistem telah berjalan dengan baik pada aspek fungsional utama, dengan beberapa area minor yang masih dapat ditingkatkan pada cakupan pengujian.

e) Kesesuaian dengan Tujuan Penelitian

Berdasarkan hasil pengujian yang telah dilakukan, ketiga tujuan penelitian yang dirumuskan pada Bab I dapat dinyatakan tercapai dengan baik.

- 1) Pertama, implementasi pengujian otomatis menggunakan *Jest* berhasil diterapkan pada sistem SiKosa dan mencakup seluruh fitur utama sesuai dengan batasan masalah. Proses pengujian berjalan secara konsisten, dapat diulang, serta meningkatkan efisiensi karena tidak memerlukan intervensi manual.
- 2) Kedua, telah disusun dan dieksekusi sebanyak 506 *test case* terstruktur, yang mencakup skenario positif (*happy path*) dan skenario negatif (*error*

51

handling). Hasil pengujian menunjukkan bahwa seluruh fungsi sistem bekerja sesuai dengan kebutuhan pengguna yang telah ditentukan.

- 3) Ketiga, target *code coverage* berhasil dicapai dengan hasil *statement coverage* 82,13%, *branch coverage* 87,06%, *function coverage* 84,87%, dan *line coverage* 82,13%. Nilai ini menunjukkan bahwa sebagian besar kode sistem telah berhasil diverifikasi melalui pengujian otomatis, dengan tingkat ketercakupan yang cukup tinggi pada aspek logika, percabangan, maupun fungsi.

169

Secara keseluruhan, hasil pengujian ini menunjukkan bahwa seluruh tujuan penelitian telah terpenuhi, baik dari sisi implementasi pengujian otomatis, cakupan skenario pengujian, maupun tingkat *code coverage* yang dicapa

BAB V

PENUTUP

Bab ini menyajikan kesimpulan dari hasil penelitian pengujian otomatis pada sistem Konseling Mahasiswa (SiKosa) menggunakan *Jest*, serta saran yang ditujukan bagi pengembangan dan penelitian lanjutan.

5.1 Kesimpulan

Berdasarkan hasil implementasi dan pembahasan pengujian otomatis pada sistem SiKosa yang telah diuraikan pada Bab IV, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Pengujian otomatis menggunakan *Jest* berhasil diterapkan pada sistem SiKosa dengan mencakup seluruh fitur utama yang telah ditetapkan, yaitu autentikasi, manajemen profil mahasiswa dan psikolog, manajemen data user, konsultasi, *chat*, *chatbot*, serta manajemen artikel. Pengujian dilaksanakan secara sistematis mengikuti enam tahapan *Software Testing Life Cycle* (STLC), mulai dari *requirement analysis* hingga *test closure*.
2. Sebanyak 506 *test case* terstruktur berhasil disusun dan dieksekusi, yang terdiri atas 248 *unit* dan 258 *integration test* yang tersebar dalam 26 *test suite*. Seluruh *test case* mencakup skenario positif (*happy path*) maupun negatif (*error path*), serta dirancang dengan format atribut (*Scenario ID*, *Test Case ID*, *Scenario Description*, *Test Steps*, dan *Expected Result*).
3. Seluruh 506 *test case* yang dieksekusi berhasil dengan status *PASS*, sehingga tingkat keberhasilan pengujian (*test success rate*) mencapai 100%. Hal ini

menunjukkan bahwa sistem SiKosa telah berjalan sesuai dengan spesifikasi kebutuhan yang ditetapkan, tanpa ditemukan kegagalan baik pada level *unit* maupun *integrasi*.

4. Standar *test coverage* telah ditentukan dan tercapai dengan nilai *statement coverage* 82,13%, *branch coverage* 87,06%, *function coverage* 84,87%, dan *line coverage* 82,13%. Nilai *coverage* pada seluruh metrik yang berada di atas 80% menunjukkan bahwa tingkat pengujian sudah cukup komprehensif dan memadai untuk sistem berbasis web, dengan sebagian besar logika, percabangan, dan fungsi utama telah berhasil diverifikasi melalui pengujian otomatis.
5. Penyusunan *Requirement Traceability Matrix* (RTM) membuktikan seluruh kebutuhan sistem telah tertelusuri ke *test case* yang memverifikasinya, sehingga cakupan penelusuran kebutuhan mencapai 100% tanpa adanya *orphan requirement* maupun *orphan test case*.
6. Penerapan pengujian otomatis memberikan manfaat berupa proses verifikasi yang konsisten, dapat diulang kapan saja tanpa intervensi manual, serta mempermudah deteksi dini kesalahan ketika terjadi perubahan kode di masa mendatang.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan pertimbangan untuk pengembangan sistem dan penelitian selanjutnya:

1. Cakupan pengujian dapat ditingkatkan dengan menguji bagian kode yang belum tercakup, sehingga nilai *code coverage* dapat mendekati 100%.

112

2. Proses pengujian otomatis disarankan untuk diintegrasikan ke dalam *pipeline Continuous Integration/Continuous Deployment (CI/CD)*, misalnya menggunakan *GitHub Actions*, agar seluruh *test case* dijalankan secara otomatis pada setiap perubahan kode sebelum proses *deployment*.
3. Pengujian pada layanan eksternal seperti *chatbot (Groq API)* dapat dikembangkan lebih lanjut dengan pengujian terhadap kondisi nyata (*real API*), tidak hanya menggunakan *mock*, untuk memastikan keandalan integrasi dengan layanan pihak ketiga.
4. Dokumentasi *test case* dan hasil pengujian sebaiknya terus dipelihara dan diperbarui seiring penambahan fitur baru pada sistem SiKosa, agar kualitas perangkat lunak tetap terjaga secara berkelanjutan.

DAFTAR PUSTAKA

- [1] A. Sulma and A. Zacky W, “Pengaruh Kualitas Sistem Dan Layanan Terhadap Kepuasan Pengguna Aplikasi SIAPP di Kabupaten Sidenreng Rappang,” *JURNAL TERAPAN PEMERINTAHAN MINANGKABAU*, vol. 3, no. 2, pp. 155–175, Nov. 2023, doi: 10.33701/jtpm.v3i2.3673.
- [2] S. P. Ramadhani,) Farsya, A. Saputra, F. Dwiansyah, I. Veritawati, and R. Artikel, “Pengujian Sistem Informasi Akademik (NeoSiak) Berbasis Website Menggunakan Equivalence Partitioning dan Metode Black Box INFO ARTIKEL ABSTRAK,” vol. 3, no. 1, p. 18, 2024, doi: 10.55123.
- [3] F. Widiyanto, U. Islam Syekh Yusuf, and J. Maulana Yusuf Kota Tangerang, “Implementasi Model SDLC Dalam Perancangan Sistem Informasi Manajemen Perpustakaan Berbasis Web,” 2024, doi: 10.69533.
- [4] J. Sains, T. dan Kesehatan, R. Amaliyah Kamisr, and R. Artikel, “Studi Komparatif Pengujian Manual Dan Pengujian Otomatis Dengan Cypress Pada Website,” vol. 2, no. 2, pp. 354–364, 2025, doi: 10.62335.
- [5] J. Lian Min, A. Istiqomah, A. Rahmani, P. Negeri Bandung, and P. P. Tester Padepokan Tujuh Sembilan-Bandung, “Evaluasi Penggunaan Manual Dan Automated Software Testing Pada Pelaksanaan End-To-End Testing,” *Jurnal Teknologi Terapan* |, vol. 6, no. 1, 2020.
- [6] F. Dhimas Wicaksono, “Rancang Bangun Automation Test Journey Pada E-Commerce,” 2022.
- [7] G. A. Prasetyo and N. Setiani, “Perbandingan Kinerja Framework Automation UI Testing Menggunakan The Distance to The Ideal Alternative,” *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 1, pp. 224–237, Jan. 2025, doi: 10.29100/jipi.v10i1.5760.
- [8] N. Hashemi, A. Tahir, S. Rasheed, A. Shi, and R. Blagojevic, “JS-TOD: Detecting Order-Dependent Flaky Tests in Jest,” Aug. 2025, [Online]. Available: <http://arxiv.org/abs/2509.00466>
- [9] Rahman Abdillah, Rudi Hermawan, Wawan Hermawansyah, Ibnu Adkha, and Heri Arifin, “Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket,” *Polygon : Jurnal Ilmu Komputer dan Ilmu Pengetahuan Alam*, vol. 2, no. 4, pp. 166–175, Jul. 2024, doi: 10.62383/polygon.v2i4.199.
- [10] M. Nur Ichsanudin, M. Yusuf, S. Jurusan Rekayasa Sistem Komputer, J. Teknik Industri, I. AKPRIND Yogyakarta, and R. Artikel, “Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan

- Metode Black Box Testing Bagi Pemula Info Artikel Abstrak,” vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123.
- [11] A. Valerian, D. Bernady, F. Wahyudi, J. P. Dinatha, and D. Barletyano, “Implementasi Unit Testing, Integration Testing, System Testing, Dan Validation Testing Pada Aplikasi Berbasis Website (Studi Kasus: Kafe Saturdays),” 2025.
 - [12] Rahmat Fauzan, Ferina Putri Soedjono, Annisa Ayu Permadani, and Muhammad Ainul Yakin, “Perbandingan Pengujian Manual dan Terotomasi pada Software Enterprise Resource Planning,” *Journal of Advances in Information and Industrial Technology*, vol. 5, no. 1, pp. 23–30, May 2023, doi: 10.52435/jaiit.v5i1.318.
 - [13] S. Melia and F. Profesio Putra, “Analisis Perbandingan Tools Pengujian Otomatis pada GUI Aplikasi Berbasis WEB,” 2023. [Online]. Available: <https://journal.irpi.or.id/index.php/sentimas>
 - [14] M. Hilman and W. Mantiri, “Automation and Selection Technique for Regression Testing: An Empirical Analysis,” *The Indonesian Journal of Computer Science*, vol. 14, no. 4, Aug. 2025, doi: 10.33022/ijcs.v14i4.4967.
 - [15] A. N. Hasibuan and T. Dirgahayu, “Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna,” 2022.
 - [16] G. A. Jasmin, D. Galih, and P. Putri, “Komparasi Metode Automasi dan Hybrid pada Pengujian Aplikasi Mobile Webrtc Menggunakan Appium,” *Journal of Internet and Software Engineering*, vol. 5, no. 1, 2024.
 - [17] A. Maspupah, A. Rahmani, J. L. Min, and T. A. Roshinta, “Software Testing in the Indonesian Industry: Survey of Methods, Tools, and Documentation ARTICLE INFORMATION ABSTRACT,” 2024. [Online]. Available: <http://innovatics.unsil.ac.id>
 - [18] A. Jailani and M. A. Yaqin, “Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik Boundary Value Analysis,” *Journal Automation Computer Information System*, vol. 4, no. 2, pp. 60–66, Jul. 2024, doi: 10.47134/jacis.v4i2.78.
 - [19] I. A. Shaleh, J. P. Yogi, P. Pirdaus, R. Syawal, and A. Saifudin, “Pengujian Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalent Partitions,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 4, no. 1, p. 38, Jan. 2021, doi: 10.32493/jtsi.v4i1.8960.
 - [20] D. Widhyaestoeti *et al.*, “Black Box Testing Equivalence Partitions Untuk Pengujian Front-End Pada Sistem Akademik Sitoda,” 2021.

- [21] M. Mintarsih, “Pengujian Black Box Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 33–35, Feb. 2023, doi: 10.47233/jteksis.v5i1.727.
- [22] M. Jibril and M. Amin, “Pengujian Sistem Informasi E-Modul Pada Smpn 1 Tempuling Menggunakan Black Box Testing 1,” 2024.
- [23] D. Yuliatwati and A. Andriyadi, “Pengujian Sistem Informasi E-Monitoring Pengelolaan Pembangunan Desa Dengan Menggunakan Metode Blackbox Testing,” *IJCCS*, vol. x, No.x, no. 93, 2022.
- [24] J. Shadiq, A. Safei, R. Wahyudin Ratu Loly, C. sitasi, L. Rwr, and P. Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing, “Information Management For Educators And Professionals Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing,” *Information Management for Educators and Professionals*, vol. 5, no. 2, pp. 97–110, 2021.
- [25] A. Samdono, A. P. Sari, and F. P. Aditiawan, “Pengujian Black Box Pada Sistem Informasi Stok Dan Penjualan Berbasis Website Menggunakan Metode Equivalence Partitioning (Studi Kasus: Cv. Algani Karya Mandiri),” 2024.
- [26] A. Amalia, S. W. Putri Hamidah, and T. Kristanto, “Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web,” *Building of Informatics, Technology and Science (BITS)*, vol. 3, no. 3, pp. 269–274, Dec. 2021, doi: 10.47065/bits.v3i3.1062.
- [27] F. S. Kinasih, H. Nur Gian, and H. Permatasari, “Pengujian Sistem Informasi Inventaris Barang Berbasis Website Menggunakan Metode Whitebox Testing,” 2023.
- [28] P. Straubinger, T. Fulcini, G. Fraser, and M. Torchiano, “IntelliGame in Action: An Experience Report on Gamifying JavaScript Unit Tests,” in *Proceedings - 2024 1st IDE Workshop, IDE 2024*, Association for Computing Machinery, Inc, Aug. 2024, pp. 110–114. doi: 10.1145/3643796.3648466.
- [29] N. Hashemi, A. Tahir, S. Rasheed, A. Shi, and R. Blagojevic, “Detecting and Evaluating Order-Dependent Flaky Tests in JavaScript,” Jan. 2025, [Online]. Available: <http://arxiv.org/abs/2501.12680>
- [30] Ruliansyah, Tukino, Baenil Huda, and April Lia Hananto, “Penerapan Software Testing Life Cycle Pada Pengujian Otomatisasi Platform Dzikra,”

- CSRID (Computer Science Research and Its Development Journal)*, vol. 15, no. 1, pp. 01–11, May 2023, doi: 10.22303/csrid.15.1.2023.01-11.
- [31] A. Arfan, “Penerapan STLC dalam Pengujian Automation Aplikasi Mobile (Studi kasus: LMS Amikom Center PT.GIT Solution),” 2022.
 - [32] A. N. Hasibuan and T. Dirgahayu, “Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna,” 2021.
 - [33] A. N. Hasibuan and T. Dirgahayu, “Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna.”
 - [34] F. W. Azhari and D. F. Suyatno, “Pengujian Otomatis GUI dengan Katalon Studio pada Situs Lowongan Kerja Jobstreet dan Glints,” 2024.
 - [35] I. Gusti, N. Putu, D. Dicky Diastama, I. Made Sukarsa, N. Kadek, and A. Wirdiani, “Pengembangan Test Script untuk Load Testing Web dengan metode Software Testing Life Cycle,” 2021.
 - [36] X. Y. Djam, N. V. Blamah, and M. E. Ezema, “A Comparative Evaluation of Test Coverage Techniques Effectiveness,” *Journal of Software Engineering and Applications*, vol. 14, no. 04, pp. 95–109, 2021, doi: 10.4236/jsea.2021.144007.
 - [37] R. White and J. Krinke, “TCTracer: Establishing test-to-code traceability links using dynamic and static techniques,” *Empir. Softw. Eng.*, vol. 27, no. 3, May 2022, doi: 10.1007/s10664-021-10079-1.

LAMPIRAN**Lampiran 1: Surat izin penelitian**

 **SISTEM KONSELING MAHASISWA**
Jalan Pangkalan Tujuh, RT. 06 Sungai Salak Indragiri Hilir Provinsi Riau Indonesia
No Hp: 0895 3090 0011 Gmail: sikosa.project01@gmail.com

SURAT PEMBERIAN IZIN PENELITIAN

Nomor : 01/SiKosa/A/VII/2026
Perihal : Surat Pemberian Izin Penelitian

Kepada Yth,
Saudara Muhammad Aidil
Mahasiswa Universitas Islam Indragiri (UNISI)
di _____
Tempat _____

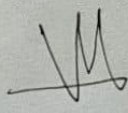
Dengan hormat,
Sehubungan dengan permohonan saudara untuk melaksanakan penelitian pada Tim Konseling Mahasiswa, dengan ini kami memberi izin kepada:

Nama : Muhammad Aidil
NIM : 403221010060
Program Studi : Sistem Informasi
Fakultas : Teknik dan Ilmu Komputer
Perguruan Tinggi : Universitas Islam Indragiri (UNISI)

Untuk melaksanakan penelitian dalam rangka penyusunan tugas akhir dengan judul **"Automation Testing Pada Website Konseling Mahasiswa (SiKosa) Menggunakan Jest"** Izin ini diberikan untuk memperoleh data dan informasi yang diperlukan selama proses penelitian dengan tetap mematuhi ketentuan yang berlaku. Seluruh data dan informasi yang diperoleh hanya digunakan untuk kepentingan akademik serta wajib menjaga kerahasiaan data yang bersifat internal.

Demikian surat pemberian izin penelitian ini dibuat agar dapat dipergunakan sebagaimana mestinya. Atas perhatian dan kerja sama yang baik, kami ucapkan terima kasih.

Tembilahan, 10 Juli 2026
Hormat kami,
Wakil Tim Sistem Konseling Mahasiswa

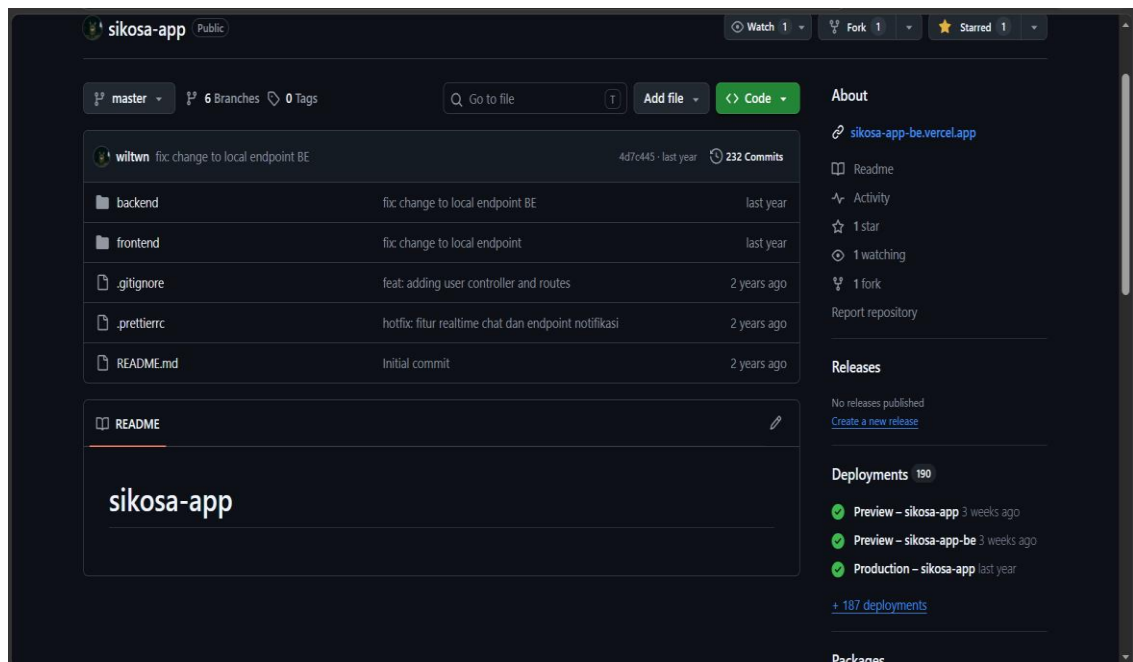

WILDAN TAUFIK WIBOWO NASUTION

115

Lampiran 2: Dokumentasi penelitian



Gambar 1. Zoom meeting izin penelitian dengan stakeholder/tim sikosa



Gambar 2. Izin akses data sikosa