



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 mengungkap 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.

1. Dilarang memperbanyak atau mendistribusikan dokumen ini untuk tujuan komersial tanpa izin tertulis dari penulis atau pihak berwenang.

2. Penggunaan tanpa izin untuk kepentingan akademik, penelitian, dan pendidikan diperbolehkan dengan mencantumkan sumber.

3. Plagiarisme juga dilarang dan dapat dikenakan sanksi.

3. Universitas hanya berhak menyimpan dan mendistribusikan dokumen ini di repositori akademik, tanpa mengalihkan hak cipta penulis, sesuai dengan peraturan yang berlaku di Indonesia.



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.

