



BAB III

METODE PENELITIAN

3.1 Pendahuluan

Penelitian ini menggunakan metode *System Development Life Cycle* (SDLC) dengan model Waterfall sebagai metode pengembangan sistem. Model Waterfall dipilih karena memiliki tahapan yang terstruktur, sistematis, dan berurutan, sehingga setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Pendekatan ini sesuai untuk pengembangan Sistem Pendukung Keputusan (SPK) karena kebutuhan sistem telah ditentukan sejak awal berdasarkan studi literatur[19].

Dalam penelitian ini, analisis kebutuhan sistem dilakukan melalui studi literatur terhadap jurnal dan penelitian terdahulu yang membahas sistem pendukung keputusan pemilihan lokasi usaha. Selanjutnya, kebutuhan sistem dianalisis menggunakan *PIECES Framework* untuk mengidentifikasi kebutuhan fungsional dan nonfungsional. Tahap perancangan dilakukan menggunakan *Unified Modeling Language* (UML), kemudian sistem diimplementasikan menggunakan Flutter sebagai framework pengembangan aplikasi mobile dan Firebase sebagai basis data. Proses pengambilan keputusan menggunakan metode *Rank Order Centroid* (ROC) untuk menentukan bobot setiap kriteria, sedangkan metode *Simple Additive Weighting* (SAW) digunakan untuk menghitung nilai preferensi dan menghasilkan rekomendasi lokasi usaha terbaik. Setelah implementasi selesai, sistem diuji menggunakan *Black Box Testing* dan *System Usability Scale* (SUS) untuk memastikan fungsi sistem berjalan dengan baik dan mudah digunakan oleh pengguna[20].

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

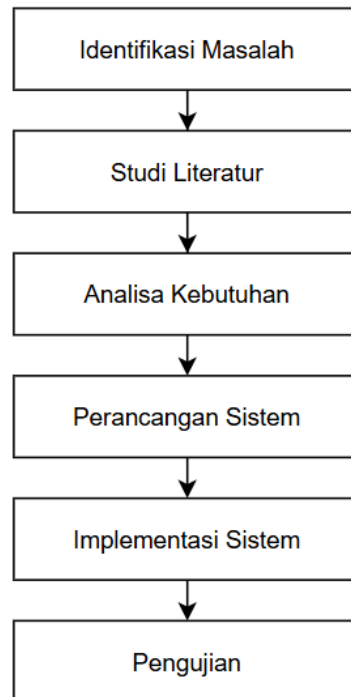
2. Penggunaan tanpa izin untuk kepentingan akademik, penelitian, dan pendidikan diperbolehkan dengan mencantumkan sumber.

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.

3.2 Kerangka Penelitian

Kerangka penelitian merupakan tahapan yang menggambarkan proses penelitian secara sistematis mulai dari identifikasi permasalahan hingga diperoleh hasil penelitian. Kerangka ini disusun agar seluruh proses pengembangan sistem dilakukan secara terarah sesuai dengan model Waterfall. Tahapan penelitian yang dilakukan meliputi identifikasi masalah, studi literatur, analisis kebutuhan, perancangan sistem, implementasi aplikasi, pengujian sistem, serta evaluasi hasil penelitian. Adapun tahapan penelitian sebagai berikut:



Gambar 3.1 Kerangka Penelitian

3.2.1 Requirements (Analisa Kebutuhan)

Tahapan awal dalam pengembangan sistem adalah melakukan analisis kebutuhan secara komprehensif. Pada fase ini, peneliti menggunakan metode *PIECES Framework* untuk mengidentifikasi permasalahan yang ada pada sistem



berjalan dan menentukan kebutuhan Sistem Pendukung Keputusan Pemilihan Tempat Usaha Metode ROC Hybrid SAW Berbasis Mobile yang akan dikembangkan.

PIECES merupakan kerangka analisis yang terdiri dari enam dimensi evaluasi, yaitu *Performance* (kinerja), *Information* (informasi), *Economy* (ekonomi), *Control* (pengendalian), *Efficiency* (efisiensi), dan *Service* (layanan).

Melalui pendekatan ini, dilakukan identifikasi terhadap:

1. *Performance Analysis*: Menganalisis kinerja proses pemilihan lokasi usaha yang sedang berjalan, termasuk waktu yang dibutuhkan pelaku usaha untuk melakukan survei lokasi, kecepatan dalam mengevaluasi berbagai alternatif lokasi, dan akurasi penilaian yang dihasilkan dari metode konvensional yang masih bersifat intuitif.
2. *Information Analysis*: Mengevaluasi kualitas informasi yang digunakan dalam pengambilan keputusan pemilihan lokasi usaha, meliputi kelengkapan data kriteria lokasi (aksesibilitas, kepadatan penduduk, tingkat persaingan, harga lahan, potensi pasar, infrastruktur), akurasi informasi yang tersedia, dan ketepatan waktu penyediaan informasi untuk mendukung keputusan.
3. *Economic Analysis*: Mengkaji aspek ekonomi dari proses pemilihan lokasi, termasuk biaya yang dikeluarkan pelaku usaha untuk survei dan evaluasi lokasi secara manual, potensi kerugian finansial akibat kesalahan pemilihan lokasi, dan efisiensi biaya yang dapat dicapai melalui implementasi sistem pendukung keputusan terkomputerisasi.





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

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

2. Penggunaan tanpa izin untuk kepentingan komersial atau pelanggaran hak cipta dapat dikenakan sanksi sesuai dengan UU Hak Cipta di Indonesia.

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.

4. *Control Analysis*: Memeriksa mekanisme pengendalian dalam proses pemilihan lokasi, seperti konsistensi dalam penetapan prioritas kriteria, validasi data alternatif lokasi, dan mekanisme untuk memastikan objektivitas dalam pengambilan keputusan tanpa bias subjektif yang berlebihan.
5. *Efficiency Analysis*: Menilai efisiensi proses pemilihan lokasi usaha yang berjalan, mulai dari proses identifikasi alternatif lokasi, pengumpulan data kriteria, penentuan urutan prioritas kriteria, hingga penentuan *ranking* lokasi terbaik. Analisis ini juga mencakup identifikasi redundansi proses dan peluang otomasi.
6. *Service Analysis*: Mengidentifikasi tingkat layanan yang dibutuhkan oleh pelaku usaha sebagai pengguna sistem, termasuk kemudahan akses sistem melalui perangkat mobile, kejelasan informasi hasil analisis, kemudahan dalam melakukan input data dan preferensi, serta responsivitas sistem dalam memberikan rekomendasi[21].

3.2.2 Design (Perencanaan Sistem)

Setelah kebutuhan sistem teridentifikasi dengan jelas, tahapan selanjutnya adalah merancang arsitektur dan desain sistem menggunakan *Unified Modeling Language* (UML). UML dipilih sebagai bahasa pemodelan karena kemampuannya dalam menggambarkan struktur dan perilaku sistem secara visual dan terstandar.

Pada fase perancangan ini, dibuat beberapa diagram UML yang saling terintegrasi untuk memberikan gambaran menyeluruh tentang sistem yang akan dibangun:

1. *Use case Diagram*



Use Case Diagram dirancang untuk menggambarkan interaksi antara pengguna dengan sistem. Diagram ini mengidentifikasi dua peran utama pengguna yaitu administrator dan pelaku usaha. Administrator memiliki akses untuk mengelola kriteria, mengelola sub-kriteria, mengelola alternatif lokasi, mengelola pengguna, dan melihat laporan. Sementara itu, pelaku usaha sebagai pengguna utama dapat melakukan login, menginput urutan prioritas (*ranking*) kriteria pemilihan lokasi, menginput nilai alternatif lokasi pada setiap kriteria, melihat hasil perhitungan bobot kriteria dengan metode ROC, melihat *ranking* lokasi hasil perhitungan SAW, dan melihat rekomendasi lokasi usaha terbaik. Output dari diagram ini memberikan identifikasi lengkap mengenai fungsionalitas sistem dari perspektif pengguna serta batasan akses untuk setiap peran[22].

2. *Activity Diagram*

Activity Diagram dibuat untuk memodelkan alur kerja dari setiap proses bisnis dalam sistem pendukung keputusan. Diagram ini menjelaskan langkah-langkah detail yang terjadi dalam beberapa proses utama sistem. Proses pertama adalah perhitungan bobot kriteria menggunakan metode ROC, dimulai dari pengguna menginput urutan prioritas kepentingan kriteria (dari yang paling penting hingga paling kurang penting), kemudian sistem menghitung bobot setiap kriteria menggunakan rumus centroid, hingga sistem menampilkan bobot akhir kepada pengguna. Proses kedua adalah penilaian dan perbandingan alternatif lokasi menggunakan metode SAW, yang dimulai dari input nilai alternatif pada setiap kriteria, normalisasi matriks keputusan, perkalian dengan bobot ROC, hingga

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

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

2. Penggunaan tanpa izin untuk kepentingan komersial atau pelanggaran hak cipta dapat dikenakan sanksi sesuai dengan UU Hak Cipta di Indonesia.

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.

menghasilkan *ranking* akhir. Proses ketiga adalah pemberian rekomendasi lokasi usaha kepada pengguna berdasarkan hasil perhitungan *Hybrid ROC-SAW* tersebut. Output yang dihasilkan berupa visualisasi proses bisnis yang terstruktur dengan titik keputusan, validasi, dan aktivitas yang jelas sehingga memudahkan pemahaman alur kerja sistem.

3. **Sequence Diagram**

Sequence Diagram dirancang untuk mendeskripsikan interaksi antar objek dalam sistem berdasarkan urutan waktu. Diagram ini menggambarkan bagaimana pertukaran pesan terjadi antara objek-objek seperti interface pengguna, controller ROC, controller SAW, dan database dalam menjalankan fungsi sistem. Beberapa sequence diagram yang dibuat mencakup proses perhitungan ROC yang menunjukkan bagaimana data urutan *ranking* kriteria diolah menjadi bobot kriteria menggunakan rumus centroid, proses perhitungan SAW yang menggambarkan alur dari normalisasi nilai alternatif, pembobotan menggunakan hasil ROC, hingga menghasilkan *ranking*, serta proses login dan autentikasi yang menjelaskan mekanisme keamanan akses sistem. Output yang dihasilkan memberikan pemahaman mendalam tentang bagaimana objek-objek dalam sistem berkolaborasi untuk menyelesaikan proses analisis dan pengambilan keputusan[23].

4. **Class Diagram**

Class Diagram dirancang untuk menampilkan struktur statis sistem yang terdiri dari kelas-kelas, atribut, metode, serta relasi antar kelas. Diagram ini menjadi blueprint untuk implementasi basis data dan struktur program.





Kelas-kelas utama yang dirancang meliputi User yang menyimpan informasi pengguna seperti `id_user`, `nama`, `email`, `password`, dan `role`. Kelas Kriteria menyimpan informasi kriteria pemilihan lokasi dengan atribut `id_kriteria`, `nama_kriteria`, dan `jenis_kriteria` (*benefit/cost*). Kelas SubKriteria menyimpan detail sub-kriteria yang terhubung dengan kriteria induknya. Kelas Alternatif menyimpan informasi lokasi yang akan dinilai termasuk `nama_lokasi`, `alamat`, dan `koordinat`. Kelas *RankingKriteria* menyimpan urutan prioritas kriteria yang diinput oleh pengguna sebagai dasar perhitungan ROC. Kelas *BobotROC* menyimpan hasil perhitungan bobot kriteria menggunakan rumus *Rank Order Centroid* beserta urutan *rank* yang digunakan. Kelas *PenilaianAlternatif* menyimpan nilai penilaian setiap alternatif terhadap masing-masing kriteria. Kelas *HasilRanking* menyimpan skor akhir hasil perhitungan SAW dan *ranking* dari setiap alternatif lokasi. Output dari class diagram ini adalah skema database dan struktur berorientasi objek yang akan diimplementasikan dalam kode program.

5. **Deployment Diagram**

Deployment Diagram dirancang untuk mengilustrasikan arsitektur fisik sistem berbasis mobile. Diagram ini menunjukkan bagaimana komponen perangkat lunak didistribusikan pada infrastruktur teknologi informasi. Arsitektur sistem terdiri dari *Mobile Device* baik Android maupun iOS yang berisi aplikasi Flutter sebagai antarmuka sekaligus logika aplikasi (*client-side*), *Firebase Authentication* yang menangani proses registrasi, login, dan manajemen sesi pengguna, *Cloud Firestore* sebagai database NoSQL

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



berbasis cloud yang menyimpan seluruh data sistem termasuk data kriteria, alternatif, *ranking* prioritas, bobot ROC, dan hasil perhitungan SAW secara real-time, serta *Firebase Cloud Storage* yang digunakan untuk menyimpan data pendukung seperti gambar lokasi dan dokumen. Seluruh layanan tersebut berjalan di atas infrastruktur *Google Cloud Platform* sehingga sistem tidak memerlukan server backend terpisah (*serverless architecture*). Output yang dihasilkan berupa spesifikasi infrastruktur dan konfigurasi deployment sistem berbasis mobile-cloud yang jelas dan terstruktur.

3.2.3 *Impelementation (Implementasi)*

Tahapan implementasi merupakan realisasi dari hasil analisis dan perancangan menjadi Sistem Pendukung Keputusan Pemilihan Tempat Usaha Metode ROC Hybrid SAW Berbasis Mobile yang dapat dioperasikan. Pada fase ini dilakukan pengkodean (*coding*) berdasarkan desain yang telah dibuat pada tahap sebelumnya. Implementasi sistem dilakukan dengan menggunakan *Flutter* sebagai *Framework* pengembangan aplikasi mobile dan *Firebase Cloud Firestore* sebagai basis data *backend*, sehingga sistem dibangun dengan arsitektur *serverless* yang ringan, *real-time*, dan tidak memerlukan pengelolaan server *backend* secara mandiri.

Pengembangan aplikasi mobile dilakukan menggunakan *Framework* Flutter dengan bahasa pemrograman Dart untuk memastikan aplikasi dapat berjalan pada satu basis kode (*single codebase*) baik di platform Android maupun iOS. Komponen yang dikembangkan mencakup antarmuka pengguna (*widget*) yang disesuaikan dengan desain mockup, manajemen status (*state management*) menggunakan Provider, Riverpod, atau BLoC untuk mengelola alur data antar

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

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

2. Penggunaan tanpa izin untuk kepentingan komersial atau pelanggaran hak cipta dapat dikenakan sanksi sesuai dengan UU Hak Cipta di Indonesia.

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.

halaman secara terstruktur, navigasi dan routing antar halaman yang smooth dan intuitif, serta integrasi langsung dengan Firebase melalui plugin resmi seperti `firebase_core`, `cloud_firestore`, dan `firebase_auth`. Aplikasi juga memanfaatkan kemampuan *offline persistence* bawaan *Cloud Firestore* sehingga data yang pernah dimuat tetap dapat diakses tanpa koneksi internet dan akan tersinkronisasi otomatis ke server saat koneksi tersedia, serta dilengkapi implementasi validasi input dan error handling untuk memastikan pengalaman pengguna yang baik.

Autentikasi dan otorisasi pengguna diimplementasikan menggunakan *Firebase Authentication*, yang menyediakan mekanisme login dan registrasi berbasis email dan kata sandi maupun *Google Sign-In*. *Firebase Authentication* terintegrasi langsung dengan *Cloud Firestore Security Rules* untuk membatasi hak akses data sesuai peran pengguna, yaitu administrator dan pelaku usaha, sehingga setiap pengguna hanya dapat mengakses dan mengubah data yang menjadi kewenangannya tanpa memerlukan token otorisasi tambahan seperti JWT pada arsitektur REST API konvensional.

Struktur data pada *Cloud Firestore* dirancang dalam bentuk koleksi (*collection*) dan dokumen (*document*) yang merepresentasikan kelas-kelas pada class diagram. Koleksi utama yang dibuat meliputi koleksi `users` untuk menyimpan data pengguna, koleksi kriteria dan sub-koleksi subkriteria untuk menyimpan data kriteria pemilihan lokasi, koleksi alternatif untuk menyimpan data lokasi usaha, koleksi `ranking_kriteria` untuk menyimpan urutan prioritas kriteria yang diinput setiap pengguna, koleksi `bobot_roc` untuk menyimpan hasil perhitungan bobot metode ROC, koleksi `penilaian_alternatif` untuk menyimpan nilai setiap alternatif pada masing-masing kriteria, serta koleksi `hasil_ranking` untuk menyimpan skor



akhir dan *ranking* hasil perhitungan SAW. Firestore Security Rules diterapkan pada setiap koleksi untuk menjaga integritas dan keamanan data, sementara composite index dibuat pada field yang sering digunakan dalam query bertingkat agar performa pengambilan data tetap optimal.

Implementasi algoritma ROC dan SAW dilakukan dalam bentuk class atau service Dart pada sisi aplikasi *Flutter* (*client-side computation*), yang memproses data secara langsung dari stream *Cloud Firestore* menggunakan *StreamBuilder* atau *Future* sehingga hasil perhitungan dapat diperbarui secara real-time begitu data alternatif atau prioritas kriteria berubah. Untuk algoritma *Rank Order Centroid* (ROC), dikembangkan fungsi untuk menerima input urutan prioritas kriteria dari pengguna (kriteria diurutkan dari yang paling penting ke yang paling kurang penting) yang disimpan pada koleksi *ranking_kriteria*, menetapkan nilai *rank* *k* untuk setiap kriteria ($k = 1, 2, 3, \dots, n$), menghitung bobot setiap kriteria menggunakan rumus centroid:

$$w_k = (1/n) \times \sum (1/i), \text{ untuk } i = k \text{ sampai } n$$

dengan *n* adalah jumlah kriteria dan *k* adalah posisi *rank* kriteria tersebut, melakukan validasi bahwa total keseluruhan bobot yang dihasilkan sama dengan satu, kemudian menuliskan (*write*) hasil bobot tersebut ke koleksi *bobot_roc* pada Firestore agar dapat digunakan pada perhitungan SAW maupun ditampilkan kembali kepada pengguna. Metode ROC dipilih karena tidak memerlukan proses perbandingan berpasangan maupun pengujian rasio konsistensi seperti pada metode pembobotan konvensional, sehingga proses input data oleh pengguna menjadi lebih sederhana, cepat, dan tetap objektif karena bobot dihasilkan secara matematis dari urutan prioritas yang diberikan. Sedangkan untuk algoritma SAW, dikembangkan



class Dart untuk melakukan normalisasi matriks keputusan yang membedakan antara benefit dan *cost criteria* berdasarkan data pada koleksi penilaian_alternatif, perhitungan skor preferensi dengan pembobotan yang diambil dari koleksi bobot_roc, perangkingan alternatif berdasarkan skor akhir, dan penyimpanan hasil rekomendasi lokasi terbaik ke koleksi hasil_ranking beserta penjelasannya. Untuk kebutuhan komputasi yang lebih berat atau agregasi data lintas pengguna, logika ROC dan SAW juga dapat dipindahkan ke *Cloud Functions for Firebase* sebagai lapisan logika sisi server yang dipicu (*trigger*) secara otomatis setiap kali terjadi perubahan data pada *Firestore*.

Integrasi seluruh komponen dilakukan dengan menghubungkan aplikasi Flutter secara langsung dengan layanan-layanan Firebase (*Firestore*, *Authentication*, dan *Cloud Storage*) melalui Firebase SDK, tanpa memerlukan lapisan REST API kustom sebagai perantara. Proses integrasi meliputi konfigurasi project Firebase untuk platform Android dan iOS melalui berkas *google-services.json* dan *GoogleService-Info.plist*, penerapan *Firestore Security Rules* sebagai pengganti fungsi otorisasi backend konvensional, pengujian *real-time listener* untuk memastikan perubahan data pada satu perangkat langsung tersinkronisasi ke perangkat lain, implementasi *error handling* khusus untuk menangani kondisi offline dan kegagalan sinkronisasi, serta pemanfaatan *Firebase Cloud Storage* untuk menyimpan data pendukung seperti foto lokasi usaha.

Output dari tahapan implementasi adalah aplikasi Sistem Pendukung Keputusan Pemilihan Tempat Usaha Metode ROC Hybrid SAW Berbasis Mobile yang dibangun menggunakan Flutter dan terintegrasi penuh dengan *Firebase Cloud Firestore*, serta siap untuk dilakukan pengujian. Sistem yang dihasilkan memiliki



fitur-fitur lengkap seperti manajemen kriteria dan sub-kriteria pemilihan lokasi, input urutan prioritas kriteria menggunakan metode ROC, perhitungan bobot kriteria secara otomatis melalui rumus centroid, input data alternatif lokasi di Kecamatan Tembilahan, penilaian setiap alternatif berdasarkan kriteria yang telah ditentukan, perhitungan *ranking* menggunakan metode SAW dengan bobot hasil ROC, sinkronisasi data secara real-time antar perangkat melalui *Cloud Firestore*, dukungan akses offline dengan sinkronisasi otomatis, visualisasi hasil analisis dalam bentuk tabel dan grafik yang informatif, rekomendasi lokasi usaha terbaik dengan penjelasan detail, serta fitur export hasil analisis dalam format PDF atau Excel untuk dokumentasi dan presentasi.

3.2.4 Verification (Pengujian)

Tahap Verification merupakan proses pengujian sistem untuk memastikan bahwa aplikasi Sistem Pendukung Keputusan (SPK) yang telah dikembangkan dapat berfungsi sesuai dengan kebutuhan dan tujuan penelitian. Pengujian dilakukan setelah tahap implementasi selesai untuk mengetahui apakah seluruh fitur aplikasi telah berjalan dengan baik serta menghasilkan keluaran yang sesuai dengan rancangan sistem. Pengujian fungsional dilakukan menggunakan metode Black Box Testing, yaitu dengan menguji setiap fitur aplikasi berdasarkan masukan (*input*) dan keluaran (*output*) yang diharapkan tanpa memperhatikan struktur kode program. Pengujian meliputi proses autentikasi pengguna, pengelolaan data kriteria, pengelolaan data alternatif, input nilai alternatif, proses perhitungan bobot menggunakan metode Rank Order Centroid (ROC), proses perhitungan nilai preferensi menggunakan metode Simple Additive Weighting (SAW), hingga proses menampilkan hasil perbandingan lokasi usaha.

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

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

2. Penggunaan tanpa izin untuk kepentingan komersial atau pelanggaran hak cipta dapat dikenakan sanksi sesuai dengan UU Hak Cipta di Indonesia. 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.





Selain pengujian fungsional, dilakukan juga pengujian terhadap hasil perhitungan metode ROC dan SAW dengan membandingkan hasil perhitungan yang dihasilkan oleh aplikasi dengan hasil perhitungan manual menggunakan data yang sama. Pengujian ini bertujuan untuk memastikan bahwa proses pembobotan, normalisasi, perhitungan nilai preferensi, dan perangkingan alternatif telah diimplementasikan sesuai dengan konsep teoritis metode ROC dan SAW. Dengan demikian, aplikasi yang dikembangkan diharapkan mampu menghasilkan rekomendasi lokasi usaha secara objektif, akurat, dan sesuai dengan perhitungan yang telah dirancang.

3.2.5 Maintenance (Pemeliharaan)

Tahapan maintenance merupakan fase berkelanjutan yang dilakukan setelah sistem diimplementasikan dan digunakan oleh pengguna. Pemeliharaan sistem pendukung keputusan menjadi aspek krusial untuk memastikan sistem tetap berfungsi optimal, akurat, dan sesuai dengan kebutuhan yang terus berkembang seiring dengan dinamika bisnis dan teknologi.

Aktivitas pemeliharaan korektif dilakukan untuk memperbaiki bug atau error yang ditemukan selama sistem beroperasi. Perbaikan meliputi error dalam perhitungan algoritma ROC atau SAW yang mungkin muncul pada kasus tertentu, bug dalam interface aplikasi mobile yang mengganggu pengalaman pengguna, masalah koneksi antara aplikasi mobile dengan backend, serta error handling yang tidak sempurna. Tim pengembang menyediakan mekanisme pelaporan bug melalui fitur *feedback* dalam aplikasi dan melakukan perbaikan secara berkala dengan merilis patch atau update aplikasi ke pengguna.

Pemeliharaan

1. Dilarang memperbanyak atau mendistribusikan dokumen ini untuk tujuan komersial tanpa izin tertulis dari penulis atau pihak berwenang. Penggunaan untuk kepentingan akademik, penelitian, dan pendidikan diperbolehkan dengan mencantumkan sumber.

2. Penggunaan tanpa izin untuk kepentingan komersial atau pelanggaran hak cipta dapat dikenakan sanksi sesuai dengan UU Hak Cipta di Indonesia. 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.

perfektif dilakukan untuk meningkatkan kualitas sistem melalui penambahan fitur baru atau optimasi fitur yang sudah ada berdasarkan *feedback* pengguna. Kegiatan ini mencakup penambahan kriteria atau sub-kriteria baru berdasarkan masukan stakeholder, implementasi *machine learning* untuk prediksi kesuksesan usaha di lokasi tertentu berdasarkan data historis, peningkatan performa algoritma untuk menangani data dalam jumlah lebih besar, serta penyempurnaan user interface berdasarkan hasil *usability testing* dan *feedback* pengguna.

Pemeliharaan preventif dilakukan sebagai tindakan pencegahan untuk menghindari masalah di masa mendatang. Kegiatan preventif meliputi optimasi database untuk menjaga performa query tetap cepat meskipun data bertambah banyak, pembersihan data redundan atau obsolete yang tidak lagi diperlukan, *update security patch* untuk *Framework* dan library yang digunakan guna menutup celah keamanan, monitoring performa sistem secara rutin untuk memantau *response time*, *error rate*, dan *server load*, serta audit keamanan berkala untuk mencegah *vulnerability* dan serangan siber.

Strategi *backup* dan *recovery* diterapkan untuk melindungi data sistem. *Backup database* dilakukan secara otomatis setiap hari menggunakan metode incremental backup, dilengkapi dengan full backup setiap minggu. Semua backup disimpan di cloud storage dengan redundancy untuk memastikan keamanan data. Prosedur *recovery* didokumentasikan dengan detail untuk berbagai skenario kegagalan seperti kerusakan hardware, human error, atau serangan malware. Prosedur *recovery* juga diuji secara berkala untuk memastikan dapat berfungsi dengan baik ketika dibutuhkan.

Monitoring sistem dilakukan melalui implementasi *tools monitoring* untuk memantau performa server dan API secara

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

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

2. Penggunaan tanpa izin untuk kepentingan komersial atau pelanggaran hak cipta dapat dikenakan sanksi sesuai dengan UU Hak Cipta di Indonesia.

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.



real-time. Dashboard monitoring disediakan untuk melihat berbagai metrics seperti jumlah request per menit, response time rata-rata, dan *error rate*. Alert otomatis dikonfigurasi untuk mengirim notifikasi via email atau SMS jika terjadi anomali seperti *server down* atau *error rate* yang meningkat drastis. Log analysis dilakukan secara berkala untuk mengidentifikasi pattern masalah dan mengambil tindakan preventif sebelum masalah menjadi serius.

Output dari tahapan maintenance adalah sistem pendukung keputusan yang tetap stabil dan reliable dalam operasi jangka panjang, aman dari ancaman security dan data breach, relevan dengan kebutuhan bisnis yang dinamis, akurat dalam memberikan rekomendasi pemilihan lokasi, serta *up-to-date* dengan teknologi dan best practices terkini. Dengan pemeliharaan yang terencana dan konsisten, sistem dapat memiliki *lifecycle* yang panjang dan memberikan return on investment yang optimal bagi stakeholder, serta terus memberikan manfaat dalam meminimalkan risiko kegagalan usaha akibat kesalahan pemilihan lokasi di Kecamatan Tembilahan, Kabupaten Indragiri Hilir.

- Aman dari ancaman security dan data breach
- Relevan dengan kebutuhan bisnis yang dinamis
- Akurat dalam memberikan rekomendasi pemilihan lokasi
- *Up-to-date* dengan teknologi dan best practices terkini

Dengan pemeliharaan yang terencana dan konsisten, sistem dapat memiliki *lifecycle* yang panjang dan memberikan return on investment yang optimal bagi stakeholder, serta terus memberikan manfaat dalam meminimalkan risiko kegagalan usaha akibat kesalahan pemilihan lokasi di Kecamatan Tembilahan, Kabupaten Indragiri Hilir.

