

BAB I

PENDAHULUAAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi yang pesat telah mendorong transformasi digital di berbagai sektor industri, termasuk dalam industri telekomunikasi. Di era digital, pelanggan semakin mengharapkan pengalaman layanan yang personal dan cepat sesuai dengan kebutuhan mereka. Salah satu cara untuk memenuhi ekspektasi tersebut adalah dengan memanfaatkan sistem rekomendasi yang dapat memberikan saran produk secara otomatis berdasarkan preferensi pengguna. Sistem rekomendasi banyak diaplikasikan dalam berbagai platform digital, seperti *e-commerce*, *e-learning*, dan layanan konten online, untuk membantu pengguna menemukan produk atau layanan yang relevan di tengah banyaknya pilihan yang tersedia (Suryono dkk., 2026).

Sistem rekomendasi berbasis web merupakan pendekatan yang populer karena dapat diakses secara fleksibel melalui internet dan terintegrasi langsung dengan aplikasi yang digunakan oleh pengguna. Sebagai contoh, penelitian yang dilakukan oleh Suryono dkk (2026). menunjukkan implementasi sistem rekomendasi produk berbasis web dengan pendekatan *content-based filtering* untuk platform penjualan mebel, yang berhasil memberikan rekomendasi sesuai preferensi pengguna pada antarmuka web (Suryono dkk., 2026). Selain itu, Fabiano dkk (2025) juga mengembangkan sistem rekomendasi *e-book* berbasis web untuk platform digital, memperlihatkan bahwa integrasi sistem rekomendasi ke dalam web meningkatkan pengalaman pengguna dalam menemukan konten yang sesuai (Fabiano dkk., 2025).

Dalam konteks yang lebih luas, *machine learning* sering digunakan sebagai mesin di balik sistem rekomendasi, karena kemampuannya untuk menganalisis data pengguna dan memberikan rekomendasi secara adaptif berdasarkan pola perilaku. Penelitian-penelitian terbaru menunjukkan bahwa penggabungan pendekatan *machine learning* dengan sistem rekomendasi mampu meningkatkan relevansi rekomendasi yang diberikan kepada pengguna (Wijaya dan Yuniarto, 2024). Meskipun demikian, sebagian besar studi terdahulu cenderung fokus pada aspek

metodologi algoritma atau performa model, dan belum banyak menekankan pada aspek merancang teknis aplikasi berbasis web secara keseluruhan.

Pada penelitian ini digunakan algoritma *Random forest* untuk membangun model sistem rekomendasi. *Random forest* merupakan metode *ensemble learning* yang menggabungkan beberapa *decision tree* untuk menghasilkan prediksi yang lebih akurat dan stabil (Agustina dkk., 2024). Algoritma ini dipilih karena mampu menangani data dengan banyak fitur serta dapat mengidentifikasi pola perilaku pelanggan dari data historis penggunaan. Dengan demikian, model yang dihasilkan diharapkan dapat memberikan rekomendasi paket telekomunikasi yang lebih sesuai dengan kebutuhan pengguna.

Untuk itu, penelitian ini bertujuan mengembangkan sistem rekomendasi produk telekomunikasi berbasis web dengan integrasi *Machine Learning* menggunakan *Random forest* sebagai modul rekomendasi. Fokus utama penelitian bukan hanya pada kemampuan model *Random forest* itu sendiri, tetapi pada perancangan sistem web, integrasi model ke dalam *backend* aplikasi, serta performa aplikasi dalam memberikan rekomendasi secara fungsional kepada pengguna.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan masalah sebagai berikut:

“Bagaimana merancang sistem rekomendasi produk telekomunikasi berbasis web dengan integrasi *machine learning* menggunakan *Random forest* untuk menghasilkan rekomendasi paket telekomunikasi yang sesuai dengan kebutuhan dan perilaku pelanggan?”.

1.3 Batasan Masalah

Berdasarkan rumusan masalah, batasan masalah diperlukan sebagai acuan untuk memfokuskan pembahasan pada tujuan penelitian dan mencegah pembahasan yang menyimpang. Adapun batasan masalah dalam penelitian ini meliputi:

1. Lingkup Sistem

- a. Sistem difokuskan pada perancangan sistem rekomendasi produk telekomunikasi berbasis web dengan integrasi *machine learning* menggunakan *Random forest* sebagai modul rekomendasi
- b. Sistem hanya berfungsi sebagai alat bantu menampilkan hasil rekomendasi paket telekomunikasi dan tidak mencakup proses transaksi pembelian atau pembayaran paket, maupun integrasi dengan sistem billing perusahaan telekomunikasi.
- c. Data yang digunakan dalam penelitian ini berasal dari *Dataset* Kaggle dan data *dummy* yang akan dibuat sesuai kebutuhan pengembangan, dengan cakupan data perilaku pelanggan telekomunikasi untuk keperluan pelatihan, pengujian, dan evaluasi sistem rekomendasi.

2. Fungsi Utama Sistem

- a. Sistem mampu mengolah data historis pelanggan untuk menghasilkan rekomendasi paket telekomunikasi.
- b. Sistem menampilkan hasil rekomendasi dalam bentuk antarmuka aplikasi web yang dapat diakses oleh pengguna.
- c. Sistem menyediakan fitur pengelolaan data paket dan memantau hasil rekomendasi pengguna melalui halaman admin.

3. Penggunaan Sistem

- a. Admin memiliki hak akses untuk mengelola data paket serta memantau hasil rekomendasi yang dihasilkan oleh sistem.
- b. Pengguna sistem hanya dapat melihat hasil rekomendasi paket yang dihasilkan oleh sistem.

4. Teknologi yang digunakan

- a. Bahasa pemrograman yang digunakan adalah Python untuk pengembangan model *Machine Learning* dengan *Random forest* dan *backend*.
- b. *Library* dan *framework Front-End* yang digunakan adalah React.js.
- c. *Framework Back-End* yang digunakan adalah FastAPI.
- d. *Database* yang digunakan adalah PostgreSQL.

- e. Model *Machine Learning* dengan algoritma *Random forest* digunakan sebagai modul rekomendasi tanpa dilakukan perbandingan dengan algoritma lain atau optimasi parameter secara mendalam.

1.4 Tujuan dan Manfaat

Tujuan dari merancang sistem ini adalah untuk membangun dan menerapkan sebuah sistem rekomendasi produk telekomunikasi berbasis web dengan integrasi *machine learning* menggunakan *Random forest* yang dapat membantu proses penawaran paket agar lebih tepat sasaran. Sistem ini dirancang untuk menganalisis data historis penggunaan pelanggan dan menghasilkan rekomendasi paket yang sesuai dengan kebutuhan serta pola penggunaan pelanggan melalui sebuah *dashboard* berbasis web. Selain itu, sistem ini bertujuan untuk merancang arsitektur aplikasi yang terstruktur dan terintegrasi antara *frontend*, *backend*, *database*, serta modul *machine learning* sehingga sistem dapat berjalan secara efektif dan fungsional.

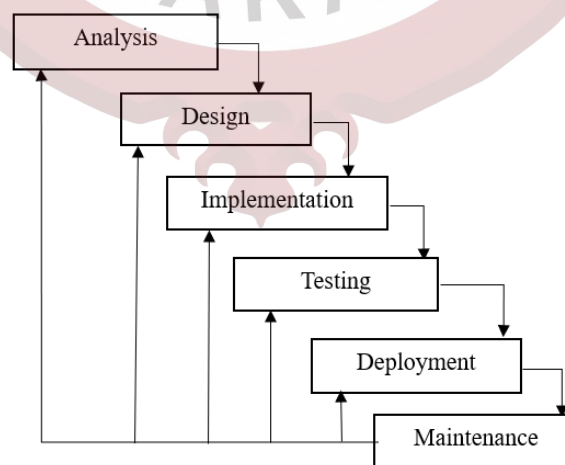
Manfaat yang diharapkan dari perancangan sistem ini adalah:

1. Bagi Mahasiswa (Penulis)
 - a. Menambah pengalaman dan pengetahuan dalam merancang dan mengembangkan sistem rekomendasi paket telekomunikasi menggunakan metode *Random forest*.
 - b. Menjadi sarana penerapan teknologi yang telah dipelajari selama perkuliahan seperti React.js, FastAPI, serta integrasi model *Machine Learning* dengan *Random forest* ke dalam sistem aplikasi.
 - c. Memberikan pengalaman dalam membangun aplikasi mulai dari perancangan sistem, pengembangan *backend* dan *frontend*, hingga implementasi model rekomendasi.
2. Bagi Perusahaan Telekomunikasi / Industri Telekomunikasi
 - a. Membantu pengguna dalam memperoleh rekomendasi paket yang sesuai dengan kebutuhan penggunaan.
 - b. Mempermudah perusahaan dalam menawarkan paket yang lebih relevan berdasarkan data pelanggan.

- c. Meningkatkan kualitas layanan melalui rekomendasi paket yang lebih tepat sasaran.
3. Bagi Universitas
- a. Menjadi salah satu hasil penelitian mahasiswa di bidang sistem rekomendasi berbasis *Machine Learning* dengan menggunakan *Random forest*.
 - b. Mendukung pengembangan penelitian yang berkaitan dengan teknologi informasi dan analisis data.
 - c. Menambah referensi penelitian yang dapat digunakan oleh mahasiswa pada topik serupa.

1.5 Metodologi Penelitian

Metode yang digunakan dalam implementasi sistem adalah metode *Software Development Life Cycle (SDLC)* dengan model *Waterfall*. Model *Waterfall* merupakan salah satu model pengembangan perangkat lunak yang dilakukan secara terstruktur dan berurutan, di mana proses pengembangan diawali dengan tahap spesifikasi, pengembangan, validasi, dan evolusi. Model ini membagi proses pengembangan ke dalam beberapa fase yang tersusun secara linier, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, deployment, dan pemeliharaan (Nurseptaji dan Ramdhani, 2021). Model *Waterfall* yang digunakan dalam pengembangan sistem ditampilkan pada Gambar 1.1.



Gambar 1. 1 Metode *Waterfall*

1. Analisis Kebutuhan Sistem

Berdasarkan permasalahan yang telah diidentifikasi, kebutuhan sistem terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional meliputi pengelolaan data pelanggan dan data paket telekomunikasi, analisis data historis penggunaan pelanggan, penyajian rekomendasi paket telekomunikasi berbasis *Machine Learning* menggunakan *Random forest*, serta penyediaan *dashboard* untuk menampilkan hasil rekomendasi dan informasi pendukung lainnya. Sementara itu, kebutuhan non-fungsional mencakup performa sistem yang responsif, keamanan data melalui mekanisme autentikasi dan otorisasi, ketersediaan sistem yang stabil, kemudahan penggunaan antarmuka, serta kemampuan sistem untuk menangani peningkatan jumlah data dan pengguna.

2. Desain

Tahap desain dilakukan untuk menyusun arsitektur sistem yang meliputi perancangan *frontend*, *backend*, *database*, serta integrasi modul *Machine Learning* dengan *Random forest*. Perancangan antarmuka dilakukan dengan mempertimbangkan kemudahan akses terhadap fitur utama seperti tampilan rekomendasi dan visualisasi data pelanggan. Selain itu, perancangan sistem divisualisasikan menggunakan *Unified Modeling Language* (UML) seperti *use case diagram* untuk menggambarkan interaksi pengguna dengan sistem, *activity diagram* untuk memodelkan alur proses, *sequence diagram* untuk menunjukkan komunikasi antar komponen, serta *class diagram* untuk menggambarkan struktur data dan relasi antar entitas dalam sistem.

3. Implementation (Implementasi)

Tahap implementasi dilakukan dengan membangun sistem berbasis web sesuai dengan desain yang telah dirancang. Pengembangan *frontend* dilakukan untuk menampilkan *dashboard* dan hasil rekomendasi secara interaktif, sedangkan *backend* dikembangkan untuk mengelola proses bisnis dan menyediakan layanan API. *Database* digunakan untuk menyimpan data pelanggan, data produk, serta hasil rekomendasi. Model *Machine Learning*

dengan algoritma *Random forest* dikembangkan untuk menganalisis pola perilaku pelanggan berdasarkan data historis dan menghasilkan rekomendasi produk yang relevan. Model tersebut kemudian diintegrasikan ke dalam sistem web melalui layanan backend sehingga proses rekomendasi dapat dilakukan secara otomatis.

4. *Testing* (Pengujian)

Tahap pengujian (*testing*) dilakukan untuk memastikan bahwa seluruh fitur sistem berjalan sesuai dengan kebutuhan yang telah ditentukan. *Pengujian fungsional dilakukan menggunakan metode black-box testing* untuk memverifikasi setiap fitur utama, seperti proses *login*, pengolahan data, dan penampilan rekomendasi. Selain itu, dilakukan pengujian integrasi untuk memastikan komunikasi antara *frontend*, *backend*, *database*, dan model *Random forest* berjalan dengan baik. Pengujian performa juga dilakukan untuk menilai waktu respons sistem dalam menghasilkan rekomendasi.

5. *Deployment*

Tahap *deployment* dilakukan dengan mengimplementasikan sistem ke lingkungan produksi agar dapat diakses oleh pengguna melalui web *browser*. Pada tahap ini dilakukan konfigurasi *server*, koneksi *database*, serta integrasi model *Machine Learning* agar sistem dapat berjalan secara optimal dalam lingkungan nyata.

6. *Maintenance* (Pemeliharaan)

Tahap terakhir yaitu *maintenance* (pemeliharaan) dilakukan setelah sistem digunakan. Pada tahap ini dilakukan perbaikan apabila ditemukan kesalahan, peningkatan performa sistem, serta pembaruan model *Machine Learning* apabila terdapat perubahan pola data pelanggan. *Monitoring* dilakukan secara berkala untuk memastikan sistem tetap stabil dan mampu memberikan rekomendasi produk yang relevan sesuai kebutuhan pengguna.

1.6 Sistematika Penulisan

Untuk memudahkan pemahaman dan penyusunan laporan skripsi ini, maka sistematika penulisan disusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat, metodologi implementasi yang digunakan, serta sistematika penulisan skripsi. Bab ini menjadi dasar dan gambaran umum mengenai penelitian yang dilakukan.

BAB II LANDASAN TEORI

Bab ini menguraikan tinjauan pustaka dan teori-teori pendukung yang berkaitan dengan sistem rekomendasi, konsep dasar *Machine Learning*, *Random forest*, *web* dan *tools* yang diperlukan dalam pembangunannya, seperti: *Unified Modelling Language (UML)*, *website*, dan sebagainya. Bab ini juga menyajikan penelitian terdahulu yang relevan sebagai landasan dalam perancangan sistem..

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas analisis kebutuhan sistem serta perancangan sistem aplikasi rekomendasi produk telekomunikasi, yang meliputi perancangan arsitektur sistem, perancangan basis data, perancangan alur proses, serta pemodelan sistem menggunakan *Unified Modeling Language (UML)*.

BAB IV IMPLEMENTASI SISTEM DAN ANALISIS HASIL

Bab ini menjelaskan proses implementasi sistem rekomendasi produk telekomunikasi berbasis web berdasarkan perancangan yang telah dilakukan. Selain itu, bab ini juga membahas hasil pengujian sistem, evaluasi performa sistem dalam menghasilkan rekomendasi, serta analisis terhadap hasil implementasi yang diperoleh.

BAB V SIMPULAN DAN SARAN

Bab ini memuat simpulan yang diperoleh dari hasil implementasi sistem aplikasi rekomendasi produk telekomunikasi serta saran untuk pengembangan dan penyempurnaan sistem di masa mendatang.

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang berjudul “*Recommender Systems Applications: Data Sources, Features, and Challenges*” mengulas lanskap sistem rekomendasi secara komprehensif, termasuk sumber data, fitur, pendekatan, serta tantangan yang dihadapi dalam implementasinya pada berbagai *domain* aplikasi digital. Penelitian ini menunjukkan bahwa sistem rekomendasi dapat menyaring data besar menjadi informasi yang relevan dan personal untuk pengguna, sehingga meningkatkan efektivitas pengambilan keputusan dalam konteks layanan digital. Studi ini dapat diunduh secara bebas dan digunakan sebagai referensi teori sistem rekomendasi berbasis *machine learning* (Alfaifi, 2024).

Selanjutnya, penelitian berjudul “*Machine Learning Algorithms in the Development of Recommender Systems*” membahas peran algoritma *machine learning* seperti *K-Nearest Neighbor* (KNN), *Decision Tree*, dan *Random forest* dalam meningkatkan performa sistem rekomendasi. Penelitian ini menjelaskan bahwa proses *preprocessing* data, pemilihan fitur, dan evaluasi model menjadi faktor penting dalam menghasilkan rekomendasi yang akurat dan relevan (Guan, 2024).

Penelitian lain yang relevan adalah berjudul “*Impact of Recommendation Systems on AI-enabled Customer Experience: Mediator Role of Perceived Usefulness and Perceived Trust*” yang diterbitkan pada *Journal of Telecommunications and the Digital Economy* membahas bagaimana sistem rekomendasi berbasis *artificial intelligence* mampu meningkatkan pengalaman pelanggan melalui personalisasi layanan. Studi ini menunjukkan bahwa rekomendasi yang tepat dapat meningkatkan kepercayaan pengguna serta memperkuat loyalitas pelanggan terhadap layanan digital (Jlassi dkk., 2025).

Selain itu, penelitian “*An Improved Product Recommender System Using Collaborative Filtering and a Comparative Study of Machine Learning Algorithms*” membahas implementasi *collaborative filtering* pada sistem rekomendasi produk serta melakukan perbandingan performa beberapa algoritma

machine learning untuk menentukan model terbaik. Penelitian ini relevan karena menunjukkan pentingnya pemilihan algoritma yang tepat dalam meningkatkan akurasi rekomendasi berbasis perilaku pengguna (Amutha dan Surya., 2023).

Berdasarkan penelitian terdahulu di atas, mayoritas studi berfokus pada pengembangan algoritma sistem rekomendasi secara umum serta peningkatan performa model *machine learning* dalam berbagai *domain* digital. Namun, masih terbatas penelitian yang secara spesifik mengimplementasikan sistem rekomendasi produk dalam konteks layanan telekomunikasi yang terintegrasi langsung ke dalam aplikasi berbasis web. Sebagian penelitian hanya menitikberatkan pada aspek model tanpa integrasi sistem, sementara penelitian lain lebih berfokus pada pengalaman pengguna tanpa membahas perancangan teknis secara menyeluruh.

Selain itu, pemilihan algoritma *Random forest* pada penelitian ini didasarkan pada keunggulannya dibandingkan algoritma klasifikasi lain seperti *Decision Tree*, *K-Nearest Neighbor*, dan *Naïve Bayes*. *Random forest* merupakan metode *ensemble* yang menggabungkan banyak *decision tree* sehingga mampu mengurangi *overfitting* dan menghasilkan prediksi yang lebih stabil. Penelitian komparatif menunjukkan bahwa *Random forest* memiliki performa yang lebih unggul dibandingkan *Decision Tree* dalam aspek akurasi, presisi, dan *recall* karena proses *voting* dari banyak pohon keputusan mampu meningkatkan kemampuan generalisasi model terhadap data baru. Selain itu, *Random forest* juga lebih baik dalam menangani data dengan jumlah fitur yang besar serta mampu mempelajari hubungan *non-linear* antar variabel, sehingga cocok digunakan pada data perilaku pengguna yang kompleks (Nugraha dkk., 2026) (Septiawan Putra dkk., 2025). Dengan keunggulan tersebut, *Random forest* dinilai lebih sesuai untuk sistem rekomendasi yang memanfaatkan data historis pelanggan untuk menghasilkan rekomendasi paket secara lebih akurat dan konsisten.

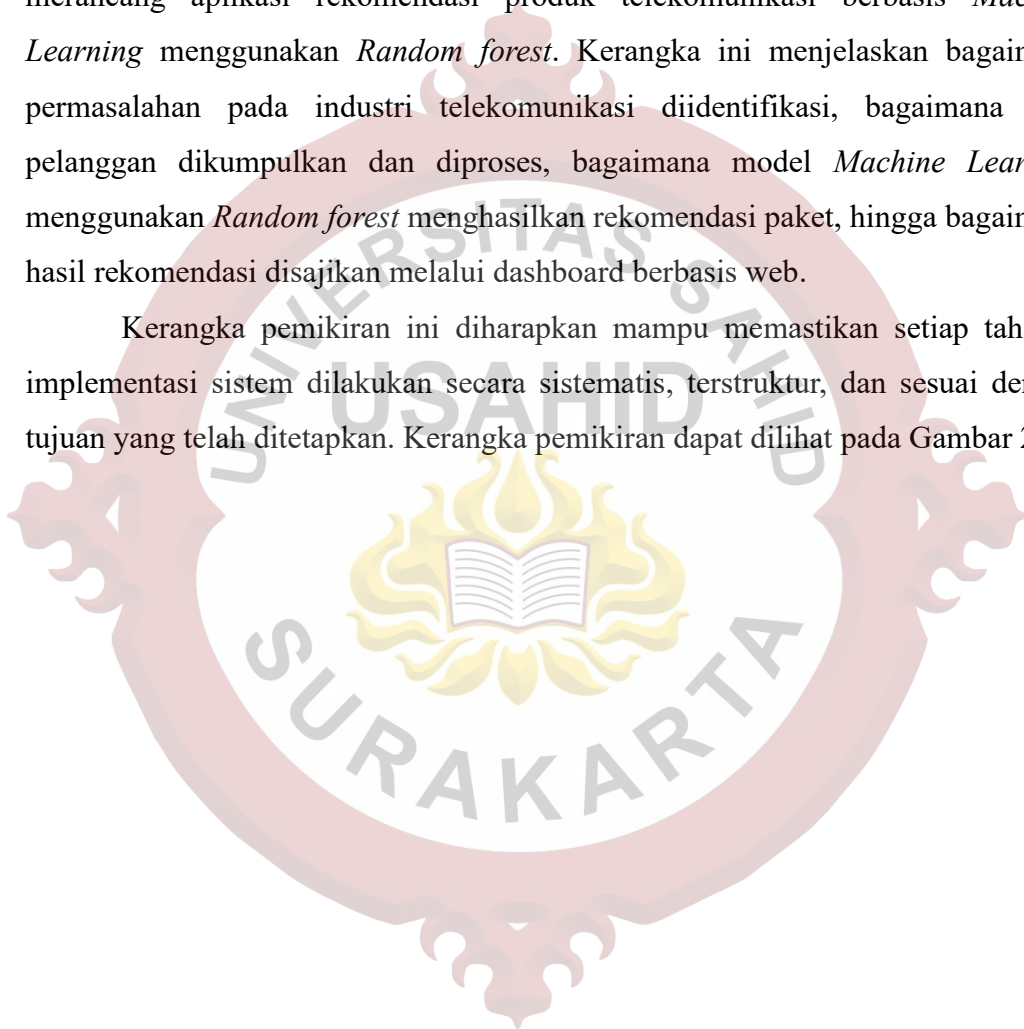
Oleh karena itu, penelitian ini memiliki kebaruan (*novelty*) berupa perancangan sistem rekomendasi produk telekomunikasi berbasis web dengan integrasi *Machine Learning* menggunakan *Random forest* yang tidak hanya membangun model prediksi berdasarkan data perilaku pelanggan, tetapi juga mengintegrasikan model tersebut ke dalam aplikasi web interaktif sehingga dapat

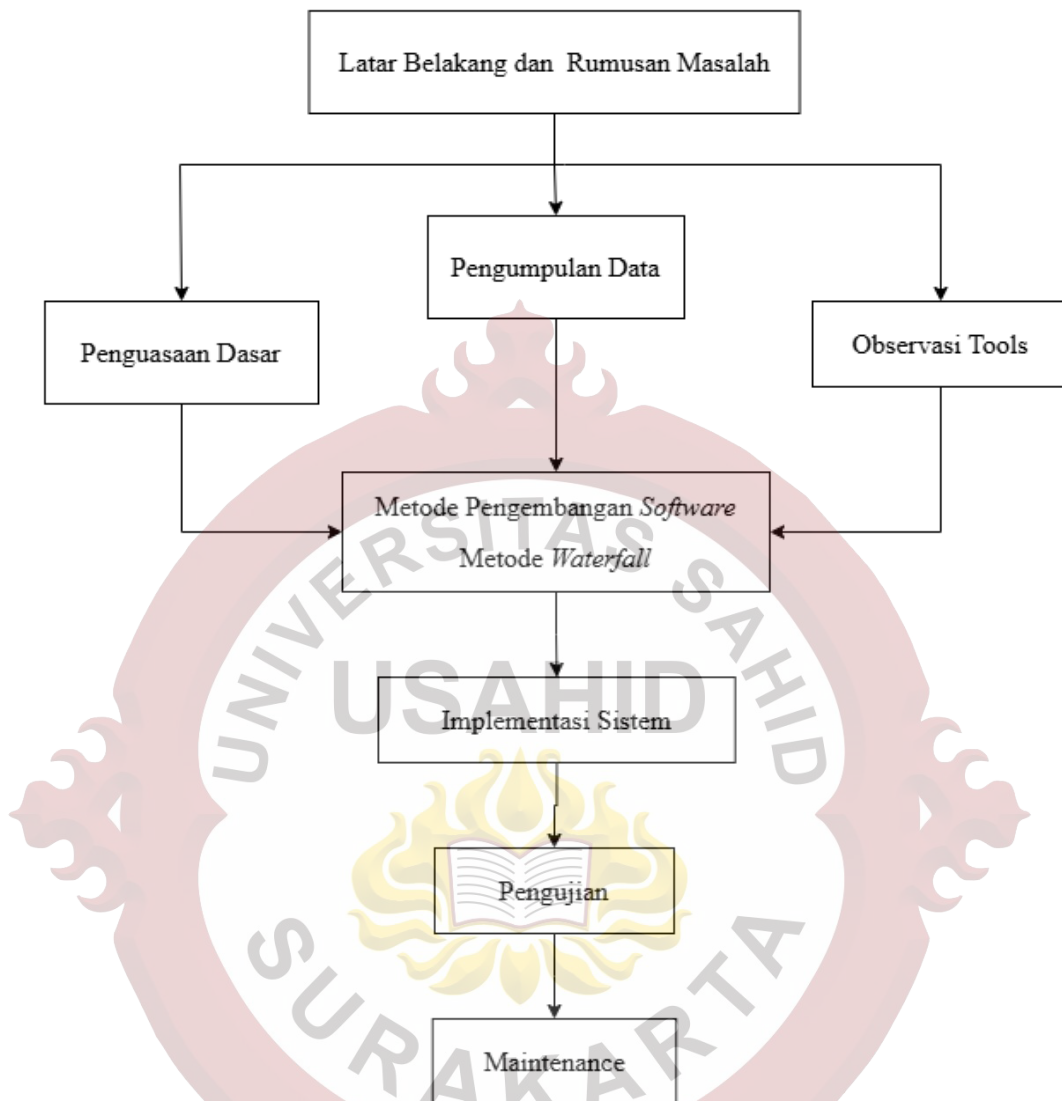
memberikan rekomendasi paket secara *real-time*. Perancangan ini diwujudkan dalam sistem sebagai solusi praktis untuk mendukung personalisasi layanan telekomunikasi berbasis data.

2.2 Kerangka Pemikiran

Kerangka pemikiran ini memberikan gambaran alur dan tahapan dalam merancang aplikasi rekomendasi produk telekomunikasi berbasis *Machine Learning* menggunakan *Random forest*. Kerangka ini menjelaskan bagaimana permasalahan pada industri telekomunikasi diidentifikasi, bagaimana data pelanggan dikumpulkan dan diproses, bagaimana model *Machine Learning* menggunakan *Random forest* menghasilkan rekomendasi paket, hingga bagaimana hasil rekomendasi disajikan melalui dashboard berbasis web.

Kerangka pemikiran ini diharapkan mampu memastikan setiap tahapan implementasi sistem dilakukan secara sistematis, terstruktur, dan sesuai dengan tujuan yang telah ditetapkan. Kerangka pemikiran dapat dilihat pada Gambar 2.1.





Gambar 2. 1 Kerangka Pemikiran

Kerangka pemikiran ini menggambarkan langkah-langkah sistematis dalam merancang sistem aplikasi rekomendasi produk telekomunikasi berbasis *Machine Learning* menggunakan *Random forest*. Berikut adalah penjelasan singkat setiap langkah:

1. Latar Belakang Masalah

Pada tahap ini, permasalahan utama yang dihadapi adalah banyaknya variasi paket layanan telekomunikasi yang ditawarkan operator, sehingga menyulitkan pelanggan dalam memilih paket yang sesuai dengan kebutuhan

dan pola penggunaan. Penawaran paket yang masih bersifat umum dan tidak personal berpotensi menurunkan tingkat kepuasan pelanggan.

2. Rumusan Masalah

Bagaimana merancang sistem rekomendasi produk telekomunikasi berbasis web dengan integrasi *Machine Learning* menggunakan *Random forest* untuk menghasilkan rekomendasi paket telekomunikasi yang sesuai dengan kebutuhan dan perilaku pelanggan?

3. Penguasaan Dasar

Tahap ini mencakup pemahaman dasar mengenai *Random forest*, sistem rekomendasi, bahasa pemrograman Python, *framework* FastAPI, *library frontend* React.js, basis data PostgreSQL.

4. Pengumpulan Data

Pengumpulan data dilakukan melalui studi pustaka dari Kaggle, jurnal dan referensi ilmiah yang relevan, serta penggunaan *Dataset* pelanggan telekomunikasi yang mencakup data penggunaan layanan dan data paket.

5. Observasi Tools

Observasi *tools* dilakukan terhadap *Visual Studio Code* sebagai lingkungan pengembangan utama dalam implementasi sistem aplikasi rekomendasi produk telekomunikasi. *Visual Studio Code* dipilih karena merupakan *code editor* yang ringan, fleksibel, serta mendukung berbagai bahasa pemrograman dan *framework* yang digunakan dalam pengembangan sistem, seperti Python, FastAPI, dan React.js.

6. Metode Pengembangan Software

Pada tahap ini, pengembangan sistem dilakukan menggunakan metode *Software Development Life Cycle (SDLC)* dengan model *Waterfall*, yang meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, *deployment*, dan pemeliharaan.

7. Implementasi Sistem

Sistem aplikasi rekomendasi produk telekomunikasi dikembangkan menggunakan *Front-End* berbasis React.js, *Back-End* menggunakan

FastAPI, *database* PostgreSQL, model *Machine Learning* dengan *Random forest* untuk sistem rekomendasi paket.

8. Pengujian

Setelah sistem berhasil diimplementasikan, dilakukan pengujian untuk memastikan seluruh fitur berjalan sesuai dengan kebutuhan. Pengujian meliputi pengujian fungsional (*blackbox*) serta pengujian algoritma *Random forest*.

9. Maintenance

Tahap pemeliharaan dilakukan setelah sistem diimplementasikan. Fokus tahap ini meliputi perbaikan *bug*, peningkatan performa sistem, pembaruan model *Machine Learning* menggunakan *Random forest*, serta *monitoring* berkelanjutan untuk memastikan sistem tetap stabil dan optimal.

2.3 Landasan Teori

2.3.1 Sistem Rekomendasi

Sistem rekomendasi adalah teknologi yang dirancang untuk membantu pengguna dalam pengambilan keputusan dengan menyajikan saran item yang relevan berdasarkan data historis, pola perilaku pengguna, dan atribut item yang tersedia. Tujuan utama sistem rekomendasi adalah mengurangi *information overload* dan memberikan rekomendasi yang lebih personal sehingga pengguna tidak perlu melakukan eksplorasi manual terhadap seluruh pilihan yang ada (Alfaifi, 2024).

Sistem rekomendasi telah banyak digunakan dalam berbagai domain layanan digital seperti *e-commerce*, media *streaming*, dan pendidikan *daring* untuk menyaring informasi besar menjadi rekomendasi yang bermakna dan relevan bagi pengguna akhir. Penerapan sistem rekomendasi di domain industri menunjukkan peningkatan pengalaman pengguna serta efektivitas dalam pengambilan keputusan berbasis preferensi dan pola data historis (Habibi dan Ramadhan, 2025).

Berbagai pendekatan umum dalam sistem rekomendasi mencakup *content-based filtering*, *collaborative filtering*, dan *hybrid approaches*. Dalam *content-based filtering*, rekomendasi dihasilkan berdasarkan kemiripan karakteristik item

dengan preferensi individu pengguna. Sementara itu, *collaborative filtering* memanfaatkan kesamaan pola perilaku antar pengguna untuk merekomendasikan item yang relevan. Pendekatan *hybrid* menggabungkan kedua metode tersebut untuk meningkatkan akurasi dan relevansi rekomendasi (Science dan Akkaya, 2025).

Di era sekarang perubahan perilaku pengguna yang cepat, teknik yang lebih maju seperti *deep learning* dan model *neural* juga mulai diintegrasikan dalam sistem rekomendasi untuk meningkatkan performa rekomendasi pada skala besar dan menangani tantangan seperti *data sparsity* dan *cold start*. Pendekatan *deep learning* memungkinkan ekstraksi fitur kompleks yang tidak dapat diungkap oleh teknik tradisional (Liu, 2024).

Secara keseluruhan, sistem rekomendasi berkontribusi dalam meningkatkan relevansi hasil yang disajikan kepada pengguna serta membantu organisasi dalam memaksimalkan interaksi pengguna terhadap layanan. Dalam konteks aplikasi rekomendasi produk telekomunikasi, sistem rekomendasi dirancang untuk memberikan rekomendasi paket telekomunikasi yang sesuai berdasarkan pola perilaku dan kebutuhan pelanggan untuk mendukung Keputusan penawaran layanan yang lebih personal dan efektif.

2.3.2 Website

Website adalah salah satu koleksi dokumen HTML milik pribadi atau perusahaan yang berisi informasi dan berada dalam Web Server dan dapat diartikan sebagai kumpulan dari halaman situs yang terangkum dalam sebuah *domain* atau *subdomain* dan berada didalam *World Wide Web* (WWW). *Website* merupakan kumpulan halaman yang saling terhubung yang dapat diakses melalui jaringan internet menggunakan protokol HTTP atau HTTPS. *Website* digunakan sebagai media penyampaian informasi, layanan, dan fungsi interaktif kepada pengguna melalui *browser* tanpa memerlukan pemasangan perangkat lunak tambahan pada perangkat pengguna. Struktur suatu *website* umumnya terdiri atas konten teks, gambar, multimedia, dan navigasi yang diatur sedemikian rupa untuk memudahkan akses informasi secara *real-time* (Miftahuljannah dan Suharso, 2023).

Secara fungsional, *website* berperan sebagai sebuah sistem informasi berbasis web yang menyimpan dan menyajikan data secara terstruktur sehingga dapat diakses dengan cepat dan efisien. Perkembangan teknologi internet mengakselerasi penggunaan *website* tidak hanya sebagai media penyampaian informasi statis tetapi juga sebagai aplikasi dinamis yang mampu mengolah data, menampilkan visualisasi, serta berinteraksi dengan sistem *backend* melalui API (Eka Indah Wahyuni dkk., 2025).

Dalam era digital saat ini, *website* telah menjadi komponen utama dalam berbagai sistem informasi karena kemampuannya untuk mengintegrasikan data, memperluas jangkauan layanan, serta mendukung pengalaman pengguna (*user experience*) yang lebih baik. *Website* dapat dikembangkan dengan berbagai teknologi, baik untuk keperluan profil organisasi, sistem informasi manajemen, maupun aplikasi layanan digital lainnya (Miftahuljannah dan Suharso, 2023).

2.3.3 React.js

React.js adalah *library* JavaScript sumber terbuka yang digunakan untuk membangun antarmuka pengguna (*user interface*) aplikasi web dinamis berbasis komponen (*component-based architecture*). React dikembangkan dan dipelihara oleh Meta (Facebook) serta komunitas *open source*, dan telah menjadi salah satu teknologi paling populer dalam pengembangan *frontend* modern karena mampu meningkatkan modularitas, efisiensi rendering, serta kemudahan pemeliharaan kode (Jonathan dan Supriyadi, 2023).

Salah satu fitur utama React adalah konsep *Single Page Application* (SPA), di mana navigasi antar halaman dilakukan tanpa memuat ulang (*reload*) seluruh halaman. Teknik ini berdampak pada pengalaman pengguna yang lebih responsif dan cepat karena hanya bagian tertentu dari antarmuka yang diubah sesuai kebutuhan interaksi pengguna. Pendekatan ini juga memudahkan pengembangan aplikasi modern yang interaktif seperti *dashboard* dan sistem *real-time* (Jonathan dan Supriyadi, 2023).

2.3.4 Fast API

FastAPI adalah *framework* web modern berbasis Python yang dirancang khusus untuk membangun RESTful API berkinerja tinggi dengan dukungan

asynchronous programming dan dokumentasi otomatis. FastAPI menggabungkan penggunaan Python *type hint* untuk melakukan validasi input secara otomatis, pengelolaan rute (*routing*), dan pemrosesan *request* secara efisien sehingga sangat cocok untuk aplikasi *backend* seperti layanan rekomendasi dan *microservices*. FastAPI dibangun di atas ASGI (*Asynchronous Server Gateway Interface*) yang memungkinkan penanganan permintaan secara *non-blocking* sehingga memberikan performa lebih baik pada beban tinggi dibandingkan beberapa kerangka kerja tradisional berbasis WSGI (Bednarz dan Miłosz, 2025).

Selain itu, FastAPI mendukung pembuatan dokumentasi API otomatis berbasis standar OpenAPI dan JSON *Schema*, mempermudah pengembang dalam mendesain, menguji, dan mengintegrasikan layanan API dengan *frontend* atau layanan lain dalam arsitektur sistem terdistribusi. Keunggulan ini membuat FastAPI menjadi *framework* favorit dalam pengembangan *microservices* modern dan sistem berbasis *cloud* yang memerlukan kecepatan, efisiensi, serta kemudahan pengembangan (Bednarz dan Miłosz, 2025).

Penelitian yang dilakukan oleh Bednarz dan Miłosz (2025) menunjukkan bahwa FastAPI secara konsisten memberikan performa yang unggul dibandingkan dengan *framework* Python lainnya seperti Django dan Flask dalam konteks pengembangan API web, baik dalam permintaan *GET* maupun *POST*, serta operasi *CRUD* terkait basis data. Hal ini menjadikan FastAPI pilihan yang tepat untuk aplikasi yang memerlukan *respons* cepat dan skalabilitas tinggi seperti pada *backend* sistem rekomendasi, di mana integrasi dengan model *Machine Learning* dan penyajian hasil RESTful API sangat diperlukan (Bednarz dan Miłosz, 2025).

2.3.5 Database PostgreSQL

Database merupakan komponen utama dalam sistem informasi yang berfungsi untuk menyimpan, mengelola, mengolah, dan mengambil data secara efisien. Sistem rekomendasi seperti aplikasi rekomendasi produk telekomunikasi membutuhkan penyimpanan data pelanggan, histori penggunaan, paket layanan, serta hasil rekomendasi secara cepat dan konsisten. Sistem manajemen basis data relasional *open-source* seperti PostgreSQL sangat cocok digunakan karena kemampuannya dalam menangani data yang kompleks, mendukung transaksi

ACID (*Atomicity, Consistency, Isolation, Durability*), serta memberikan performa tinggi pada operasi skala besar (Salunke dan Ouda, 2024).

Selain itu, PostgreSQL mendukung fitur lanjutan seperti *indexing*, replikasi, partisi data, serta tipe data JSON/JSONB yang memudahkan penyimpanan data semi-terstruktur tanpa mengorbankan konsistensi data relasional. Kemampuan ini membuat PostgreSQL tidak hanya unggul sebagai basis data relasional tradisional tetapi juga fleksibel untuk kebutuhan aplikasi modern seperti sistem rekomendasi berbasis *Machine Learning*. Integrasi PostgreSQL dengan *backend* misalnya FastAPI memungkinkan pengelolaan data dengan cepat melalui RESTful API serta mendukung skalabilitas sistem secara keseluruhan (Salunke dan Ouda, 2024).

2.3.6 *Machine Learning*

Machine Learning adalah cabang dari kecerdasan buatan yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer belajar dari data untuk melakukan prediksi atau pengambilan keputusan tanpa diprogram secara eksplisit setiap langkahnya. Dalam konteks sistem rekomendasi seperti aplikasi rekomendasi produk telekomunikasi, *Machine Learning* digunakan untuk menganalisis pola perilaku pelanggan dan memprediksi paket telekomunikasi mana yang paling sesuai untuk ditawarkan kepada masing-masing pengguna berdasarkan data historis mereka (Habibi dan Ramadhan, 2025).

Pendekatan *Machine Learning* dalam sistem rekomendasi memungkinkan model untuk mengenali hubungan kompleks antara pengguna dan item tanpa aturan tetap yang dibuat secara manual. Berbagai teknik seperti *collaborative filtering*, *content-based filtering*, serta kombinasi keduanya (*hybrid approach*) termasuk dalam strategi umum yang didukung oleh algoritma *Machine Learning* untuk meningkatkan akurasi rekomendasi (Habibi dan Ramadhan, 2025).

Selanjutnya, *Machine Learning* dapat juga menggunakan teknik lanjutan seperti pembelajaran dalam (*deep learning*) untuk mengolah representasi fitur yang lebih kompleks dan memberikan rekomendasi yang lebih adaptif terhadap perubahan pola data pelanggan (Liu, 2024). Studi lain menunjukkan bahwa tren terkini dalam penerapan *Machine Learning* pada sistem rekomendasi mengarah pada penggunaan gabungan berbagai metode untuk mengatasi tantangan seperti

cold start, *data sparsity*, dan adaptasi *real-time* terhadap preferensi pengguna, sehingga menghasilkan rekomendasi yang lebih responsif dan personal (Habibi dan Ramadhan, 2025).

Dalam perancangan aplikasi rekomendasi produk telekomunikasi, *Machine Learning* berperan dalam proses pembelajaran pola penggunaan layanan pelanggan, seperti konsumsi data atau durasi panggilan, sehingga model dapat memberikan rekomendasi paket yang relevan. Dengan pendekatan ini, sistem rekomendasi mampu memperhitungkan variasi besar dalam perilaku pengguna secara otomatis, mendukung proses pengambilan keputusan yang lebih efektif bagi operator maupun pelanggan.

2.3.7 Random forest

Random forest merupakan salah satu algoritma *Machine Learning* yang termasuk dalam metode *ensemble learning* dengan membangun sejumlah *decision tree* dan menggabungkan hasil prediksi dari setiap pohon untuk menghasilkan keputusan yang lebih akurat. Proses pembentukan model dilakukan menggunakan teknik *bootstrap sampling* serta pemilihan fitur secara acak, sehingga mampu mengurangi *overfitting* dan meningkatkan kemampuan generalisasi model terhadap data baru. Pendekatan ini membuat *Random forest* cukup efektif digunakan pada data dengan jumlah fitur yang besar serta hubungan variabel yang kompleks (Agustina dkk., 2024).

Dalam penerapannya, *Random forest* banyak digunakan untuk klasifikasi dan sistem rekomendasi karena mampu mempelajari pola dari data historis pengguna. Penelitian sebelumnya menunjukkan bahwa *Random forest* dapat menghasilkan performa yang baik dengan nilai akurasi tinggi pada berbagai kasus, seperti deteksi anomali jaringan dan analisis data pengguna. Hal ini menunjukkan bahwa algoritma *Random forest* mampu mengidentifikasi hubungan antar fitur secara lebih stabil dibandingkan metode Tunggal (Agustina dkk., 2024).

Dalam penelitian aplikasi rekomendasi produk telekomunikasi, algoritma *Random forest* digunakan untuk mempelajari pola perilaku pelanggan berdasarkan data historis penggunaan layanan telekomunikasi, seperti pembelian paket, jenis paket yang digunakan, serta intensitas penggunaan layanan. Model yang dihasilkan

kemudian digunakan untuk memprediksi paket yang paling sesuai bagi masing-masing pelanggan. Dengan pendekatan ini, sistem aplikasi rekomendasi produk telekomunikasi diharapkan mampu memberikan rekomendasi paket telekomunikasi yang lebih relevan dan personal sesuai kebutuhan pengguna.

2.3.8 *Telco Product Recommender System*

Telco Product Recommender System adalah jenis khusus dari sistem rekomendasi yang dikembangkan untuk industri telekomunikasi dengan tujuan memberikan rekomendasi layanan atau paket telekomunikasi yang paling sesuai dengan kebutuhan individual pelanggan berdasarkan pola perilaku mereka. Dalam industri telekomunikasi, pelanggan umumnya memiliki preferensi dan pola penggunaan layanan yang berbeda-beda, seperti konsumsi data, durasi panggilan, penggunaan SMS, dan layanan tambahan lainnya. Karena itu, sistem rekomendasi bertindak sebagai alat bantu dalam menyaring berbagai pilihan paket layanan yang tersedia dan menghasilkan paket yang paling relevan bagi setiap pelanggan (Sulistyowati dkk., 2024).

Telco product recommender system dapat dibangun menggunakan beragam pendekatan, termasuk *content-based filtering*, *collaborative filtering*, *knowledge-based recommendation*, atau kombinasi keduanya (*hybrid*). Pendekatan *knowledge-based* menggunakan aturan eksplisit berdasarkan karakteristik pelanggan dan kriteria paket, sehingga cocok dalam konteks di mana data perilaku pengguna mungkin belum tersedia dalam jumlah besar atau ketika preferensi *domain* sangat spesifik (misalnya, jenis layanan atau batasan harga (Lestari dkk., 2025)).

Penelitian penerapan *recommender system* pada pemilihan paket internet menunjukkan bahwa penggunaan pendekatan berbasis pengetahuan (*knowledge-based*) dapat membantu pengguna dalam memilih paket layanan yang sesuai dengan kebutuhan mereka secara efektif melalui penerapan aturan rekomendasi yang disesuaikan. Dalam studi tersebut, sistem rekomendasi dikembangkan sebagai aplikasi berbasis web yang mampu menerjemahkan preferensi pengguna ke rekomendasi paket tertentu sehingga memudahkan proses pengambilan Keputusan (Lestari dkk., 2025).

Dalam konteks aplikasi rekomendasi produk telekomunikasi, sistem rekomendasi telekomunikasi dirancang untuk menganalisis data pelanggan, seperti pola penggunaan layanan dan karakteristik perilaku, kemudian menghasilkan rekomendasi paket layanan yang relevan melalui *dashboard* web. Sistem ini tidak melakukan transaksi pembelian, tetapi memberikan rekomendasi paket yang dapat digunakan operator atau pelanggan sebagai alat bantu pengambilan keputusan yang lebih personal dan tepat sasaran.

2.3.9 Preprocessing Data dan Evaluasi Model

Preprocessing data merupakan tahap awal dalam *machine learning* untuk menyiapkan data sebelum digunakan dalam pelatihan model. Tahap ini dapat mencakup pembersihan data, penanganan data kosong, pengubahan data kategorikal menjadi numerik, serta pemilihan fitur yang relevan agar data dapat diproses oleh algoritma. Kualitas data yang baik diperlukan karena proses *preprocessing* dan pemilihan fitur dapat memengaruhi performa model *machine learning* (Ahmed dkk., 2024).

Evaluasi model klasifikasi digunakan untuk mengetahui kemampuan model dalam memprediksi data uji. Evaluasi dapat dilakukan menggunakan *confusion matrix* dan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*. *Confusion matrix* merupakan tabel yang membandingkan hasil prediksi model dengan label aktual, sehingga dapat diketahui jumlah prediksi benar dan salah pada setiap kelas (Riehl dkk., 2023).

Accuracy menunjukkan perbandingan jumlah prediksi benar terhadap seluruh data yang diuji. Pada klasifikasi *multiclass*, *accuracy* dapat dipahami sebagai rasio prediksi benar terhadap seluruh data uji. Rumus *accuracy* yang digunakan pada penelitian ini ditunjukkan menggunakan Persamaan (1) (Opitz, 2024):

$$Accuracy = \frac{\sum_i m_{i,i}}{\sum_{(i,j)} m_{i,j}} \quad (1)$$

$$Accuracy = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Seluruh Data Uji}}$$

Keterangan:

- $m_{i,i}$ = elemen diagonal *confusion matrix* (prediksi benar)
- $\sum_i m_{i,i}$ = jumlah seluruh prediksi benar
- $\sum_{(i,j)} m_{i,j}$ = jumlah seluruh data pada *confusion matrix*

Rumus tersebut digunakan karena model pada penelitian ini melakukan klasifikasi ke dalam enam kelas rekomendasi paket telekomunikasi. Dengan demikian, prediksi dianggap benar apabila kelas rekomendasi yang diprediksi model sama dengan kelas aktual pada data uji (Opitz, 2024).

Precision digunakan untuk melihat ketepatan prediksi pada suatu kelas dengan Persamaan (2) :

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (2)$$

Keterangan:

- TP_i = data kelas i yang diprediksi benar.
- FP_i = data dari kelas lain yang salah diprediksi sebagai kelas i.

Recall digunakan untuk melihat kemampuan model menemukan data pada kelas tertentu dengan Persamaan (3):

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (3)$$

Keterangan:

- TP_i = data kelas i yang diprediksi benar.
- FN_i = data kelas i yang salah diprediksi sebagai kelas lain.

Sedangkan *F1-score* digunakan untuk melihat keseimbangan antara *precision* dan *recall* dengan Persamaan (4):

$$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (4)$$

Dalam penelitian ini, metrik tersebut digunakan untuk mengevaluasi model *Random forest* dalam mengklasifikasikan pelanggan ke dalam kelas rekomendasi paket telekomunikasi (Opitz, 2024).

2.3.10 Integrasi Model *Machine Learning* dengan *Backend*

Integrasi model *Machine Learning* dengan *backend* merupakan proses menghubungkan model yang telah dilatih dengan aplikasi agar model dapat digunakan untuk menghasilkan prediksi secara langsung. Pada aplikasi berbasis web, *backend* berperan sebagai pengelola proses bisnis dan penghubung antara antarmuka pengguna, *database*, serta proses pengolahan data. REST API dapat digunakan sebagai gaya arsitektur untuk membangun web API yang menghubungkan sistem melalui internet (Yasmin dkk., 2020).

Dalam sistem rekomendasi berbasis web, API digunakan sebagai jalur komunikasi antara *frontend* dan *backend*. *Frontend* mengirim *request* kepada *backend*, kemudian *backend* mengambil data yang dibutuhkan dari *database*, menyesuaikan format data dengan kebutuhan model, menjalankan proses prediksi, dan mengirimkan hasil rekomendasi kembali kepada *frontend*. Model *Machine Learning* juga dapat disediakan sebagai layanan yang diakses melalui REST *interface*, sehingga hasil prediksi dapat digunakan oleh aplikasi tanpa pengguna menjalankan model secara langsung (Neumann dkk., 2022).

Keberhasilan integrasi model dengan *backend* dapat dilihat dari kemampuan sistem dalam menerima *request* rekomendasi, memproses data pengguna menggunakan model *Machine Learning*, dan mengembalikan hasil rekomendasi kepada *frontend*. Integrasi ini penting agar aplikasi web dapat menyajikan layanan yang interaktif, terhubung dengan *database*, dan mampu menghasilkan output sesuai kebutuhan pengguna (Neumann dkk., 2022).

2.3.11 Desain Sistem

Desain sistem merupakan fase penting dalam pengembangan perangkat lunak yang menentukan bagaimana sebuah sistem akan memenuhi kebutuhan fungsional dan non-fungsional yang telah dianalisis sebelumnya. Pada tahap ini dilakukan perancangan komponen utama sistem termasuk antarmuka pengguna, struktur data, dan alur proses yang harus dijalankan oleh aplikasi. Desain sistem bertujuan untuk menghasilkan spesifikasi fungsional yang menjadi acuan dalam tahap perancangan sehingga setiap aspek sistem dapat dibangun secara terstruktur dan sesuai dengan kebutuhan pengguna (Putri dkk., 2024)

2.3.12 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah bahasa standar yang digunakan untuk memvisualisasikan, merancang, dan mendokumentasikan sistem berbasis objek. UML memberikan gambaran sistem secara grafis sehingga memudahkan pengembang dalam memahami struktur, alur, dan interaksi antar komponen sistem sebelum implementasi. UML membantu membuat blueprint sistem, termasuk proses bisnis, struktur data, relasi antar kelas, serta interaksi antar modul dalam suatu aplikasi (Wayahdi dan Ruziq, 2023).

Dalam penelitian ini, UML digunakan untuk memodelkan sistem rekomendasi agar alur data, interaksi pengguna, dan proses *backend* terlihat jelas. Terdapat beberapa diagram UML yang digunakan, yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

a. *Use Case Diagram*

Use Case Diagram menggambarkan interaksi antara pengguna dan sistem serta jenis-jenis aktivitas yang dapat dilakukan pengguna. *Use Case Diagram* memodelkan interaksi pengguna dengan fitur rekomendasi paket telekomunikasi, autentikasi akun, dan penelusuran produk. Diagram ini membantu tim pengembang memahami fungsionalitas utama sistem dari perspektif pengguna (Aurellia, 2025). Simbol *Use Case* bisa dilihat pada Tabel 2.1.

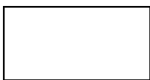
Tabel 2. 1 Simbol *Use Case*

Simbol	Keteramngan
 Actor	Aktor: Mewakili peran orang, sistem yang lain, atau alat Ketika berkomunikasi
	<i>Use Case</i> : Abstraksi dan interaksi antarasistem dan aktor.
	<i>Association</i> : Abstraksi dari penghubungantara aktor dengan <i>use case</i> .
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

b. *Activity Diagram*

Activity Diagram memodelkan alur proses atau aktivitas yang terjadi dalam sistem secara paralel maupun sekuensial. *Activity Diagram* menunjukkan alur pengguna mulai dari login, memilih paket telekomunikasi, hingga menerima rekomendasi. Diagram ini memastikan proses berjalan sesuai urutan logis dan memudahkan identifikasi tahapan yang bisa dioptimasi (Purwanti dkk., 2024). Simbol *Activity* bisa dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol *Activity Diagram*.

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan / <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / <i>Join</i>	Penggabungan dimana yang manalah dari satu aktivitas lalu digabungkan jadi satu.
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
	<i>Fork</i>	<i>Fork</i> merupakan aliran yang dapat bercabang menjadi dua atau lebih aliran paralel.

c. *Class Diagram*

Class Diagram memodelkan struktur data dan hubungan antar kelas dalam sistem. Diagram ini menampilkan atribut dan metode tiap kelas serta relasi antar kelas. *Class Diagram* menggambarkan kelas pengguna, paket telekomunikasi, rekomendasi, dan transaksi. *Class Diagram* menjadi panduan penting saat pengembangan *backend* dan penyusunan *database* (Wayahdi dan Ruziq, 2023). Simbol-Simbol bisa di lihat pada Tabel 2.3.

Tabel 2. 3 *Class Diagram*

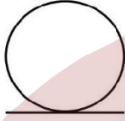
Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak(<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk(<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri(<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

d. *Sequence Diagram*

Sequence Diagram memodelkan interaksi antar objek sistem berdasarkan urutan waktu. Diagram ini menunjukkan aliran pesan yang dikirim dan diterima antar objek saat suatu skenario dijalankan. *Sequence*

Diagram digunakan untuk menjelaskan interaksi antara pengguna, *server*, *database*, dan model rekomendasi Top-K. Dengan diagram ini, tim pengembang dapat memastikan setiap fungsi sistem saling terkoordinasi dengan baik dan berjalan sesuai alur (Purwanti dkk., 2024). Simbol-Simbol bisa dilihat pada Tabel 2.4.

Tabel 2. 4 Simbol *Sequence Diagram*.

Simbol	Nama	Keterangan
	<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam Menyusun basis data.
	<i>Boundary Class</i>	Menangani komunikasi antar lingkungan system.
	<i>Control Class</i>	Bertanggung jawab terhadap kelas kelas terhadap objek yang berisi logika
	<i>Message call</i>	Gambaran sistem sebagai landasan dalam Menyusun basis data
	<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi.
	<i>Life Line</i>	Komponen yang digambarkan garis putus-putus dengan objek.

2.3.13 *Deployment Diagram*

Deployment Diagram adalah salah satu diagram dalam *Unified Modelling Language* (UML) yang menggambarkan struktur fisik sistem, yaitu bagaimana *software artifacts* atau komponen perangkat lunak ditempatkan pada *hardware nodes* serta hubungan antar *node* tersebut. Diagram ini menunjukkan konfigurasi eksekusi sistem serta hubungan komunikasi antar *server*, *client*, *middleware*, dan basis data yang digunakan saat sistem dijalankan di lingkungan produksi. *Deployment Diagram* menggambarkan aspek fisik dan lingkungan eksekusi dari

sistem perangkat lunak sehingga membantu perancang dan pengembang memastikan sistem berjalan sesuai arsitektur yang direncanakan (Andriati, 2023).

Deployment diagram bisa dilihat pada Tabel 2.5.

Tabel 2. 5 Simbol *Deployment Diagram*

Simbol	Keterangan
	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih.
	Biasanya mencakup pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri(<i>software</i>), jika didalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.
	Ketergantungan antar node, arah panah mengarah pada node yang dipakai.
	Relasi antar <i>node</i>

2.3.14 Perancangan

Perancangan sistem merupakan tahapan penting dalam rekayasa perangkat lunak yang bertujuan untuk mengubah hasil analisis kebutuhan menjadi model teknis yang siap diwujudkan dalam bentuk perangkat lunak. Pada tahap perancangan, spesifikasi rinci mengenai struktur data, alur proses, hubungan antar modul, serta komponen antarmuka ditetapkan sehingga menjadi *blueprint* bagi proses implementasi. Perancangan yang baik memungkinkan pengembang memahami kebutuhan dengan jelas sehingga mengurangi risiko kesalahan selama pembangunan aplikasi berbasis web (Annisa Tri Hidayati dkk., 2023).

2.3.15 User interface

User Interface (UI) adalah bagian dari sistem yang digunakan pengguna untuk berinteraksi dengan aplikasi atau perangkat lunak. UI berfungsi sebagai tampilan *visual* yang menjadi media penghubung antara sistem dengan pengguna, sehingga pengguna dapat memberikan perintah (*input*) dan menerima balasan dari

sistem (*output*) secara efektif. UI terdiri dari berbagai elemen *visual* seperti teks, ikon, tombol, menu navigasi, tata letak, serta warna yang disusun sedemikian rupa agar memudahkan pengguna dalam mengakses dan menjalankan fitur aplikasi (Firdani dkk., 2024).

Selain itu, beberapa penelitian telah menunjukkan bahwa pendekatan desain yang terstruktur, seperti penggunaan metode *Design Thinking* dan *User-Centered Design* (UCD), membantu desainer memahami kebutuhan pengguna dan menghasilkan antarmuka yang lebih tepat guna. Pendekatan desain ini umumnya melibatkan tahapan empati terhadap pengguna, perancangan prototipe, hingga pengujian tampilan antarmuka sebelum dilakukan implementasi terakhir (Firdani dkk., 2024).

2.3.16 Pengujian Sistem

Pengujian sistem merupakan tahap penting dalam proses pengembangan perangkat lunak untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan fungsional dan non-fungsional yang telah ditetapkan. Pengujian sistem mencakup evaluasi terhadap fungsionalitas, interaksi antar komponen, serta pengalaman pengguna, sehingga sistem layak untuk digunakan oleh pengguna akhir (Aurelius, 2026).

Dalam pengujian sistem, salah satu teknik yang umum digunakan adalah *Black Box Testing*, yang menitikberatkan pada pemeriksaan fungsi sistem dari luar tanpa memperhatikan struktur internal kode. Pengujian ini fokus pada input dan output sistem untuk memvalidasi apakah setiap fitur bekerja sesuai dengan spesifikasi yang diharapkan (Sahyudi dan Voutama, 2025).

Selain pengujian fungsional, penelitian ini juga melakukan pengujian terhadap model *Machine Learning* yang digunakan yaitu algoritma *Random forest*. Pengujian model dilakukan dengan membagi *Dataset* menjadi data pelatihan dan data pengujian untuk mengetahui kemampuan model dalam melakukan prediksi terhadap data baru. Selanjutnya performa model dievaluasi menggunakan *confusion matrix* dengan metrik *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengukur tingkat ketepatan model dalam memberikan rekomendasi. Pengujian ini bertujuan untuk memastikan bahwa algoritma *Random forest* mampu mempelajari

pola perilaku pengguna dan menghasilkan rekomendasi yang sesuai dengan kebutuhan pengguna (Ketut dkk., 2023).

Dengan adanya pengujian *Black Box Testing* dan evaluasi model *Random forest*, sistem diuji baik dari sisi fungsional aplikasi maupun dari sisi performa model rekomendasi sehingga sistem yang dikembangkan dapat berjalan dengan baik dan menghasilkan rekomendasi yang akurat.

2.3.17 Visual Studio Code

Visual Studio Code adalah *editor* kode sumber yang sederhana namun memiliki banyak fitur yang ditawarkan oleh *Microsoft*. Alat ini mendukung berbagai bahasa pemrograman dan memiliki fitur seperti penekanan *syntax*, *intellisense* (sugesti otomatis), alat *debugging*, dan ekosistem ekstensi yang luas. Kemampuannya untuk menyediakan pengalaman pengembangan yang terpadu dan efisien adalah keunggulan utama *VS Code* (Priyoga Listyo Ananda, 2024).

VSCode merupakan IDE ringan namun sangat kaya fitur, yang memungkinkan untuk menulis kode dalam berbagai bahasa pemrograman, termasuk yang digunakan dalam pengembangan web. *VSCode* dilengkapi dengan berbagai ekstensi yang dapat membantu dalam menulis, menguji, dan *mendebug* kode secara lebih efisien. Kemampuan *VSCode* dalam mendukung berbagai sistem operasi seperti *Windows*, *macOS*, dan *Linux* membuatnya sangat fleksibel dan dapat digunakan oleh berbagai kalangan, termasuk mahasiswa yang menggunakan perangkat dengan spesifikasi berbeda-beda (Priyoga Listyo Ananda, 2024).