

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Teori Dasar**

#### **2.1.1. Rancang Bangun**

Rancang bangun ini adalah langkah krusial dalam pengembangan sistem informasi yang mengubah hasil analisis menjadi desain detail siap implementasi Menurut Rahayu, Fajri, dan Hambali (2019), rancang bangun melibatkan beberapa langkah kunci yang saling terkait, yaitu analisis, perancangan, dan implementasi. Tahap pertama, analisis kebutuhan melibatkan identifikasi masalah dan kebutuhan pengguna melalui wawancara, observasi, dan dokumentasi. Data ini digunakan untuk memahami cara kerja sistem saat ini dan mengidentifikasi area yang perlu diperbaiki. Setelah itu, kebutuhan fungsional dan non-fungsional didokumentasikan secara rinci. Selanjutnya, dalam tahap perancangan sistem, dilakukan desain arsitektur dan desain detail, termasuk pembuatan diagram UML seperti diagram use case, diagram kelas, dan diagram aktivitas. Proses juga didefinisikan secara terperinci untuk menggambarkan cara setiap komponen sistem berfungsi.

#### **2.1.2. Sistem Informasi**

Sistem adalah kumpulan dan kelompok komponen, seperti prosedur atau perangkat lunak, yang saling terhubung untuk mencapai tujuan tertentu. Oleh karena itu, sebuah sistem yang menawarkan informasi disebut sistem informasi. yang membantu operasi dan pengambilan keputusan bisnis. Ginting, Fadlina, dan Karim pada tahun 2022. Sistem informasi dapat melakukan banyak hal, seperti menyimpan data dalam jumlah besar dan memberikan akses ke data tersebut, serta meningkatkan produktivitas dan efisiensi karyawan organisasi atau kelompok. Sistem informasi inventori suku cadang di

Tepepa Garage membantu mengelola stok dengan lebih baik, membantu operasi harian bengkel, dan memberi manajer data yang akurat untuk membuat keputusan. Sistem ini menjamin bahwa semua komponen bekerja sama untuk mencapai tujuan utama, yaitu meningkatkan efisiensi operasional dan

### **2.1.3. Pengertian Inventaris**

Menurut Kinaswara, T. A. (2019). Inventarisasi adalah suatu proses yang bertujuan untuk mencatat dan mengelola barang yang keluar dan masuk. Proses ini mencakup pencatatan detail dari setiap transaksi, seperti tanggal, jumlah, dan jenis barang yang terlibat. Proses ini juga memastikan bahwa sekumpulan barang disusun dengan teratur dan mematuhi peraturan yang telah ditentukan. Organisasi dapat memastikan bahwa barang-barang mereka dikelola dengan efisien, menghindari kekurangan atau kelebihan stok, dan mematuhi kebijakan dan prosedur yang berlaku dengan melakukan inventarisasi yang sistematis, yang mencakup penempatan barang di lokasi yang tepat, pengelompokan barang berdasarkan kategori yang relevan, dan pemeliharaan catatan yang akurat untuk memudahkan pelacakan dan pengelolaan barang.

Manajemen Inventaris adalah proses penting dalam operasi bisnis yang mengatur persediaan barang atau bahan baku agar efisien dan efektif. Berikut adalah penjelasan lebih detail mengenai aspek-aspek utama dalam manajemen inventaris (Solehudin, 2023, p. 66)

1. Perhitungan Level Inventaris Optimal:
  - Tujuan

Tujuan utama perhitungan ini untuk menentukan persediaan yang optimal atau ideal untuk dipertahankan. Hal ini dilakukan dengan mempertimbangkan beberapa faktor, termasuk Biaya Pengelolaan Persediaan terdiri dari biaya penyimpanan barang (holding cost), yang merupakan biaya

yang dikeluarkan untuk menyimpan barang dalam gudang. Biaya pemesanan (order cost), yang merupakan biaya yang muncul setiap kali seseorang memesan barang untuk persediaan. Biaya kekurangan persediaan (shortage cost), yang merupakan biaya yang muncul karena kekurangan stok yang dapat mengganggu operasional atau layanan pelanggan.

- Metode EOQ (Jumlah Pesanan Ekonomis): Ini adalah metode yang paling umum untuk menghitung jumlah pesanan yang ideal untuk mengurangi semua biaya persediaan, termasuk biaya pemesanan dan penyimpanan.

- Pengelolaan Stok Minimum dan Maksimum (Stok Keamanan): Ini adalah jumlah persediaan yang dipertahankan untuk mengantisipasi permintaan yang tidak pasti atau waktu pemesanan. Tujuan dari pengelolaan stok minimum ini adalah untuk menghindari kekurangan stok yang dapat mengganggu produksi atau layanan pelanggan.

Stok Maksimum (Maximum Stock):

Batas atas jumlah persediaan yang dapat dipertahankan dalam periode waktu tertentu. Menetapkan stok maksimum penting untuk mengontrol biaya penyimpanan, mengurangi risiko kepunahan barang, dan mengoptimalkan penggunaan ruang gudang.

3. Monitoring Perputaran Inventaris:

- Tujuan

Perputaran inventaris mengukur seberapa cepat persediaan berputar dalam suatu periode. Ini membantu dalam mengevaluasi efisiensi pengelolaan persediaan.

- Rasio Perputaran Inventaris

Rasio ini diperoleh dari perhitungan jumlah penjualan bersih

dibagi rata-rata persediaan yang ada selama periode yang sama. Rasio ini memberikan wawasan tentang efisiensi dalam mengelola persediaan, di mana perputaran inventaris yang lebih tinggi menunjukkan bahwa persediaan berputar lebih cepat, yang bisa mengurangi biaya penyimpanan dan meningkatkan likuiditas perusahaan.

#### 4. Teknik Pengendalian Inventaris:

- Metode Just In Time (JIT)

Sistem ini mengarah pada pengiriman barang saat diperlukan untuk produksi atau penjualan, sehingga mengurangi kebutuhan akan persediaan besar.

- Teknik ABC (Pengelompokan ABC)

Memisahkan barang berdasarkan nilai relatif mereka dalam persediaan untuk memprioritaskan pengelolaan dan kontrol yang lebih intensif terhadap item yang memiliki dampak finansial yang signifikan.

- Perusahaan dapat meningkatkan efisiensi operasional, pangkas biaya sekaligus meningkatkan dukungan pelanggan dengan menerapkan manajemen inventaris yang efektif.

#### **2.1.4. System Development Life Cycle (SDLC)**

Berdasarkan Kendall, K. E., & Kendall, J. E. (2010), SDLC (Systems Development Life Cycle) adalah rangkaian langkah yang diikuti oleh para profesional sistem informasi dalam proses pengembangan dan implementasi sistem informasi. SDLC mencakup tahap-tahap perencanaan, analisis, desain, implementasi, dan pemeliharaan. Berikut ini penjelasan lebih detail mengenai setiap tahap, dengan fokus pada pembuatan aplikasi inventori suku cadang untuk Tepepa Garage:

##### A. Perencanaan (Planning)

Pada tahap ini, tujuan utama adalah menentukan ruang lingkup proyek, mengidentifikasi kebutuhan bisnis, dan merencanakan sumber daya yang diperlukan. Para profesional sistem informasi bekerja sama dengan pemangku kepentingan untuk mengidentifikasi masalah yang ada dan peluang untuk perbaikan.

Berikut Merupakan aspek – aspek dalam tahap perencanaan (Planning):

1. Penentuan Ruang Lingkup Proyek: Menetapkan apa saja yang akan dicakup dalam aplikasi inventori, termasuk fitur utama seperti pengelolaan stok, pencatatan barang masuk dan keluar, serta pelaporan.
2. Identifikasi Kebutuhan Bisnis: Memahami kebutuhan Tepepa Garage dalam mengelola inventori suku cadang, seperti pengelolaan data suku cadang yang akurat dan efisien.
3. Perencanaan Sumber Daya: Menetapkan sumber daya yang diperlukan seperti tim pengembang, perangkat keras, perangkat lunak, dan anggaran.

Tahap analisis melibatkan pengumpulan dan pemahaman kebutuhan sistem dari pengguna. Para analis sistem melakukan studi kelayakan, analisis kebutuhan, dan mendokumentasikan persyaratan sistem. Mereka menggunakan teknik seperti wawancara, survei, dan observasi untuk mengumpulkan informasi yang diperlukan.

Berikut Merupakan aspek – aspek dalam tahap Analisis:

1. Studi Kelayakan: Menilai apakah proyek ini layak dilakukan dari segi teknis, ekonomi, dan operasional.
2. Analisis Kebutuhan: Mengumpulkan informasi detail tentang kebutuhan pengguna, yang mencakup fungsi-fungsi utama yang diinginkan dalam aplikasi.
3. Dokumentasi Persyaratan Sistem: Membuat dokumen spesifikasi kebutuhan yang menjelaskan secara rinci Apa yang seharusnya

dilakukan oleh system.

### C. Perancangan (Design)

Tahap desain, arsitektur Pengembangan sistem yang diusulkan didasarkan pada persyaratan yang dikumpulkan selama fase analisis. Desain tersebut mencakup elemen-elemen desain data, desain antarmuka pengguna, desain proses, dan spesifikasi teknis lainnya. Hasil dari tahap ini adalah cetak biru yang rinci untuk pengembangan sistem.

Berikut Merupakan aspek – aspek dalam tahap Perancangan (Design):

1. Desain Data melibatkan perancangan struktur basis data untuk menyimpan informasi suku cadang, transaksi, dan laporan yang diperlukan oleh sistem."
2. Desain Antarmuka Pengguna mencakup pembuatan prototipe atau mockup dari antarmuka sistem untuk memastikan bahwa pengguna, khususnya staf bengkel, dapat dengan mudah menggunakan sistem."
3. Dalam Desain Proses, alur kerja aplikasi dirancang untuk menentukan bagaimana data akan mengalir dari input hingga menghasilkan output dalam sistem."
4. Spesifikasi Teknis menjelaskan perangkat lunak dan perangkat keras yang diperlukan untuk menjalankan aplikasi dengan baik.

### D. Implementasi (Implementation)

Selama tahap implementasi, sistem dibangun menggunakan desain pra-dibangun. Ini termasuk menulis kode, mengintegrasikannya, menguji komponen individu, dan melakukan pengujian sistem yang komprehensif untuk menjamin bahwa setiap komponen beroperasi

sebagaimana dimaksud.

Berikut Merupakan aspek – aspek dalam tahap Implementasi:

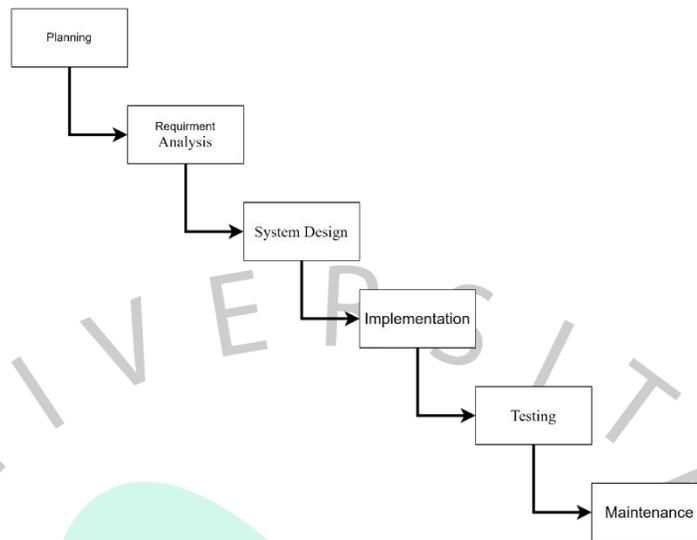
1. Penulisan Kode: Mengembangkan kode program berdasarkan desain yang telah dibuat, mencakup modul-modul untuk pengelolaan stok, transaksi barang masuk dan keluar, serta laporan.
2. Pengujian Unit: Melakukan pengujian pada setiap unit atau modul secara individual untuk memastikan bahwa setiap bagian bekerja dengan benar.
3. Integrasi: Menggabungkan semua modul yang telah dikembangkan menjadi satu sistem yang utuh dan memastikan integrasi berjalan lancar.
4. Pengujian Sistem: Melakukan pengujian menyeluruh pada sistem untuk menemukan dan memperbaiki bug atau kesalahan yang ada. Pengujian ini mencakup uji fungsi, uji performa, dan uji keamanan.
5. Pelatihan Pengguna: Memberikan pelatihan kepada staf Tepepa Garage tentang cara menggunakan aplikasi inventori suku cadang.
6. Implementasi Sistem: Memasang dan mengkonfigurasi aplikasi di lingkungan produksi, memastikan semua data yang diperlukan telah dimigrasi, dan sistem siap digunakan.

#### **2.1.5. Waterfall**

Menurut Sommerville, I. (2011). Model Waterfall adalah pendekatan tradisional dalam pengembangan perangkat lunak yang mengikuti proses sekuensial, dimana fase proyek tidak dapat tumpang tindih dan masing-masing harus diselesaikan secara keseluruhan sebelum pindah ke yang berikutnya. Model ini memiliki struktur yang

jas, dengan setiap langkah dalam pengembangan memiliki deliverable yang terdokumentasi dengan baik. Tepepa Garage menghadapi beberapa masalah dalam sistem inventaris suku cadang mereka, seperti pengelolaan stok yang masih manual dan rentan terhadap kesalahan, kurangnya integrasi data antara bagian-bagian bengkel, dan kesulitan dalam pemantauan barang masuk dan keluar. Untuk mengatasi masalah ini, perancangan dan pembangunan aplikasi inventaris suku cadang berbasis web menggunakan metode waterfall dipilih sebagai solusi. Proses ini dimulai dengan tahap perencanaan, di mana ruang lingkup proyek ditentukan dan kebutuhan bisnis diidentifikasi.

Tahap analisis kemudian dilakukan untuk mengumpulkan dan mendokumentasikan kebutuhan sistem dari pengguna. Pada tahap desain, arsitektur sistem dikembangkan berdasarkan persyaratan yang telah dikumpulkan. Implementasi mencakup penulisan, selain mengevaluasi unit dan sistem secara menyeluruh, Kode perangkat lunak yang dirancang sesuai dengan desain. Amati fase-fasenya implementasi, pemeliharaan dilakukan untuk mengatasi masalah yang muncul dan melakukan pembaruan sesuai kebutuhan. Dengan pendekatan ini, aplikasi inventaris suku cadang diharapkan dapat meningkatkan efisiensi dan akurasi pengelolaan stok di Tepepa Garage, menjamin integrasi data yang baik, serta mempermudah pemantauan barang masuk dan keluar.



Gambar 2. 1 Waterfall model

Perencanaan Persyaratan (*Requirement Planning*), Pemodelan Desain (*Design Workshop*), dan Penerapan Implementasi (*Implementation*).

#### 1. Perencanaan (*Planning*)

Tahap ini melibatkan identifikasi kebutuhan proyek, penentuan ruang lingkup, dan perencanaan sumber daya. Tujuannya adalah untuk menetapkan dasar yang solid untuk proyek dan mengidentifikasi tujuan utama yang ingin dicapai.

Berikut tahapan dari planning:

- Identifikasi kebutuhan proyek: Memahami tujuan bisnis dan kebutuhan spesifik yang harus dipenuhi oleh aplikasi inventori.
- Penentuan ruang lingkup proyek: Menentukan batasan proyek dan mengidentifikasi fitur utama yang akan dikembangkan.
- Perencanaan sumber daya: Mengalokasikan

sumber daya yang diperlukan, termasuk waktu, anggaran, dan tenaga kerja.

## 2. *Requirements Analysis*

Tahap ini, kebutuhan pengguna sistem dikumpulkan dan dianalisis. Analisis sistem bekerja sama dengan pemangku kepentingan untuk mendefinisikan spesifikasi dan persyaratan yang jelas dan mendetail, yang akan menjadi dasar bagi desain sistem.

Berikut adalah tahapan dari Requirements analysis:

- Pengumpulan kebutuhan: Menggunakan teknik seperti wawancara, survei, dan observasi untuk mengumpulkan informasi dari pengguna dan pemangku kepentingan.
- Analisis kebutuhan: Mengevaluasi informasi yang dikumpulkan untuk memastikan bahwa semua kebutuhan pengguna dan bisnis telah diidentifikasi dan dipahami.
- Dokumentasi kebutuhan: Membuat dokumen spesifikasi kebutuhan yang mendetail dan dapat dijadikan acuan untuk tahap desain.

## 3. *Desain Sistem (System Design)*

Desain sistem mencakup pembuatan arsitektur sistem, desain data, desain antarmuka, dan spesifikasi teknis lainnya. Hasil dari tahap ini adalah blueprint atau cetak biru yang rinci yang akan digunakan pada tahap pengembangan.

Berikut adalah tahapan dari Design:

- Desain arsitektur sistem: Menentukan struktur

keseluruhan sistem dan bagaimana komponen-komponen utama akan berinteraksi.

- Desain data: Merancang struktur data yang akan digunakan dalam sistem, termasuk basis data dan hubungan antar data.
- Desain antarmuka pengguna: Mengembangkan prototipe antarmuka pengguna membantu memverifikasi bahwa desain sistem memenuhi kebutuhan pengguna dan memastikan sistem tersebut mudah digunakan.
- Spesifikasi teknis: Menyusun spesifikasi teknis rinci untuk setiap komponen sistem.

#### 4. *Implementation*

Tahap ini, para pengembang menulis kode berdasarkan desain yang dibuat. Mereka memastikan bahwa kode yang dihasilkan sesuai dengan spesifikasi yang ditetapkan sambil menerapkan fitur yang telah ditunjukkan dalam persyaratan dan desain.

Berikut adalah tahapan dari Implementasi:

- Penulisan kode: Mengembangkan perangkat lunak berdasarkan spesifikasi desain.
- Pengujian unit: Menguji setiap komponen sistem secara individual untuk memastikan bahwa mereka berfungsi dengan baik.
- Integrasi: Menggabungkan semua komponen yang telah dikembangkan untuk menjadi satu sistem.

#### 5. *Testing*

Pada tahap pengujian, dilakukan pengujian unit,

pengujian Pengujian dan integrasi sistem untuk memastikan bahwa program beroperasi sudah sesuai dengan spesifikasi yang ditentukan dan bebas dari kesalahan. Setiap bagian dimasukkan melalui proses pengujian yang ketat untuk menjamin keandalan dan kualitas perangkat lunak.

Berikut adalah tahapan dari Testing:

- Pengujian unit: Menguji setiap bagian dari perangkat lunak secara terpisah.
- Pengujian integrasi: Memverifikasi bahwa komponen sistem berfungsi sebagai komponen yang kohesif.
- Pengujian sistem: Memeriksa struktur lengkap untuk memastikan bahwa semua persyaratan telah terpenuhi dan tidak ada bug.
- Pengujian penerimaan: Melibatkan pengguna untuk memastikan bahwa sistem memenuhi kebutuhan dan harapan mereka.

#### 6. Pemeliharaan (Maintenance)

Setelah perangkat lunak diimplementasikan, tahap pemeliharaan dimulai. Ini mencakup pemantauan sistem, perbaikan bug, pembaruan, dan penambahan fitur baru sesuai kebutuhan pengguna. Pemeliharaan bertujuan untuk memastikan bahwa sistem tetap berfungsi optimal sepanjang siklus hidupnya.

Berikut adalah tahapan dari Maintenance:

- Penulisan kode: Mengembangkan perangkat lunak berdasarkan spesifikasi desain.
- Pengujian unit: Menguji setiap komponen

sistem secara individual untuk memastikan bahwa mereka berfungsi dengan baik.

- Integrasi: Menggabungkan semua komponen yang telah dikembangkan menjadi satu sistem yang utuh.

#### **2.1.6. OOAD (*Object Oriented Analysis and Design*)**

Berdasarkan yang sudah di jelaskan Sutanto (2020) menyatakan bahwa adalah metode pengembangan sistem yang berbasis konsep objek yang menempatkan fokus utama pada objek daripada data atau proses. Pendekatan yang mencakup objek, kelas objek, dan pewarisan menandai OOAD. Objek adalah entitas yang menggabungkan sifat dan metode yang beroperasi berdasarkan sifat tersebut. Kelas objek adalah kelompok barang yang berperilaku dan memiliki struktur yang sama. Dalam pewarisan, setiap tipe entitas atau kelas objek mewarisi atribut dan metode dari kelas yang lebih tinggi dalam hierarki.

#### **2.1.7. UML (*Unified Modeling Language*)**

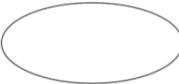
Menurut Sommerville, I. (2023) UML (*Unified Modeling Language*), berfungsi untuk mendefinisikan, mendeskripsikan secara visual, dan merinci elemen sistem perangkat lunak. UML menggunakan simbol grafis terstruktur untuk menunjukkan bagaimana komponen sistem berinteraksi satu sama lain dan beroperasi. Hal ini penting untuk membantu tim pengembangan dan pemangku kepentingan proyek berkomunikasi dengan baik dan memahami desain. Terdapat beberapa jenis diagram dalam membuat UML untuk melakukan rancang bangun sistem informasi. Berikut diagram-diagramnya:

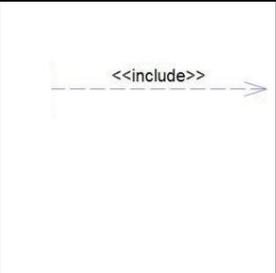
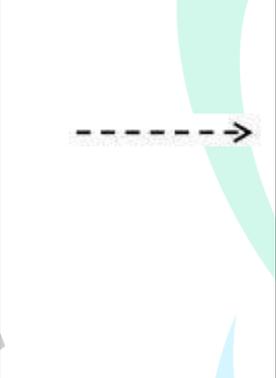
1. *Use Case*

Gunakan Use Case diagram membantu dalam pemodelan dan visualisasi persyaratan fungsional sistem. sudut pandang pengguna atau aktor, diagram adalah tahap awal dan analisis desain sistem perangkat keras. Ini menunjukkan bagaimana aktor bekerja sistem sesuai dengan kebutuhan mereka.

Selain itu, diagram ini menggambarkan bagaimana suatu sistem berinteraksi dengan aktor-aktornya di lingkungannya digambarkan dalam diagram use case. Jumlah use case dalam satu model tidak boleh lebih dari tiga sampai sembilan agar lebih mudah dipahami. Untuk mengurangi kompleksitas diagram, aktor harus ditempatkan lebih dekat dengan use case yang relevan. Hubungan dan interaksi antar aktor dan use case pada sistem yang dibangun digambarkan dalam diagram ini dengan notasi tertentu.

Tabel 2. 1Notasi Usecase Diagram

Simbo	Nama	Keterangan
1 	<i>Actor</i>	Subjek yang mewakili peraninteraksi pada sistem.
	<i>Use Case</i>	Gambaran utama antarasistem danaktor.

	<i>Include relationship</i>	Include digunakan untuk menunjukkan hubungan antar dua kasus penggunaan.
	<i>Extend relationship</i>	Extend digunakan ketika salah satu use case memiliki aksi tambahan yang dapat terjadi, bersifat opsional.
	<i>generalization</i>	Digunakan untuk spesialisasi aktor agar dapat berpartisipasi dalam Use Case.
	<i>association</i>	Penghubung antara <i>actor</i> dan <i>use case</i> .

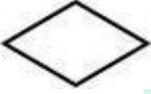
Sumber: (Dokumentasi Pribadi)

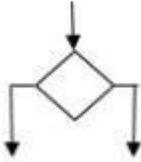
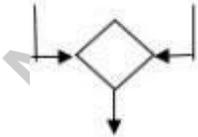
## 2. Activity Diagram

UML adalah instrumen untuk mengkarakterisasi aliran suatu proses atau kegiatan melalui suatu sistem. Aliran kontrol, percabangan, dan urutan kegiatan dari satu langkah proses ke langkah berikutnya digambarkan dalam diagram ini. Biasanya, diagram aktivitas digunakan dalam analisis dan desain sistem perangkat lunak serta dalam manajemen proses

bisnis untuk memahami alur kerja dan proses pengambilan keputusan dalam aktivitas tersebut.

Tabel 2. 2 Notasi Activity Diagram

Simbol	Nama	Keterangan
	<i>Initial state</i>	Simbol dimulainya alur suatu diagram aktivitas.
	<i>activities</i>	Aktivitas yang terjadi pada alur kerja.
	<i>Decision</i>	Simbol dimana adanya kemungkinan pada alur sistem.
	<i>Final state</i>	Simbol akhirnya alur suatu diagram aktivitas.
	<i>Swimlane</i>	Pembagian alur bisnis atas aktivitas yang terjadi.
	<i>Association</i>	Tanda untuk menunjukkan alur dari aktivitas diagram tersebut.

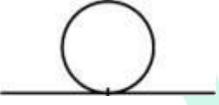
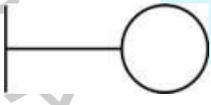
	<p><i>Decision node</i></p>	<p>Mengambarkan adanya kemungkinan lebih dari satu dari alur pada sistem.</p>
	<p><i>Merge node</i></p>	<p>Menggabungkan kembali alur yang dipisah oleh <i>decision node</i>.</p>
	<p><i>Final flow node</i></p>	<p>Akhir dari alur.</p>

### 3. Sequence

Dalam Unified Modeling Language (UML), diagram urutan adalah semacam diagram yang menunjukkan bagaimana objek berinteraksi dalam skenario atau proses. Rantai panggilan atau pesan yang dipertukarkan antar objek ditunjukkan pada gambar ini. Respons yang diberikan oleh masing-masing objek terhadap pesan tersebut. Secara khusus, Diagram alir (sequence diagram) memperlihatkan bagaimana objek saling berkomunikasi dalam rangkaian waktu tertentu dan mencakup informasi mengenai pengirim pesan, urutan pesan, dan tanggapan yang diterima. Diagram

ini berguna dalam memodelkan perilaku dinamis sistem atau proses, membantu pengembang memahami dan merancang interaksi antar objek untuk mencapai tujuan yang diinginkan.

Tabel 2. 3 Notasi *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Subjek yang mewakili peran interaksi pada sistem.
	<i>Entity Class</i>	Simbol yang menggambarkan bagaimana pesan ataupun panggilan metode melibatkan objek dari kelas entitas tersebut.
	<i>Boundary Class</i>	Simbol yang digunakan untuk berinteraksi dengan sistem lain, baik berupa <i>userinterface</i> dan lain sebagainya.

	<i>Control Class</i>	Digunakan untuk menggambarkan logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
	<i>Recursive</i>	Simbol di mana sebuah objek memanggil metode atau berinteraksi dengan objek yang memiliki tipe yang sama dengan dirinya sendiri.
	<i>Lifeline</i>	Menandakan panjang objek.

#### 4. *Class Diagram*

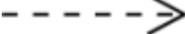
Diagram class dalam format statis sistem perangkat lunak digambarkan dalam diagram kelas yang dibuat menggunakan UML. Kelas objek, karakteristiknya, metode yang tersedia, dan dependensi serta hubungan di antara mereka semuanya digambarkan dalam diagram ini.

Class diagram berfungsi untuk memodelkan entitas-entitas utama dalam sistem perangkat lunak, menunjukkan bagaimana kelas-kelas tersebut berinteraksi dan saling berhubungan untuk menciptakan

fungsionalitas sistem secara keseluruhan. Diagram ini sering digunakan dalam tahap analisis dan desain sistem untuk mengidentifikasi entitas-entitas kunci, atribut, dan metode yang diperlukan untuk mengimplementasikan kebutuhan bisnis atau teknis yang ada.

Tabel 2. 3 Notasi *Class Diagram*

Simbol	Nama	Keterangan
	<p><i>Class</i></p>	<p>Deskripsi dari struktur sistem.</p>
	<p>Association</p>	<p>Garis yang menunjukkan bahwakelas-kelas tersebut berinteraksi atau berhubungan dalam suatu sistem.</p>
	<p>Aggregation</p>	<p>Hubungan antar kelas dengan makna semua bagian.</p>

	<p>Generalization</p>	<p>Hubungan antara dua atau lebih kelas yang menunjukkan bahwa kelas-kelas tersebut memiliki sifat atau perilaku yang sama atau tingkatan yang berbeda dalam suatu sistem.</p>
	<p>Dependency</p>	<p>Hubungan antara duakelas di mana perubahan pada kelas yang satu akan mempengaruhi kelas yang lainnya.</p>

Sumber: (Dokumentasi Pribadi)

### 2.1.8. *Testing*

Menurut (Sommerville, I., (2023) *Testing* adalah tahap krusial dalam pengembangan perangkat lunak yang bertujuan untuk mengevaluasi kualitas, kebenaran, dan kinerja suatu aplikasi. Pengujian aplikasi melibatkan serangkaian metode dan teknik yang dirancang tujuan dari langkah pengujian

adalah untuk memastikan aplikasi berfungsi sebagaimana dimaksud dan sesuai dengan persyaratan yang ditetapkan. Tujuan utamanya adalah untuk memastikan aplikasi dapat diandalkan, aman, dan memenuhi ekspektasi pengguna akhir dalam penggunaannya.

Metode dan teknik pengujian yang beragam digunakan untuk mengidentifikasi kelemahan atau bug dalam aplikasi sebelum dirilis ke pengguna akhir. Hal ini mencakup pengujian fungsionalitas, pengujian non-fungsionalitas seperti keamanan dan kinerja, serta pengujian integrasi untuk memastikan komponen-komponen aplikasi bekerja dengan baik bersama-sama.

Pengujian aplikasi tidak hanya melibatkan validasi fungsional dan teknis, tetapi juga aspek-aspek seperti kemudahan penggunaan (usability), keamanan data, dan responsivitas aplikasi terhadap input pengguna. Keseluruhan, pengujian aplikasi memainkan peran penting dalam memastikan bahwa aplikasi yang dikembangkan untuk memastikan bahwa aplikasi memberikan manfaat maksimal bagi pengguna akhir.

## **2.2 Tinjauan Studi**

Bagian ini menyajikan temuan penyelidikan ini serta ikhtisar studi sebelumnya tentang topik tersebut. Literatur yang dipilih terhubung dan relevan dengan topik penyelidikan ini, dan ini terdiri dari:

- 1) Jurnal disusun oleh Meri Audrilia dan Arief Budiman dan diterbitkan di JURNAL MADANI: Science, Technology, and Humanities, Volume 3, nomor 1, Maret 2020, dengan judul "Designing a Web-Based Workshop Management Information System (Studi Kasus: Bengkel Anugrah)," membahas masalah yang dihadapi oleh Bengkel Anugrah. Pada awalnya, bengkel ini melakukan semua transaksi secara manual dengan mencatatnya di buku penjualan, yang mengakibatkan masalah dalam pengelolaan stok barang. Stok sering habis tanpa pemberitahuan, dan pelanggan harus datang langsung ke bengkel untuk melakukan servis atau

memeriksa ketersediaan barang. Untuk mengatasi masalah ini, penulis mengadopsi metode web engineering yang meliputi langkah-langkah Communication, Planning, Modeling, Construction, dan Deployment, serta menggunakan UML saat merancang sistem. Program berbasis web yang dibuat dengan MySQL dan Dreamweaver sebagai basis data, sehingga Bengkel Anugrah dapat menciptakan sistem manajemen bengkel yang memungkinkan pencatatan transaksi secara efisien, memberikan informasi stok barang secara real-time, dan meningkatkan layanan pelanggan. Bengkel Anugrah menyediakan jasa perawatan dan menjual sparepart untuk kendaraan roda empat. Temuan penulis berdasarkan wawancara dengan pemilik bengkel mengungkapkan bahwa proses sistem di bengkel ini dimulai dengan perawatan kendaraan pelanggan dan pengumpulan data penjualan serta layanan berdasarkan layanan yang diberikan. Namun, sistem tersebut belum terkomputerisasi. Setiap transaksi masih dilakukan secara manual dan dicatat dalam buku penjualan, serta tidak ada laporan penjualan yang dihasilkan selama periode tersebut. Masalah kehabisan stok barang sering terjadi, karena Pelanggan harus berkunjung sendiri untuk menanyakan ketersediaan barang, karena tidak ada pemberitahuan stok habis.

- 2) Rinto Diantara, Siswanto, dan Yupianti (2021) dalam artikel mereka “Perancangan Aplikasi Sistem Booking Service Online Berbasis Web dengan Menggunakan Metode Software Development Life Cycle” membahas tentang peningkatan permintaan layanan pemesanan servis kendaraan yang mendorong sebuah bengkel untuk mengembangkan sistem yang mempermudah proses pemesanan dan manajemen data secara efisien. Penelitian ini diharapkan untuk mengembangkan aplikasi layanan pemesanan online yang berbasis

web, dengan tujuan meningkatkan efektivitas operasional bengkel dan kualitas layanan. Penelitian ini bertujuan untuk membuat aplikasi yang dapat mendukung lokakarya di mengelola data pemesanan servis kendaraan secara terstruktur. Aplikasi ini juga dimaksudkan untuk mempermudah proses pemesanan, memudahkan administrasi, dan memberikan informasi yang akurat kepada customer. Metode Sistem pengembangan perangkat lunak (SDLC) digunakan dalam penelitian ini sebagai panduan pengembangan aplikasi. Prosedur SDLC, yang terdiri dari pengorganisasian, analisis, desain, implementasi, pengujian, dan pemeliharaan, diintegrasikan dalam pengembangan aplikasi ini. Aplikasi ini memiliki dua antarmuka: antarmuka pelanggan dan administrator. Fungsi utama antarmuka pelanggan melibatkan proses pemesanan servis kendaraan, pemilihan paket servis, penambahan layanan tambahan, dan estimasi biaya. Sementara itu, antarmuka administrator memberikan akses untuk kontrol admin, klien, reservasi, mekanik, halaman, dan informasi kontak. Pembuatan faktur pembayaran adalah fitur lain yang didukung oleh aplikasi ini. Pengujian aplikasi dilakukan baik secara online maupun offline. Hasil pengujian menunjukkan bahwa aplikasi ini mampu mengelola data booking service dengan baik, memberikan informasi lengkap tentang layanan yang ditawarkan, dan berhasil menghasilkan invoice pembayaran. Aplikasi ini juga diuji menggunakan kuesioner, dan hasilnya menunjukkan bahwa Perangkat lunak ini memfasilitasi pengelolaan data yang terkait dengan layanan pemesanan dan memberikan informasi yang bermanfaat kepada pelanggan. Aplikasi ini memberikan manfaat signifikan bagi bengkel dalam hal efisiensi operasional, peningkatan kualitas layanan, dan kemudahan akses bagi pelanggan. Pembuatan invoice secara otomatis juga membantu mempercepat proses

pembayaran dan meningkatkan transparansi biaya layanan. Dengan mengimplementasikan melalui penggunaan sistem layanan pemesanan online berbasis web ini, bengkel akan dapat meningkatkan kepuasan pelanggan, merampingkan operasi, dan mengantarkan era baru pengelolaan data layanan pemesanan mobil digital.

- 3) Alter Junior Fernando Pangalila dan Maulana Ardhiansyah (2022) dalam artikel mereka "Perancangan Sistem Aplikasi Management Booking Service Bengkel Berbasis Website Dengan Metode Prototype" membahas tentang masalah yang dihadapi oleh Bengkel Rido Racing. Bengkel ini melayani berbagai jenis layanan servis kendaraan seperti servis ringan, servis berat, ganti oli, dan pengisian angin ban. Meskipun memiliki banyak pelanggan setiap hari, bengkel ini mengalami kendala dalam efisiensi layanan. Masalah-masalah yang diidentifikasi termasuk ketidakjelasan antrian pelanggan, kurangnya informasi mengenai status pekerjaan mekanik, ketidakpastian waktu buka-tutup bengkel, serta ketidakteraturan informasi mengenai harga layanan dan sparepart. Penelitian ini menggunakan teknik prototipe untuk membuat sistem aplikasi manajemen layanan pemesanan berbasis website dengan tujuan meningkatkan efisiensi layanan di Rido Racing Workshop. Dengan sistem ini, diharapkan pelanggan dapat mengakses informasi dengan lebih transparan, memahami antrian dengan jelas, mempercepat proses layanan, dan mendapatkan informasi yang akurat terkait harga layanan dan sparepart. Penelitian ini menggunakan metode prototype dalam perancangan sistem aplikasi. Tahapan penelitian meliputi analisis kebutuhan pengguna, perancangan antarmuka, pembuatan prototipe, uji coba, dan pengembangan lanjutan berdasarkan umpan balik pengguna. Sistem

ini diharapkan dapat memperbaiki proses servis motor di bengkel, membuatnya lebih efisien, dan meningkatkan kepuasan pelanggan sehingga tidak ada lagi keluhan terkait waktu pengerjaan yang lama. Penulis menggunakan Xampp, PHP, dan Visual Studio Code untuk membuat website. Sistem yang akan dibangun berbentuk aplikasi web dengan Xampp sebagai inti sistem dashboard dan MySQL sebagai media penyimpanan data (Nurmiati & Hafidz, 2021). Informasi bengkel akan ditampilkan melalui browser pada ponsel pintar atau perangkat konsumen yang akan digunakan oleh mekanik dan pemilik bengkel (Yanuar et al., 2022).

- 4) Yandhika Saputra dan Ardiansyah Dores dalam artikel mereka yang berjudul "Analisa Dan Perancangan Sistem Informasi Pelayanan Jasa Bengkel Service Motor Online Berbasis Web" membahas tentang pentingnya perawatan rutin untuk sepeda motor di Indonesia yang telah dikenal sejak tahun 1893. Dengan jutaan sepeda motor yang beredar, baik di perkotaan maupun pedesaan, perawatan yang rutin sangat penting untuk memastikan kendaraan tetap dalam kondisi optimal. Banyak bengkel non-resmi bermunculan untuk memenuhi kebutuhan ini, dan penyedia jasa bengkel harus menjaga kualitas layanan untuk tetap bersaing. Bengkel sepeda motor biasanya menawarkan layanan perbaikan, penyetelan, dan suku cadang bagi pelanggan, dengan ketepatan waktu sebagai faktor kunci. Saat ini, banyak bengkel menyediakan opsi reservasi layanan, namun kebanyakan masih dilakukan secara manual, baik melalui kunjungan langsung maupun telepon. Proses manual ini bisa menjadi membosankan dan memerlukan pembukuan yang teliti untuk melacak reservasi yang banyak. Penelitian ini adalah untuk menciptakan sistem reservasi berbasis web yang responsif. Database PHP dan MySQL digunakan dalam konstruksi sistem,

dimulai dengan analisis infrastruktur dasar. Pengembangan sistem ini diharapkan dapat memenuhi kebutuhan pelanggan untuk memesan layanan sebelumnya dan menemukan bengkel terdekat secara otomatis. Pengujian dengan metode Black box menunjukkan bahwa semuanya beroperasi sebagaimana mestinya dengan sistem. Jurnal Sistem Informasi, Teknologi Informasi dan Komputer, Volume 11, No. 3, Mei 2021, memuat artikel yang diterbitkan. Pemanfaatan yang dikembangkan dalam penelitian ini menghadirkan solusi dengan dukungan desain web responsif, memastikan data akurat dan memberikan keleluasaan bagi pengelola bengkel untuk memberikan informasi yang tepat waktu kepada pelanggan.

- 5) Sri Mulyani dan Fauzan Natsir dalam karya mereka yang berjudul "Sistem Pakar Diagnosis Kerusakan Sepeda Motor di Bengkel Rahmat Cort menggunakan Metode Forward Chaining" membahas tentang pentingnya transportasi dalam kehidupan sehari-hari masyarakat. Transportasi, yang kini dianggap sebagai kebutuhan dasar, memerlukan kesinambungan dan ketersediaan layanan untuk memenuhi kebutuhan produksi, konsumsi, dan distribusi (Menteri Perhubungan Budi Karya Sumadi, 2022). Banyak orang bergantung pada sepeda motor sebagai alat transportasi utama, bahkan sebagian menggunakannya untuk menambah penghasilan. Sepeda motor lebih disukai karena kenyamanannya dibandingkan jenis non-matic. Dalam aktivitas perkotaan yang padat, kecepatan dan akurasi menjadi penting. Namun, Kurangnya pemahaman tentang kerusakan sepeda motor sering menjadi masalah, terutama bagi pengendara yang tidak menyadari berbagai jenis kerusakan. Tujuan dari pekerjaan ini adalah untuk membuat perangkat lunak yang akan membantu pemilik sepeda motor dalam mengidentifikasi

kerusakan. Hasil dari penelitian berupa aplikasi untuk bengkel motor yang menggunakan pendekatan Forward Chaining, teknik pencarian data atau fakta untuk mengidentifikasi dan mengatasi masalah yang dihadapi. Aplikasi ini memungkinkan pengguna mendiagnosis kerusakan sepeda motor dan menjadwalkan servis, sehingga mekanik bisa datang langsung ke rumah. Bengkel Rahmat Cort, menurut pemiliknya, masih menggunakan sistem manual untuk pengecekan mesin yang rusak dan menerima 8 hingga 15 kunjungan sepeda motor per hari. Semua pengguna menginginkan sepeda motor mereka tetap dalam kondisi baik dan bebas dari gangguan atau bahaya. Namun, sepeda motor otomatis sering mengalami masalah. yang menghambat aktivitas pengendara, sering disebabkan oleh kurangnya pemahaman tentang perawatan sepeda motor matic yang rusak. Ketika menangani kerusakan, pelanggan harus selalu menginformasikan mekanik tentang gejala yang dialami pada sepeda motor. Proses manual ini memakan banyak waktu bagi pelanggan, mulai dari berangkat ke bengkel hingga perbaikan selesai. Oleh karena itu, dibutuhkan aplikasi yang dapat menganalisis, menemukan, dan memberikan solusi untuk masalah kerusakan sepeda motor.