

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

2.1.1. Pengembangan

Menurut KBBI, pengembangan didefinisikan sebagai proses, cara, atau perbuatan mengembangkan (membangun, memperluas, memajukan, memperbaiki dan lain-lain). Meskipun demikian, pengembangan umumnya dapat merujuk pada proses atau tindakan untuk membuat, memperluas, atau meningkatkan sesuatu. Sehingga dapat diartikan bahwa, pengembangan diartikan sebagai proses atau tindakan untuk membuat atau memperluas sesuatu menjadi lebih besar, lebih baik, atau lebih maju. Ini bisa merujuk pada pengembangan fisik, seperti membangun infrastruktur, atau pengembangan non-fisik, seperti meningkatkan keterampilan atau pengetahuan seseorang.

a) Pengembangan Sistem Informasi

Proses membuat, merancang, menerapkan, dan merawat sistem informasi untuk memenuhi kebutuhan atau target bisnis tertentu dikenal sebagai pengembangan sistem informasi. Proses ini termasuk mengevaluasi sistem yang sudah ada, menentukan kebutuhan, membuat rencana solusi, membuat perangkat lunak atau aplikasi, melakukan pengujian, dan menerapkan sistem akhir (Dennis, 2015). Dengan menggunakan teknologi yang tepat, pengembangan sistem informasi dimaksudkan untuk meningkatkan efisiensi bisnis, membantu pengambilan keputusan yang lebih baik, meningkatkan efisiensi, dan membantu organisasi mencapai tujuannya. Metode ini memerlukan kombinasi keahlian teknis dan pemahaman yang mendalam tentang kebutuhan bisnis untuk membuat solusi yang menguntungkan yang juga mendukung kemajuan perusahaan.

b) Pengembangan Perangkat Lunak

Mengembangkan, merancang, mengkode, menguji, dan merawat aplikasi atau sistem perangkat lunak disebut pengembangan perangkat lunak . Upaya terstruktur untuk membuat solusi perangkat lunak yang memenuhi kebutuhan pengguna atau bisnis dikenal sebagai proses ini (Dennis, 2015). Pengumpulan persyaratan, perancangan, penulisan kode, pengujian, penerapan, dan pemeliharaan adalah semua tahapan pengembangan perangkat lunak. Pengembang melakukan proses ini dengan menggunakan berbagai alat, bahasa pemrograman, dan kerangka kerja untuk membuat aplikasi yang berfungsi dengan baik dan memenuhi standar kualitas dan harapan pengguna. Keberhasilan dalam pengembangan perangkat lunak bergantung pada kemampuan untuk bekerja sama, memecahkan masalah, memperhatikan detail, dan membuat aplikasi yang handal dan mudah digunakan.

c) Pengembangan Proses Bisnis

Analisis, desain, pelaksanaan, dan peningkatan proses bisnis dilakukan untuk meningkatkan efisiensi, produktivitas, dan kinerja organisasi secara keseluruhan. Fokusnya adalah memahami cara alur kerja yang beroperasi, menemukan area yang perlu diperbaiki, dan menerapkan perubahan untuk menyederhanakan alur dan mencapai tujuan yang direncanakan (Dennis, 2015). Dengan menggabungkan proses dengan tujuan dan praktik terbaik industri, pengembangan proses bisnis bertujuan untuk meningkatkan alur kerja, mengurangi biaya, mengatasi masalah, dan meningkatkan kepuasan pelanggan. Ini biasanya mencakup pemetaan proses yang ada, menemukan masalah, merancang ulang alur kerja, menerapkan teknologi baru, dan terus memperbaiki proses untuk mendorong inovasi dan perbaikan yang berkelanjutan dalam perusahaan.

d) Pengembangan Basis Data

Merancang, membuat, menerapkan, dan merawat basis data untuk menyimpan, mengatur, dan mengelola data dengan baik disebut pengembangan basis data. Ini mencakup menentukan jenis data, menentukan hubungan antar tabel, menetapkan struktur basis data, dan menetapkan batasan untuk memastikan bahwa data tetap utuh dan konsisten (Dennis, 2015). Menulis kueri, prosedur tersimpan, dan pemicu untuk menarik dan memanipulasi data serta meningkatkan kinerja basis data untuk meningkatkan akses dan skalabilitasnya juga merupakan bagian dari pengembangan basis data. *Database Management System* (DBMS) seperti *MySQL*, *Oracle*, *SQL Server*, dan *PostgreSQL* digunakan oleh pengembang untuk membangun dan mengelola basis data yang mendukung aplikasi, situs web, dan operasi bisnis. Untuk menyimpan dan mengambil data dengan aman, akurat, dan sesuai dengan kebutuhan pengguna dan aplikasi, pengembangan basis data yang efektif sangat penting.

e) **Pengembangan Antarmuka**

Pengembangan antarmuka adalah proses merancang dan membuat antarmuka pengguna (UI) aplikasi atau sistem perangkat lunak. Ini termasuk membuat elemen visual, komponen interaktif, dan fitur navigasi yang berinteraksi dengan pengguna untuk memungkinkan penggunaan aplikasi dengan efektif (Dennis, 2015). Tujuan utama pengembangan antarmuka adalah membuat antarmuka yang mudah dipahami, ramah pengguna, dan menarik secara visual untuk meningkatkan pengalaman pengguna secara keseluruhan. Dalam prosesnya membuat elemen seperti tata letak, tombol, menu, formulir, ikon, dan grafis lainnya yang membentuk antarmuka pengguna (UI) aplikasi. Untuk melakukannya dengan baik, pengembang menggunakan berbagai teknologi, kerangka kerja, dan prinsip

desain. Selain itu, proses ini memeriksa konsistensi elemen desain, responsif pada berbagai perangkat, aksesibilitas bagi orang dengan disabilitas, dan pengujian kegunaan untuk mengumpulkan dan memperbaiki umpan balik.

2.1.2. Sistem Informasi

Menurut (Dennis, 2015), sistem informasi adalah suatu struktur yang kompleks dan terorganisir yang terdiri dari berbagai elemen yang saling terkait. Elemen-elemen ini bekerja sama untuk melakukan sejumlah tugas penting dalam suatu organisasi. Pertama-tama, sistem informasi bertanggung jawab untuk mengumpulkan data dari berbagai sumber, baik sumber internal maupun eksternal organisasi. Data yang terkumpul ini kemudian diolah melalui serangkaian operasi atau proses tertentu hingga menghasilkan informasi yang relevan dan berguna bagi organisasi. Informasi yang dihasilkan dari proses ini kemudian disimpan dalam berbagai bentuk, mulai dari database elektronik hingga dokumen fisik, tergantung pada jenis informasi dan kebutuhan organisasi. Pentingnya pengelolaan data ini adalah untuk memastikan informasi dapat diakses dengan mudah dan aman, serta dapat dikelola secara optimal.

a) Karakteristik Sistem Informasi

Menurut (Dennis, 2015), sistem informasi yang merupakan bagian penting dari operasi suatu organisasi, terdiri dari banyaknya karakteristik penting yang membantu mencapai tujuan organisasi dan meningkatkan kinerjanya. Salah satu karakteristik utama sistem informasi adalah untuk menyediakan infrastruktur yang memungkinkan pengumpulan, penyimpanan, pemrosesan, dan distribusi informasi yang membantu berbagai aspek pengelolaan organisasi, seperti pengambilan keputusan, koordinasi, pengendalian, analisis, dan visualisasi. Komponen sistem informasi termasuk data, perangkat lunak, perangkat keras, dan prosedur, serta peran aktif dari orang-orang yang

bekerja sama untuk mengelola informasi. Untuk mendukung berbagai proses bisnis dan aktivitas organisasi, sistem informasi menangani berbagai jenis data, mulai dari yang terstruktur (misalnya *database*) hingga yang tidak terstruktur (misalnya dokumen) dan semi-terstruktur (misalnya file XML). Data dikumpulkan, disimpan, diproses, diambil, dan didistribusikan melalui proses dan prosedur yang terintegrasi dalam sistem informasi. Selain itu, sistem informasi menawarkan antarmuka interaktif yang memungkinkan pengguna berinteraksi dengan sistem, memasukkan data, mengambil data, membuat laporan, dan berinteraksi dengan sistem. Untuk memastikan bahwa informasi konsisten, akurat, dan mudah diakses, proses dan data harus diintegrasikan di berbagai departemen, fungsi, dan tingkat organisasi. Selain itu, keamanan data sangat penting. Langkah-langkah seperti otentikasi pengguna, enkripsi data, kontrol akses, dan prosedur pencadangan digunakan untuk melindungi data dari akses yang tidak sah. Kemudian, sistem informasi yang fleksibel memungkinkan perubahan kebutuhan bisnis, perkembangan teknologi, dan permintaan pengguna. Alat dan kemampuan seperti analisis data, pelaporan, visualisasi, dan perkiraan membantu pengguna membuat keputusan yang akurat. Selain itu, sistem informasi memungkinkan pengguna untuk memberikan umpan balik, saran, dan permintaan perbaikan untuk meningkatkan kegunaan, fungsionalitas, dan kinerja sistem secara keseluruhan. Oleh karena itu, keseluruhan karakteristik ini membantu organisasi dalam mengelola informasi, mendukung proses bisnis, dan memudahkan pengambilan keputusan.

b) Komponen Sistem Informasi

Menurut (Dennis, 2015), komponen sistem informasi, juga disebut kerangka sistem informasi, adalah komponen

penting yang bekerja sama untuk mengelola informasi di suatu organisasi. Perangkat lunak, perangkat keras, data, prosedur, dan orang-orang yang terlibat dalam penggunaan sistem termasuk dalam komponen ini. Perangkat keras mencakup semua perangkat fisik, seperti komputer dan server, yang digunakan untuk memproses data, dan perangkat lunak mencakup program dan aplikasi yang menjalankan operasi sistem informasi. Sementara prosedur adalah peraturan dan instruksi yang mengatur penggunaan sistem, data adalah informasi mentah yang dikumpulkan dan diproses oleh sistem. Terakhir, orang yang terlibat dalam sistem terdiri dari pengguna, pemangku kepentingan, dan profesional TI yang mengelola dan memelihara sistem. Untuk memastikan bahwa sistem informasi berfungsi dengan baik dan mendukung kegiatan organisasi, sangat penting untuk memastikan bahwa komponen-komponen ini selaras dan terintegrasi satu sama lain.

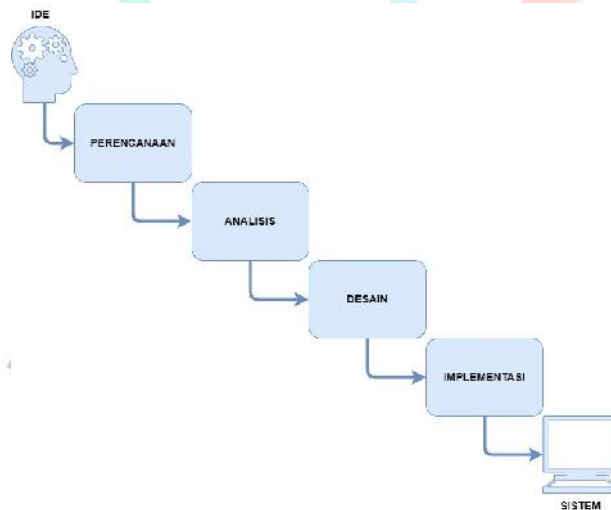
c) Fungsi Sistem Informasi

Menurut (Dennis, 2015), sistem informasi berfungsi sebagai bagian organisasi, seperti pengambilan keputusan, koordinasi, pengendalian, analisis, dan visualisasi, dengan menghimpun, menyimpan, memproses, mengekstraksi, dan menyebarkan informasi. Sistem informasi sangat penting dalam mengelola data dan mengubahnya menjadi informasi yang dapat digunakan dan signifikan, yang dapat digunakan oleh individu, departemen, dan manajemen untuk mencapai tujuan dan target organisasi. Secara keseluruhan, fungsi utama sistem informasi adalah manajemen informasi dengan efektif, mendukung proses bisnis, meningkatkan pengambilan keputusan, memperbaiki komunikasi, dan meningkatkan kinerja dan keberhasilan organisasi.

2.1.3. *System Development Life Cycle (SDLC)*

Untuk memastikan keberhasilan dan kesesuaian dengan kebutuhan organisasi atau perusahaan, pengembangan sistem informasi memerlukan proses formal. Tujuan pembuatan dan penyusunan sistem baru adalah untuk menggantikan sistem lama atau meningkatkan sistem yang ada agar lebih optimal dalam menjalankan proses bisnis organisasi atau perusahaan. Proses formal yang dimaksud yaitu Siklus hidup pengembangan sistem atau dikenal dengan sebutan SDLC (*system development life cycle*).

Menurut (Dennis, 2015) menyatakan bahwa proses Pengembangan Siklus Hidup Perangkat Lunak (*System Development Life Cycle*) terdiri dari empat tahap inti yaitu: perencanaan, analisis, desain, dan implementasi. Meskipun fokus proyek mungkin berbeda mengenai aspek tertentu dari SDLC atau menggunakan pendekatan yang berbeda untuk melihat tahapan SDLC, semua proyek memiliki komponen dari keempat tahap tersebut. Setiap langkah terdiri dari sejumlah tindakan yang menggunakan metode khusus untuk mencapai hasil yang diinginkan. Dalam SDLC, ada empat tahap yakni:



Gambar 2. 1 Siklus Hidup Pengembangan Sistem

SUMBER: Dokumentasi Pribadi

A. Perencanaan (*Planning*)

Pada tahapan ini merupakan langkah awal yang bertujuan untuk mendapatkan pemahaman tentang sistem atau aplikasi yang akan dibangun dan rencana pelaksanaan yang akan digunakan oleh tim proyek. Tahap ini diperlukan karena untuk melakukan evaluasi dampak sistem yang akan dibangun terhadap nilai bisnis, baik dalam hal pengurangan biaya atau peningkatan pendapatan, serta memahami bagaimana sistem tersebut akan mempengaruhi alur bisnis, apakah akan mempermudah atau mempersulit pengguna. Sebagian besar permintaan sistem baru berasal dari berbagai departemen seperti pemasaran, keuangan, SDM, produksi dan lain-lain. Permintaan sistem baru memerlukan dokumen, yang disebut dengan dokumen permintaan sistem. Dokumen ini menyajikan ringkasan singkat tentang kebutuhan bisnis dan cara sistem dapat membantu menciptakan nilai bisnis. Selanjutnya, tim pengembang melakukan perencanaan kelayakan, tim pengembang sistem informasi bekerja sama dengan pihak yang mengajukan permintaan. Perencanaan ini dilakukan oleh tim pengembang untuk memastikan bahwa kebutuhan dapat dipenuhi. Perencanaan ini meliputi tiga aspek yaitu:

1. kelayakan Teknis

Aspek ini bertujuan untuk memastikan apakah ada kebutuhan akan teknologi baru atau apakah sistem usulan dapat dikembangkan dan diterapkan menggunakan teknologi yang ada saat ini. Contoh pertanyaan kelayakannya seperti: apakah sistem dapat dibangun?

2. Kelayakan Organisasi

Aspek ini bertujuan untuk memastikan bahwa sistem usulan akan berfungsi dengan baik untuk mendukung proses bisnis organisasi setelah dibangun dan diterapkan. Contoh pertanyaan kelayakannya seperti: apakah sistem akan digunakan setelah dibangun?

3. Kelayakan Ekonomi

Aspek ini bertujuan untuk memastikan bahwa sistem usulan saat ini akan menghasilkan keuntungan dari sisi biaya (lebih efisien) jika diimplementasikan, dan untuk memastikan bahwa dana yang ada tersedia untuk mendukung estimasi biaya sistem usulan. Contoh pertanyaan kelayakannya seperti: akankah itu menghasilkan keuntungan bagi perusahaan?

B. Analisis (*Analysis*)

Pada tahapan ini data yang sudah dikumpulkan pada tahapan perencanaan kemudian di analisa. Lebih rincinya pada tahap ini, tim pengembang sistem informasi melakukan pengamatan mendalam dan pemahaman tentang sifat, fungsi, dan komponen organisasi bersama dengan kebutuhannya dilakukan. Tahap analisis ini juga digunakan untuk menentukan persyaratan yang diperlukan. Ini mencakup identifikasi proses yang berhubungan dengan pengguna sistem, alur sistem, dan lokasi penggunaan sistem. Contoh beberapa pertanyaan yang harus terjawab dalam tahap analisis yaitu: siapa yang akan memanfaatkan sistem, fungsi-fungsi yang akan dijalankan oleh sistem, serta waktu dan tempat penggunaannya sistem yang akan dikembangkan. Dalam menjawab pertanyaan-pertanyaan tersebut tim pengembang sistem informasi menggunakan beberapa teknik

seperti: wawancara, observasi, kuesioner, JAD (*joint application development*), dan analisis dokumen. Teknik tersebut adalah teknik-teknik yang dapat digunakan dalam tahap analisis untuk memahami sistem yang ada serta kebutuhan sistem baru. Teknik-teknik ini juga dapat menjawab pertanyaan mengenai siapa, apa, kapan, dan di mana terkait dengan sistem saat ini dan yang akan datang. Pada tahapan ini juga tim pengembang sistem informasi akan menentukan beberapa persyaratan, persyaratan yang dimaksud ialah persyaratan apa saja yang harus dilakukan sistem atau fitur baru. Persyaratan-persyaratan tersebut adalah: Persyaratan tentang kebutuhan yang diperlukan oleh organisasi atau bisnis (*business requirements*), apa yang harus dilakukan oleh pengguna (*user requirements*), fungsi yang harus dapat dijalankan oleh perangkat lunak (*functional requirements*), karakteristik yang harus dimiliki oleh sistem (*nonfunctional requirements*), dan cara pembangunan sistem (*system requirements*)."

C. Perancangan (*Design*)

Pada tahapan ini, perancangan dilakukan untuk memahami bagaimana aplikasi diterapkan pada perangkat keras, perangkat lunak, dan jaringan yang tersedia. Proses ini juga mencakup menentukan spesifikasi antarmuka pengguna (*User Interface*), formulir, laporan, *database*, dan berkas yang diperlukan. Tahap *design* memiliki empat langkah yaitu:

1. Menentukan strategi desain, termasuk apakah sistem akan dikembangkan secara internal, oleh perusahaan konsultan IT, atau dengan membeli paket perangkat lunak yang sudah tersedia.

2. Menjelaskan apakah sistem baru akan menambah atau mengubah infrastruktur yang ada dalam organisasi. Desain antarmuka menentukan interaksi pengguna dengan sistem, seperti navigasi menggunakan menu dan tombol, serta formulir dan laporan yang akan digunakan dan dihasilkan oleh sistem.
3. Menggambarkan database dan spesifikasi file sebagai tempat penyimpanan data yang diperlukan dan dihasilkan oleh sistem.
4. Mengembangkan rancangan proses yang akan dikonversi menjadi program, serta mendefinisikan tugas dari setiap program.

Setelah langkah-langkah di atas selesai dilaksanakan, hasil desain didokumentasikan dan diserahkan kepada tim pemrograman untuk diimplementasikan dan menghasilkan keluaran berupa:

1. Desain keluaran mencakup bentuk laporan dan dokumen yang dirancang.
2. Desain masukan meliputi tata letak layar untuk memasukkan data.
3. Desain antarmuka mencakup interaksi antara pengguna dan sistem, termasuk menu, ikon, bentuk formulir, tata letak, dan lainnya.
4. Desain database mencakup rancangan database dan spesifikasi file, seperti nama file, tipe data, panjang karakter, dan sebagainya.
5. Desain modul mencakup rancangan program dan alur kerjanya.

D. Implementasi (*Implementation*)

Dalam Siklus Hidup Pengembangan Perangkat Lunak (SDLC), implementasi adalah fase tertinggi. Pada tahapan ini, sistem harus dibangun berdasarkan hasil desain. Tahap ini biasanya mendapat perhatian terbesar karena merupakan tahap terpanjang dalam proses pengembangan sistem. Tahapan implementasi memiliki beberapa langkah seperti:

1. Pemrograman merupakan langkah pembuatan program sesuai dengan rancangan dan kebutuhan sistem.
2. Pengujian dilakukan untuk memastikan sistem berfungsi sesuai dengan kebutuhan dan desain. Ada empat tahap pengujian: pengujian unit, pengujian integrasi, pengujian sistem, dan pengujian penerimaan.
3. Instalasi merupakan langkah di mana sistem lama dimatikan dan sistem baru diaktifkan. Ada beberapa metode untuk transisi dari sistem lama ke sistem baru, seperti langsung, paralel, dan bertahap.
4. Dokumentasi penting karena digunakan sebagai acuan untuk penggunaan dan pemeliharaan sistem. Dokumentasi berisi informasi rinci tentang sistem, spesifikasi desain, cara kerja, dan fungsinya. Terdapat tiga bentuk dokumen: dokumen referensi, manual prosedur, dan panduan pengguna atau tutorial.
5. Pelatihan dan dukungan pengguna sangat penting. Langkah ini memberikan pemahaman kepada pengguna akhir tentang cara menggunakan sistem dengan baik, serta memberikan dukungan dalam penggunaannya, seperti menjawab pertanyaan dan

membantu memahami cara menjalankan fungsi tertentu.

6. Pemeliharaan sistem adalah proses penyempurnaan untuk memastikan sistem memenuhi kebutuhan bisnis. Kadang-kadang, hasil dari proses ini bisa berupa pengembangan sistem baru.

2.1.4. Rapid Application Development (RAD)

Menurut (Pratama, 2023) Pengembangan aplikasi cepat (*Rapid Application Development*) atau RAD adalah model proses pembangunan perangkat lunak yang dilakukan secara bertahap, terutama untuk proyek yang memiliki batas waktu yang singkat. Dalam pengembangan aplikasi yang cepat, pendekatan iteratif digunakan untuk membangun sistem. Metode ini bertujuan untuk membuat model kerja sistem pada awal tahap pengembangan untuk mengidentifikasi kebutuhan pengguna dan mengatasi masalah yang muncul. Model kerja ini dapat berfungsi sebagai landasan untuk desain dan implementasi sistem akhir.

Menurut (Pratama, 2023) *Rapid Application Development* (RAD) merupakan kombinasi teknik prototyping dengan *Join Application Development* (JAD) yang bertujuan untuk mendefinisikan dengan jelas kebutuhan aplikasi dan mempercepat proses pembangunan sistem. Tahapan



Gambar 2. 2 Siklus *Rapid Application Development*

metode *Rapid Application Development* (RAD) meliputi Perencanaan Persyaratan (*Requirement Planning*), Pemodelan Desain (*Design Workshop*), dan Penerapan Implementasi (*Implementation*).

1. Perencanaan Persyaratan (*Requirement Planning*)

Pada tahapan ini akan dilakukan pengidentifikasian dan analisis kebutuhan sistem yang akan dibangun untuk mencapai tujuan. Dalam kebanyakan kasus, proses ini memakan waktu antara satu hingga empat minggu. Pengembang dan klien harus bekerja sama untuk mencapai tujuan dan harapan proyek dan untuk mengatasi masalah yang mungkin muncul selama proses perancangan dan pembangunan.

2. Pemodelan Desain (*Design Workshop*)

Pada tahapan ini sistem akan dirancang untuk memiliki fungsionalitas yang diperlukan dan mudah digunakan. Untuk memenuhi persyaratan dan memenuhi kebutuhan aplikasi yang tepat, desain disusun secara menyeluruh. Penerapan model sesuai keinginan pengguna dan pemahaman tentang pembuatan sistem sesuai dengan prosedur yang ada juga dilakukan.

3. Penerapan Implementasi (*Implementation*)

Pada tahapan ini akan dilakukan penerapan metode pemrograman pada desain yang telah dibuat, yang dijelaskan secara rinci dalam tahap implementasi database dan penulisan kode. Tujuan dari tahap ini adalah untuk membangun sistem dan menerapkan metode pemrograman sesuai dengan kebutuhan sistem.

2.1.5. *Object Oriented Analysis and Design (OOAD)*

Menurut (Dennis, 2015) Analisis dan Desain Berorientasi Objek (OOAD) adalah sebuah metodologi yang menggabungkan ide-ide pemrograman berorientasi objek dengan pendekatan sistematis untuk menganalisis, merancang, dan mengembangkan sistem perangkat lunak. Metodologi ini menekankan pemodelan sistem sebagai kumpulan objek yang berinteraksi untuk melakukan fungsi tertentu. Metode OOAD berfokus pada mengidentifikasi objek dalam domain masalah, menentukan karakteristik dan perilaku objek tersebut, dan memodelkan interaksi di antara mereka untuk merancang solusi perangkat lunak yang secara akurat mencerminkan sistem dunia nyata. Teknik ini membantu dalam pembuatan sistem perangkat lunak yang modular, dapat digunakan kembali, dan mudah dipelihara.

Aspek utama OOAD meliputi:

1. Analisis Berorientasi Objek (OOA): Ini mencakup pengidentifikasian objek, hubungan mereka, dan persyaratan yang menentukan perilaku sistem.
2. Desain Berorientasi Objek (OOD): Melibatkan penerapan model desain menggunakan bahasa pemrograman berorientasi objek seperti Java, C++, atau Python.
3. Pemrograman Berorientasi Objek (OOP): Melibatkan penerapan model desain menggunakan kelas, antarmuka, dan interaksi untuk mengimplementasikan sistem.

OOAD mendorong pendekatan pengembangan perangkat lunak yang terstruktur dan iteratif, yang menekankan pemahaman domain masalah, penggunaan prinsip berorientasi objek dalam desain solusi, dan penerapan desain tersebut untuk membangun sistem perangkat lunak yang stabil dan dapat diukur. Selain itu, Proses perancangan yang menggunakan *Object Oriented Analysis and Design (OOAD)* membutuhkan penggunaan teknik tertentu yang bergantung pada *Unified Modeling Language (UML)* sebagai

sarana standar dalam pembuatan sistem perangkat lunak yang berbasis objek.

2.1.6. *Unified Modeling Language (UML)*

Menurut (Dennis, 2015) UML merupakan salah satu alat atau model yang digunakan dalam merencanakan pengembangan perangkat lunak dengan pendekatan berbasis objek. UML juga berfungsi sebagai kerangka kerja yang komprehensif untuk menyusun rancangan sistem secara sistematis dan terstruktur. Salah satu aspek utama dari UML adalah pengaturan proses bisnis. Ini berarti bahwa UML membantu dalam mengidentifikasi, mendefinisikan, dan mengatur proses-proses yang terlibat dalam pengembangan perangkat lunak. Mulai dari tahap awal pemetaan kebutuhan hingga tahap implementasi dan pemeliharaan, UML memberikan panduan yang jelas untuk mengelola proses-proses ini secara efisien. Selain itu, UML juga memberikan panduan tentang cara menyusun kelas-kelas dalam bahasa pemrograman tertentu. Ini termasuk dalam hal pengorganisasian struktur kelas-kelas, hubungan antara kelas-kelas tersebut, serta perilaku dan atribut yang terkait. Dengan demikian, UML membantu dalam memastikan konsistensi dan pemahaman yang lebih baik tentang struktur kode dalam pengembangan perangkat lunak. Dalam buku yang ditulis oleh Dennis, Wixom, & Tegarden (2015), juga diungkapkan bahwa *Unified Modeling Language (UML)* memiliki tiga jenis model untuk menggambarkan diagram.



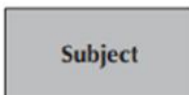

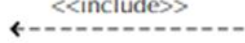
1. *Functional Model*



Diagram model fungsional UML digunakan untuk menunjukkan proses bisnis dan bagaimana sistem informasi berinteraksi dengan lingkungannya. Dalam pendekatan OOAD, dua jenis diagram digunakan untuk menampilkan model fungsional: *use case diagram* dan *activity diagram*.

) *Use Case Diagram*

Gambaran dasar tentang bagaimana suatu sistem berinteraksi dengan lingkungannya diwakili oleh aktor dalam diagram use case. Untuk membuat diagram use case lebih mudah dipahami, disarankan agar jumlah use case dalam satu model tidak lebih dari tiga hingga sembilan. Selain itu, aktor harus lebih dekat dengan use case yang relevan untuk mengurangi kompleksitas. Diagram use case ini menggunakan notasi berikut.

Tabel 2. 1 Notasi Usecase Diagram

Notasi	Deskripsi
 Actor/Role Actor/Aktor	Notasi yang menggambarkan manusia atau sistem yang memperoleh manfaat tetapi berada diluar dari subjek(sistem).
 Use Case Use Case	Notasi yang menggambarkan fungsionalitas yang penting pada sistem.
 Subject Subject	Notasi yang menggambarkan jangkauan dari sebuah subjek (sistem) ataupun suatu proses bisnis.
 Association	Notasi yang menghubungkan antara aktor dengan <i>use case</i> yang berinteraksi
 Include	Notasi yang merepresentasikan penyertaan fungsional dari suatu <i>use case</i> dengan lainnya.

Notasi	Deskripsi
 Extend	Notasi yang merepresentasikan perluasan dari suatu <i>use case</i> untuk menyertakan suatu behavior yang opsional.
 Generalization	Notasi yang merepresentasikan <i>use case</i> atau aktor khusus kepada yang lebih umum (<i>general</i>)

Sumber: (Dennis, Wixom, & Tegarden, 2015)





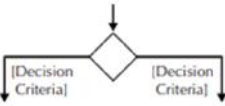
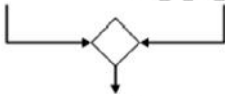

Untuk memberikan informasi yang diperlukan untuk membangun diagram struktural dan perilaku, deskripsi *use case* disertakan dalam diagram *use case* dan mencakup aliran atau urutan peristiwa yang terjadi dalam satu *use case*, mulai dari proses yang memicu hingga kemungkinan alur alternatif yang terjadi. Semua deskripsi ini disusun dalam format yang ringkas agar mempermudah pemahaman pengguna terhadap *use case* tersebut.



) Activity Diagram

Activity Diagram dapat menggambarkan atau merancang berbagai proses dan alur kerja. Ini karena diagram aktivitas dapat merancang alur kerja pada tingkat yang lebih tinggi, yang mencakup berbagai *use case* hingga detailnya. Dalam proses membuat *Activity Diagram*, sangat penting untuk mempertimbangkan kebutuhan sistem yang sedang dibangun dan dokumentasi yang sudah ada, seperti *Use Case* diagram. Menentukan apa yang diperlukan untuk mendukung proses bisnis juga

diperlukan. Di bawah ini adalah notasi yang digunakan dalam *Activity diagram*

Tabel 2. 2 Notasi Activity Diagram

Notasi	Deskripsi
 Activity	Notasi yang menggambarkan manusia atau sistem yang memperoleh manfaat tetapi berada diluar dari subjek (sistem).
 Control Flow	Notasi yang menggambarkan urutan dari eksekusi aktifitas.
 Initial node	Notasi yang menggambarkan awal dari sekumpulan aktifitas (<i>activity diagram</i>).
 Final activity node	Notasi yang menggambarkan berhentinya control flow pada sebuah <i>activity diagram</i>
 Decision node	Notasi yang merepresentasikan kondisi sebuah pengujian agar memastikan alur hanya menyusuri satu jalan.
 Merge Node	Notasi yang digunakan untuk menyatukan alur keputusan yang berbeda dimana berasal dari decision node.
 Fork Node	Notasi yang digunakan untuk memisah suatu behavior menjadi berbeda alur yang berjalan secara bersamaan.

Notasi	Deskripsi
 Join node	Notasi yang digunakan untuk menyatukan alur berbeda yang berjalan secara bersamaan.
 Swimlane	Notasi yang digunakan untuk memecah activity diagram menjadi baris dan kolom dimana bertujuan untuk menetapkan suatu aktivitas individu/objek yang bertanggung jawab dalam menjalankan aktivitas tersebut

Sumber: (Dennis, Wixom, & Tegarden, 2015)

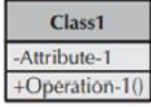
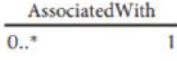

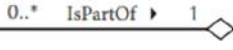

2. Structure Model

Diagram model struktur UML menunjukkan struktur objek yang mendukung proses bisnis dalam sebuah perusahaan. Diagram ini menunjukkan perilaku sistem dalam model fungsional, tetapi penting juga untuk memahami informasi yang digunakan dan dihasilkan oleh sistem, seperti data pengguna dan transaksi. Oleh karena itu, pemodelan struktur menunjukkan objek-objek yang digunakan dan dihasilkan oleh sistem bisnis. Diagram kelas adalah representasi model struktur dalam pendekatan OOAD.

) Class Diagram

Class diagram adalah representasi statis yang menunjukkan kelas dan hubungan antara kelas yang tidak berubah dalam sistem. Class diagram juga menunjukkan perilaku kelas, seperti metode atau operasi. Ini adalah notasi yang digunakan untuk membuat diagram kelas.

Tabel 2. 3 Notasi Class Diagram

Notasi	Deskripsi
 <p>Class</p>	<p>Notasi yang merepresentasikan terkait orang, tempat, atau benda dimana dibutuhkan sistem dalam menangkap dan mengumpulkan informasi.</p>
<p>attribute name /derived attribute name</p> <p>Attribute</p>	<p>Notasi yang merepresentasikan properti yang menggambarkan pernyataan dari suatu objek.</p>
<p>operation name ()</p> <p>Operation</p>	<p>Notasi yang merepresentasikan suatu fungsi atau method yang ada pada suatu class.</p>
 <p>Association</p>	<p>Notasi yang merepresentasikan relasi atau hubungan antar class atau dengan class itu sendiri.</p>
 <p>Generalization</p>	<p>Notasi yang merepresentasikan jenis relasi sejenis terhadap beberapa class.</p>
 <p>Aggregation</p>	<p>Notasi yang merepresentasikan logika dari bagian hubungan antara beberapa class juga dengan class itu sendiri.</p>
 <p>Composition</p>	<p>Notasi yang merepresentasikan secara fisik dari bagian hubungan antara beberapa class juga dengan class itu sendiri.</p>

Sumber: (Dennis, Wixom, & Tegarden, 2015)


3. Behavior Model

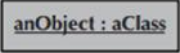




Diagram model perilaku UML dibuat untuk menunjukkan aspek internal yang selalu berubah dari sistem informasi yang mendukung proses bisnis suatu perusahaan. Dalam model fungsional, pembuatan model perilaku mengacu pada diagram *use case* karena fokusnya adalah interaksi antara sistem dan pengguna sistem informasi yang terkait. *interaction* diagram seperti diagram urutan dan *communication* diagram digunakan untuk pemodelan perilaku. *Sequence* Diagram digunakan untuk pemodelan perilaku dalam penelitian ini.

) Sequence Diagram

Sequence diagram ialah representasi visual yang menunjukkan objek-objek yang terlibat dalam suatu skenario penggunaan serta aliran pesan yang terjadi antara mereka selama proses tertentu. Oleh karena itu, diagram ini menjadi model dinamis yang menunjukkan urutan pesan yang dikirim antara objek dalam interaksi yang telah ditetapkan. Selain itu, Anda dapat menyesuaikan pembuatan *sequence* diagram dengan struktur diagram penggunaan yang telah disusun sebelumnya. *Sequence* diagram menggunakan notasi khusus untuk menunjukkan komponen proses.

Tabel 2. 4 Notasi Sequence Diagram

Notasi	Deskripsi
 anActor	Notasi yang menggambarkan manusia atau sistem yang memperoleh manfaat tetapi berada diluar dari subjek (sistem).
Actor/Aktor	

Notasi	Deskripsi
 <p data-bbox="544 447 630 478">Object</p>	<p data-bbox="824 373 1218 457">Notasi yang berpartisipasi dalam mengirim dan menerima pesan.</p>
<p data-bbox="581 520 597 615">⋮</p> <p data-bbox="535 640 636 672">Lifeline</p>	<p data-bbox="824 514 1318 651">Notasi yang menunjukkan jarak waktu kehidupan pada suatu objek selama suatu <i>sequence</i>.</p>
 <p data-bbox="524 814 646 846">Execution</p>	<p data-bbox="824 693 1318 829">Notasi yang menunjukkan saat suatu objek sedang mengirim atau menerima pesan.</p>
 <p data-bbox="532 1113 646 1144">Message</p>	<p data-bbox="824 997 1318 1134">Notasi yang menggambarkan penyampaian informasi dari objek satu dengan lainnya.</p>
 <p data-bbox="483 1381 690 1413">Guard Condition</p>	<p data-bbox="824 1312 1318 1449">Notasi yang merepresentasikan sebuah tes harus bertemu untuk suatu pesan agar dapat terkirim.</p>
 <p data-bbox="544 1669 630 1701">Frame</p>	<p data-bbox="824 1564 1318 1648">Notasi yang menunjukan konteks dari suatu <i>sequence diagram</i>.</p>

Sumber: (Dennis, Wixom, & Tegarden, 2015)

2.1.7. *User Requirement*

Menurut (Sommerville, I, 2003) *user requirement* adalah gambaran dari layanan yang akan diberikan oleh sistem kepada pengguna. Ini mencakup deskripsi yang rinci tentang apa yang diinginkan atau diharapkan oleh pengguna dari sistem yang akan dikembangkan. Selain itu, *user requirement* juga memuat batasan-batasan tertentu yang diberlakukan pada sistem tersebut. Batasan-batasan ini mencakup pembatasan fungsional, non-fungsional, dan aturan-aturan lain yang mengatur cara sistem beroperasi atau berinteraksi dengan pengguna. Dengan demikian, *user requirement* tidak hanya membahas apa yang akan dilakukan oleh sistem, tetapi juga membatasi ruang lingkup dan perilaku sistem itu sendiri. Dengan memahami *user requirement* dengan baik, diharapkan dapat merancang sistem yang memenuhi harapan pengguna dan mematuhi batasan-batasan yang ada, sehingga menghasilkan produk yang lebih bermanfaat dan lebih sesuai dengan kebutuhan.

Menurut (Martono, Sucipto, & Maulana, 2019) Salah satu cara untuk memahami kebutuhan pengembangan sistem atau perangkat lunak adalah metode elisitasi, yang melibatkan proses komunikasi antara pengembang dan pengguna melalui wawancara atau percakapan. Elisitasi dilakukan secara mandiri dalam tiga tahap, yang terdiri dari:

-) Elisitasi tahap 1 merupakan tahapan untuk mengumpulkan semua ide dari pengguna untuk merancang sistem yang akan dibangun.
-) Elisitasi tahap 2 merupakan tahapan hasil-hasil dari tahap pertama, yang disusun menurut metode *Mandatory, Desireable, and Inessential* (MDI). Tujuan dari metode ini adalah untuk membedakan rancangan sistem yang wajib dan esensial (wajib) dari rancangan sistem lainnya yang diinginkan atau tidak penting (diinginkan).

- J) Elisitasi tahap 3 merupakan tahapan menyempurnakan hasil yang telah diperoleh pada tahap sebelumnya, tahap ketiga proses elisitasi melibatkan menghilangkan semua kebutuhan pengguna yang termasuk dalam kategori I (tidak penting) dalam skema MDI. Kemudian, pada tahap ketiga, kebutuhan-kebutuhan tersebut dikelompokkan kembali menggunakan metode Teknik, Operasional, dan Ekonomi (TOE).
- J) Elisitasi tahap final merupakan tahapan yang akan digunakan oleh tim pengembang sebagai panduan untuk mengembangkan sistem atau perangkat lunak, adalah tahap terakhir dari proses elisitasi. Ini terdiri dari kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional mencakup fitur yang diakomodasi oleh sistem itu sendiri, sedangkan kebutuhan non-fungsional mencakup elemen yang berada di luar sistem.

2.1.8. Database

Sistem yang terdiri dari kumpulan data terstruktur yang disimpan dalam sistem komputer disebut *database*. Data ini mudah diakses, dikelola, dan diperbarui. *Database* adalah gudang terpusat yang dapat mengatur dan menyimpan data dari berbagai sumber, seperti data pelanggan, transaksi, produk, dan informasi lainnya yang relevan (Dennis, 2015). Untuk berbagai tujuan, seperti analisis bisnis, pengambilan keputusan, dan pembuatan laporan, orang dapat dengan mudah mengambil, mengubah, dan menganalisis data. Karena strukturnya yang terorganisir, *database* memungkinkan penyimpanan data yang efisien dan memudahkan pengelolaan data dalam skala besar. Ini menjadikan database sebagai bagian penting dari infrastruktur teknologi informasi modern, yang memungkinkan orang untuk menggunakannya untuk berbagai tujuan.

a) **Jenis Database**

) **Operational Database**

Operational Database, juga dikenal sebagai basis data pemrosesan transaksi online (OLTP), adalah jenis basis data yang dirancang untuk membantu dan mengatur kegiatan transaksi harian perusahaan (Dennis, 2015). Basis data operasional berfokus pada pengolahan transaksi dalam jumlah besar secara langsung dan dengan kecepatan yang lebih tinggi.

) **Database Warehouse**

Database Warehouse, juga dikenal sebagai "gudang data" dalam Bahasa Indonesia, adalah tempat penyimpanan data yang mengumpulkan sejumlah besar data, baik yang baru maupun yang lama, dari berbagai sumber dalam suatu organisasi (Dennis, 2015). Sebagai alternatif untuk pemrosesan transaksi, gudang data dirancang untuk keperluan analisis. Ini memungkinkan organisasi untuk menganalisis dan mengekstrak informasi dari data mereka untuk mendukung proses pengambilan keputusan.

) **Distributed Database**

Tidak seperti database terpusat, yang menyimpan semua datanya di satu tempat, database terdistribusi menyebarkan data ke berbagai node dalam jaringan (Dennis, 2015). Hal ini meningkatkan kinerja, skalabilitas, dan ketersediaan.

) **Relational Database**

Sistem manajemen basis data (DBMS) basis data relasional mengolah dan berinteraksi dengan data dengan menggunakan bahasa kueri terstruktur (SQL). SQL sangat penting untuk desain database relasional karena menyimpan data dalam tabel dan memungkinkan pengguna melakukan kueri, memasukkan, memperbarui, dan menghapus data dalam database (Dennis, 2015). Ini adalah alasan mengapa banyak orang memilihnya untuk menyimpan dan mengelola data

terstruktur melalui banyak aplikasi. Salah satu cara yang efektif untuk mengatur data dalam database relasional adalah normalisasi basis data. Ini dilakukan untuk mengurangi jumlah data yang tidak relevan dan duplikat (Dennis, 2015). Untuk mencapai tujuan ini, tabel besar dibagi menjadi tabel yang lebih kecil dan menetapkan hubungan di antara mereka. Normalisasi memiliki tujuan utama untuk mengurangi redundansi data, mencegah pembaruan yang tidak sesuai, dan memastikan integritas data. Dengan menyimpan data dalam format yang terstruktur dan ternormalisasi, duplikasi data berkurang, sehingga penggunaan ruang penyimpanan lebih efisien. Normalisasi, yang mengatur data dalam berbagai tabel, membantu mencegah inkonsistensi data selama pembaruan database dan memastikan bahwa data tetap konsisten. Proses ini dibagi menjadi berbagai bentuk normal, seperti Bentuk Normal Pertama, Kedua, dan Ketiga, dan berdasarkan aturan tertentu yang harus dipatuhi untuk mencapai desain database yang terstruktur dan ternormalisasi dengan baik. Dengan mengikuti aturan normalisasi ini, database dapat dirancang dengan lebih fleksibel, dapat diukur, dan efisien, yang menghasilkan peningkatan kualitas data dan kinerja sistem. Dalam kasus basis data relasional dengan SQL, Basis data relasional merupakan pilihan yang bagus untuk menyimpan dan mengelola data terstruktur dalam berbagai aplikasi karena normalisasi, yang memainkan peran penting dalam memastikan bahwa data disimpan dengan benar, mengurangi redundansi, dan meningkatkan integritas.

) End-User Database

Sistem basis data yang disebut basis data pengguna akhir dirancang untuk digunakan oleh individu atau kelompok dalam organisasi yang langsung berinteraksi dengan basis data untuk

menyelesaikan tugas tertentu atau mendapatkan informasi (Dennis, 2015). Basis data pengguna akhir biasanya lebih kecil dan kurang cakupan daripada basis data tingkat perusahaan, dan biasanya dibuat dan dikelola oleh pengguna non-teknis yang tidak memiliki keahlian yang mendalam.

b) Manfaat Database

Menurut (Dennis, 2015). baik organisasi maupun individu menggunakan database untuk lebih baik mengelola dan mengakses data. Dengan menggunakan database, data dapat disentralisasi, yang memungkinkan pembaruan dari satu sumber sambil memastikan integritas dan konsistensi data. Alat keamanan seperti otentikasi pengguna dan enkripsi mencegah orang yang tidak berhak mengakses informasi sensitif. Batasan integritas data memastikan bahwa data akurat dan mencegah duplikasi. Pada basis data, kriteria ACID (atomicity, consistency, isolation, and durability) memastikan bahwa transaksi selalu andal dan konsisten, bahkan ketika ada kegagalan. Dengan memperluas basis data, Anda dapat mengatasi beban pengguna yang meningkat dan volume data yang lebih besar tanpa mengurangi kinerja. Akses data yang cepat dan efisien dimungkinkan oleh alat dan antarmuka seperti SQL. Fitur pencadangan dan pemulihan memastikan bahwa data tetap dapat diakses bahkan dalam kegagalan atau bencana. Mengekstrak wawasan dan pengambilan keputusan dibantu oleh alat analisis dan pelaporan. Database meningkatkan efisiensi dan produktivitas dengan membuat proses manajemen data lebih mudah dan mengotomatiskan tugas berulang. database juga memungkinkan orang bekerja sama dan menyebarkan data antara pengguna atau departemen. Secara keseluruhan,

mengorganisasi, mempertahankan, dan memanfaatkan data sangat penting untuk mendukung inovasi, pengambilan keputusan, dan operasi bisnis.

2.1.9. *Web*

Istilah "*web*" umumnya merujuk pada *World Wide Web*, yaitu sistem halaman *web* dan sumber daya yang saling terkait yang diakses melalui internet menggunakan browser *web* (Dennis, 2015). *World Wide Web*, yang sering disebut sebagai *web*, ditemukan oleh Tim Berners-Lee pada tahun 1989 dan sejak itu menjadi bagian fundamental dari *internet*, memungkinkan pengguna untuk mengakses dan berbagi informasi di seluruh dunia. Secara keseluruhan, *web* telah mengubah cara komunikasi, berbagi informasi, *e-commerce*, hiburan, dan kolaborasi di tingkat global, menjadikannya *platform* penting untuk mengakses dan berinteraksi dengan konten dan layanan digital melalui internet. Pada penerapannya *web* dibagi menjadi dua sisi yaitu dari sisi *server* (*Server Side*) dan sisi klien (*Client Side*):

a) *Server Side*

Menurut (Dennis,2015). Dalam konteks *web*, *server side* mengacu pada komponen dari aplikasi *web* atau situs yang beroperasi di *server* alih-alih di browser pengguna. Teknologi ini bertugas untuk menangani permintaan, menjalankan kode, dan membuat konten dinamis yang kemudian dikirim ke browser pengguna untuk ditampilkan. Berbagai macam alat yang dapat digunakan dalam mengoperasikan sisi server ini seperti: *PHP* (*Hypertext Preprocessor*), *Perl*, *Phyton*, *XAMPP* dan sebagainya. Dengan mengelola pemrosesan yang kompleks dan manipulasi data di sisi server, aplikasi *web* dapat menyajikan konten dinamis dan interaktif kepada pengguna sambil memastikan keamanan, skalabilitas, dan kinerja tetap terjaga.

Teknologi sisi *server* memiliki peran krusial dalam menentukan fungsionalitas dan perilaku aplikasi *web modern*..

b) Client Side

Menurut (Dennis, 2015) dalam konteks *web*, *client side* mengacu pada bagian dari aplikasi *web* atau situs *web* yang berjalan di *browser* pengguna (klien). Teknologi *client side* bertanggung jawab untuk menampilkan antarmuka pengguna, mengelola interaksi pengguna, dan menjalankan tugas-tugas langsung di dalam browser. Berbagai macam alat yang dapat digunakan dalam mengoperasikan sisi klien ini seperti: *HTML (HyperText Markup Language)*, *CSS (Cascading Style Sheet)*, *JavaScript* dan sebagainya. Dengan memanfaatkan teknologi *client side*, pengembang *web* dapat membuat aplikasi *web* yang interaktif, responsif, dan menarik secara visual, yang meningkatkan keterlibatan pengguna dan memberikan pengalaman pengguna yang mulus langsung di dalam *browser*. Teknologi *client side* memegang peran penting dalam membentuk fungsionalitas dan tampilan *front-end* dari aplikasi *web modern*.

2.1.10. Testing

Menurut (Dennis, 2015) *Testing* adalah prosedur yang digunakan dalam pengembangan perangkat lunak untuk memeriksa aplikasi atau sistem perangkat lunak untuk menemukan kesalahan atau bug dan memastikan bahwa perangkat lunak memenuhi persyaratan dan standar kualitas yang ditetapkan. Dalam siklus pengembangan, tahap ini sangat penting untuk memverifikasi fungsionalitas, kinerja, keamanan, dan kegunaan perangkat lunak sebelum dirilis ke pengguna. Pengujian membantu menemukan masalah awal dalam proses pengembangan, meningkatkan kualitas perangkat lunak, meningkatkan kepuasan pengguna, dan mengurangi kemungkinan kegagalan perangkat lunak di produksi.

Produk perangkat lunak yang andal dan berkualitas tinggi membutuhkan praktik pengujian yang baik.

a) *Black Box Testing*

Menurut (Devianty, D., Ibrahim, R. N., & Wahyudi, H, 2021) Black-box Testing adalah metode pengujian perangkat lunak yang memprioritaskan pemeriksaan terhadap fungsi-fungsi yang dijanjikan dan hanya memperhatikan detail-detail dari domain informasi yang relevan. Sehingga, pemeriksaan Black-box difokuskan pada fitur-fitur khusus yang diidentifikasi dalam spesifikasi perangkat lunak. Ini menunjukkan bahwa pengujian dilakukan dengan memasukkan input yang sesuai dengan spesifikasi dan memeriksa apakah outputnya memenuhi harapan. Pada saat yang sama, tidak diperlukan pengetahuan mendalam tentang struktur kode program atau implementasi internal perangkat lunak; pengujian dapat dilakukan dari sudut pandang eksternal tanpa memiliki akses ke detail internal perangkat lunak, yang memungkinkan untuk menguji perangkat lunak tanpa mengetahui detailnya. Metode ini memungkinkan untuk menemukan kesalahan atau cacat dalam fungsi-fungsi yang dimaksud tanpa bergantung pada cara kode program diimplementasikan.

b) *White Box Testing*

Menurut (Ichsanudin, M. N., Yusuf, M., & Suraya, S., 2022). *White Box Testing* adalah teknik pengujian perangkat lunak yang menguji struktur internal dan kode program untuk menemukan kesalahan atau cacat. Teknik ini menggunakan teknik seperti pengujian jalur, kondisi, loop, dan alur data untuk memastikan bahwa semua jalur eksekusi dalam kode diuji dan bahwa logika program beroperasi dengan benar. Analisis kode adalah langkah pertama dalam proses, kemudian perancangan dan eksekusi kasus uji dan analisis hasil untuk memperbaiki

kesalahan. *White Box Testing* sangat bermanfaat karena dapat mendeteksi kesalahan lebih awal dan meningkatkan stabilitas dan kualitas perangkat lunak. Karena memerlukan pemahaman mendalam tentang kode dan kemungkinan tidak menemukan masalah pada tingkat sistem atau integrasi, metode ini mahal dan memakan waktu. Namun, *White Box Testing* masih menjadi teknik penting dalam pengembangan perangkat lunak untuk memastikan bahwa aplikasi akan berfungsi dan andal sesuai spesifikasi.

2.1.11. Aplikasi

Aplikasi merupakan sebuah alat yang dirancang untuk melancarkan dan mempercepat proses kerja, tidak menjadi beban bagi individu yang menggunakannya (Syabania & Rosmawarni, 2021). Sehingga Dengan adanya aplikasi, tugas-tugas yang sebelumnya mungkin memakan waktu dan tenaga yang besar dapat diselesaikan dengan lebih optimal. Hal ini memungkinkan pengguna untuk fokus pada hal-hal yang lebih penting dalam pekerjaan mereka tanpa terbebani oleh proses yang lambat atau rumit. Selain itu, aplikasi juga dapat membantu meningkatkan produktivitas dan kualitas pekerjaan dengan menyediakan berbagai fitur dan fungsi yang mendukung berbagai kebutuhan pengguna. Dengan demikian, penggunaan aplikasi menjadi sebuah solusi yang menguntungkan bagi individu maupun organisasi dalam menjalankan berbagai aktivitas kerja.

a) Fitur Aplikasi

Fitur aplikasi adalah komponen kunci dari karakteristik sebuah aplikasi yang mencerminkan kegunaan dan keunggulan fungsionalnya. Mereka juga merupakan aspek yang unik dan spesifik yang ditambahkan ke dalam aplikasi, membedakannya dari aplikasi-aplikasi sejenis dari pesaingnya (Putri, Rosa & Sabathini, 2022). Sehingga fitur-fitur ini tidak hanya menjadi faktor penting dalam menarik minat konsumen, tetapi juga

menjadi dasar pertimbangan utama bagi mereka saat memilih aplikasi untuk digunakan.

b) Fungsi Aplikasi

Perangkat lunak yang disebut aplikasi dapat digunakan untuk memenuhi kebutuhan pengguna atau bisnis dengan menyediakan berbagai fitur dan fungsi (Dennis, 2015). Salah satu fungsi utama aplikasi adalah pengelolaan data, yang memungkinkan penyimpanan data yang aman, pencarian cepat, dan pengaturan akses yang aman. Aplikasi juga membantu strategi bisnis, meningkatkan efisiensi operasional, dan memenuhi kebutuhan pengguna dengan fitur seperti integrasi sistem, kemampuan berbagi informasi, dan dukungan transaksi. Aplikasi juga sangat penting untuk otomatisasi proses bisnis, yang menghemat waktu dan biaya dengan mengotomatisasi alur kerja dan pemrosesan transaksi. Secara keseluruhan, aplikasi menjadi sangat penting untuk memenuhi kebutuhan pengguna di era digital dan memungkinkan bisnis berkembang.

2.1.12. Visualisasi Data

Visualisasi data merupakan alat untuk mempermudah pemahaman dan analisis data dengan maksud membantu organisasi, perusahaan, atau individu dalam memahami makna dan hubungan data (Nugraha, 2022). Sehingga, dengan mengubah data-data yang ada dari format tabel ke format visual, memungkinkan untuk pengamatan dan pemahaman yang lebih baik atas informasi yang terkandung dalam data tersebut. Dengan visualisasi, kompleksitas data dapat disederhanakan sehingga pola, tren, dan anomali dapat lebih mudah teridentifikasi, membantu pengambilan keputusan yang lebih optimal.

2.1.13. Rekapitulasi Data

Rekapitulasi adalah suatu kegiatan meringkaskan data dari kertas kerja atau isi, baik itu pada akhir laporan, sehingga data tersebut menjadi lebih bermanfaat dalam hal bentuk, susunan, sifat, atau isinya (Thoriq, 2023). Proses ini dilakukan dengan bantuan tenaga tangan atau dengan menggunakan suatu peralatan, serta mengikuti rangkaian langkah, rumus, atau pola tertentu. Sehingga, hasil rekapitulasi memberikan gambaran yang lebih jelas dan terstruktur mengenai informasi yang terdapat dalam dokumen kerja atau laporan tersebut.

2.1.14. Advokasi

Pengertian advokasi adalah sebuah kebijakan atau keputusan yang dibuat oleh pemerintah atau lembaga lainnya yang berdampak baik pada masyarakat secara keseluruhan maupun individu-individu di dalamnya (Lestari, Subakti & Afandi, 2023). Kebijakan jenis ini tidak hanya dimaksudkan untuk memenuhi kebutuhan, keamanan, dan kesejahteraan umum, tetapi juga untuk mendukung kemajuan dan perkembangan sosial. Mereka juga dapat dirancang untuk memenuhi kebutuhan atau kepentingan khusus sebagian anggota masyarakat yang membutuhkan perlindungan atau dukungan khusus dari pemerintah atau lembaga lainnya. Dengan demikian, kebijakan ini bertujuan untuk mencapai keseimbangan antara kepentingan umum.

2.2. Tinjauan Studi

Dalam bagian ini, merupakan ringkasan penelitian sebelumnya yang berkaitan dengan subjek dan materi penelitian saat ini. Penelitian yang dipilih memiliki hubungan dan terkait dengan subjek penelitian ini, dan ini mencakup:

- 1) Tugas Akhir penelitian yang dilakukan oleh Adinda Putri Narewari yang berjudul “RANCANG BANGUN APLIKASI PENGADUAN MAHASISWA DI UNIVERSITAS PEMBANGUNAN JAYA (STUDI KASUS: MODUL PENGADUAN)” dan diterbitkan oleh Universitas Pembangunan Jaya. pada *website repository* pada tahun 2024 (<https://eprints.upj.ac.id/id/eprint/7764/>) mengulas tentang dampak positif yang diberikan oleh kemajuan teknologi, khususnya internet, pada berbagai sektor kehidupan, termasuk sektor pendidikan di Universitas Pembangunan Jaya (UPJ). Di UPJ, teknologi internet telah dimanfaatkan untuk mendukung kegiatan mahasiswa yang diorganisir oleh Badan Eksekutif Mahasiswa (BEM). Namun, meskipun ada kemajuan teknologi, proses pengaduan mahasiswa di UPJ masih menghadapi sejumlah kendala. Prosedur pengaduan yang tidak baku, kendali yang tidak terstruktur, dan kurangnya keamanan data pengaduan merupakan beberapa masalah utama yang dihadapi. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi pengaduan di UPJ dengan fokus utama pada modul pengajuan aduan, persetujuan aduan, dan advokasi mahasiswa. Identifikasi masalah dalam penelitian ini mencakup kesulitan yang dihadapi mahasiswa dalam melakukan pengaduan, validasi aduan yang diajukan, histori aduan untuk melacak status dan progres pengaduan, serta pengelolaan data mahasiswa yang lebih efisien dan aman. Sistem informasi yang dirancang juga bertujuan untuk meningkatkan upaya advokasi dalam menjaga hak dan kepentingan mahasiswa dengan cara yang lebih terstruktur dan baku. Secara keseluruhan, tugas akhir ini menggambarkan upaya untuk memanfaatkan kemajuan teknologi guna mengatasi masalah yang ada dalam proses pengaduan mahasiswa di UPJ. Sistem informasi yang dirancang

diharapkan dapat meningkatkan efisiensi, keamanan, dan efektivitas dalam menangani pengaduan mahasiswa, serta memperkuat advokasi terhadap hak-hak mahasiswa.

- 2) Jurnal hasil penelitian yang dilakukan oleh Iffat Dwi Ananto yang berjudul “RANCANG BANGUN APLIKASI ADVOKASI MAHASISWA PADA UNIVERSITAS PEMBANGUNAN JAYA (STUDI KASUS: MODUL PENYELESAIAN)” dan diterbitkan oleh Universitas Pembangunan Jaya. Pada website repository pada tahun 2024 (<https://eprints.upi.ac.id/id/eprint/7738/>) berisi tentang pentingnya advokasi, yang umumnya dikenal dalam konteks hukum sebagai tindakan pembelaan, dukungan, dan rekomendasi melalui komunikasi persuasif. Penelitian ini menyoroti bahwa proses advokasi pengaduan mahasiswa di Universitas Pembangunan Jaya (UPJ) belum optimal. Mahasiswa sering mengalami kesulitan dalam menghubungi pihak terkait dan memahami prosedur pengaduan yang rumit. Selain itu, keterbatasan akses informasi mengenai cara mengajukan pengaduan dan status pengaduan juga mempengaruhi penanganan masalah. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan menghadirkan platform advokasi yang dilengkapi dengan fitur-fitur penyelesaian, status pengaduan, dan informasi regulasi. Untuk mengembangkan platform ini, metode yang digunakan adalah Rapid Application Development (RAD) yang didukung oleh Unified Modeling Language (UML). Dengan adanya platform ini, diharapkan proses advokasi pengaduan mahasiswa di UPJ dapat menjadi lebih efisien, transparan, dan mudah diakses, sehingga dapat memberikan dukungan yang lebih baik kepada mahasiswa dalam menyelesaikan masalah yang mereka hadapi.

3) Jurnal hasil penelitian yang dilakukan oleh Alvin Ramdhani dan Ahmad Muhammad Thantawi yang berjudul “RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN BERBASIS WEB DAN DASHBOARD VISUALISASI DATA UNTUK MONITORING MINAT BACA PADA SMK NEGERI 21” dan diterbitkan oleh IKRA-ITH Informatika: Jurnal Komputer dan Informatika, pada website repository pada tahun 2024(<https://journals.upi-yai.ac.id/index.php/ikraith-informatika/article/download/3036/2206>) berisi tentang pemecahan masalah terkait kurangnya penggambaran yang jelas mengenai permasalahan yang spesifik yang dihadapi oleh perpustakaan SMK Negeri 21 Jakarta sebelum diterapkannya sistem informasi berbasis web. Salah satu permasalahan yang dihadapi adalah proses pengolahan data yang masih mengandalkan metode manual, yang seringkali rentan terhadap kesalahan dan memakan waktu. Selain itu, terdapat kendala dalam mengawasi tindakan manual siswa dan guru saat mengakses koleksi buku, yang dapat memperlambat proses peminjaman dan pengembalian. Tujuan utama dari sistem informasi ini adalah untuk meningkatkan efisiensi pengolahan data dan memberikan kemudahan akses bagi karyawan dan pengguna perpustakaan SMK Negeri 21 Jakarta. Dengan memanfaatkan solusi berbasis web, sistem informasi perpustakaan berbasis web ini bertujuan untuk memecahkan masalah mengawasi tindakan manual yang dilakukan oleh siswa dan guru saat mengakses koleksi buku. Sistem ini dibangun menggunakan bahasa PHP dan *database* MySQL dengan dukungan framework Codeigniter. Struktur ini dirancang untuk memenuhi kebutuhan dan infrastruktur SMK Negeri 21 Jakarta. Salah satu fokus utama dalam pengembangan sistem ini adalah pembuatan dashboard visualisasi data yang diharapkan dapat

membantu pengelola perpustakaan memantau minat baca siswa dan memungkinkan penyesuaian pelayanan perpustakaan dan koleksi buku secara lebih tepat. Oleh karena itu, diharapkan bahwa sistem yang diusulkan akan membuat pengelolaan data menjadi lebih mudah bagi karyawan perpustakaan dan meningkatkan akses pengguna dan staf perpustakaan terhadap informasi. Selain itu, diharapkan bahwa dashboard visualisasi akan membuat siswa lebih tertarik untuk membaca melalui pengawasan yang lebih tepat guna.

- 4) Jurnal hasil penelitian yang dilakukan oleh M Yassir Saputra Jamia yang berjudul “RANCANG BANGUN SISTEM INFORMASI REKAPITULASI DATA DOSEN BERBASIS WEB DI UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU” dan diterbitkan oleh Universitas Islam Negeri Sultan Syarif Kasim Riau. pada *website repository* pada tahun 2021(<https://repository.uin-suska.ac.id/46799/>) berisi tentang bagaimana merancang dan membangun sistem informasi rekapitulasi data dosen dengan mengimplementasikan berbasis web. Sistem informasi ini dirancang untuk membuat proses rekapitulasi dan penyajian data lebih mudah. Tujuan sistem ini adalah untuk membantu dosen di Bidang Administrasi dan Tata Usaha Fakultas mengecek kelengkapan informasi dan SKS. alasan Rekapitulasi data adalah proses penting dalam pengolahan informasi karena dapat membuat data lebih mudah diakses dan diakses dengan cepat. Saat data yang telah dikumpulkan disajikan dalam format yang tepat dan benar, itu akan bermanfaat. Akibatnya, diperlukan sebuah sistem yang dapat membuat rekapitulasi data yang akurat dan terstruktur. Diharapkan proses rekapitulasi data akan menjadi lebih akurat setelah sistem informasi berbasis web ini diterapkan. Staf administrasi dan dosen akan dapat dengan mudah mendapatkan

dan menggunakan informasi yang diperlukan untuk menyelesaikan tugas mereka. Selain itu, diharapkan bahwa penerapan sistem informasi ini akan meningkatkan daya guna dalam pengelolaan data karena informasi yang diolah akan tersusun secara sistematis, yang memudahkan penggunaan data untuk berbagai kebutuhan. Selain itu, karena semua informasi akan didokumentasikan dengan baik, sistem ini juga dapat membantu mencegah kehilangan data. Oleh karena itu, diharapkan bahwa sistem informasi ini akan membantu meningkatkan kinerja dan efisiensi Fakultas Sains dan Teknologi UIN SUSKA RIAU dalam pengelolaan data dan informasi.

- 5) Jurnal hasil penelitian yang dilakukan oleh Harsih Rianto dan Amrin yang berjudul “RANCANG BANGUN SISTEM INFORMASI INVENTORY MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT” dan diterbitkan oleh INSANtek–Jurnal Inovasi dan Sains Teknik Elektro. Pada *website repository* pada tahun 2023 (<http://eprints.bsi.ac.id/index.php/insantek/article/view/1942/1214>) berisi tentang permasalahan Sistem *inventory* yang masih sederhana dan manual menghadirkan banyak masalah bagi perusahaan retail. Kondisi ini menyebabkan masalah penyimpanan dokumen dan meningkatkan kemungkinan kesalahan selama proses pengolahan data. Selain itu, pengelolaan *inventory* yang tidak efektif menjadi masalah, yang dapat berdampak pada layanan pelanggan. Selain itu, sistem yang ada tidak memenuhi kebutuhan bisnis untuk bersaing di pasar. Kepuasan pelanggan dan kinerja bisnis perusahaan secara keseluruhan dapat dipengaruhi oleh ketidakakuratan dalam pengelolaan *inventory*. Perusahaan retail harus mempertimbangkan solusi untuk meningkatkan efisiensi,

akurasi, dan fleksibilitas pengelolaan inventory saat menghadapi masalah ini. Mereka juga harus memastikan bahwa sistem yang digunakan dapat memenuhi tuntutan pasar dan kebutuhan pelanggan. Oleh sebab itu, penulis membuat rancang bangun sistem informasi inventory menggunakan metode Rapid Application Development (RAD) yang diharapkan dapat meningkatkan efisiensi pengelolaan persediaan barang. Didalamnya, terdapat tahapan-tahapan dalam pengembangan sistem informasi, seperti analisis kebutuhan sistem, desain pengguna, konstruksi sistem, pengujian sistem, serta implementasi dan pemeliharaan sistem. Selain itu, terdapat juga hasil pengujian *Black Box Testing* pada sistem.