

BAB II TINJAUAN PUSTAKA

1.1 Teori Umum

2.1.1 Pengertian Pengembangan

Menurut Rahmat Mulyawan (2015), Pengembangan perangkat lunak didefinisikan sebagai rangkaian kegiatan terstruktur dengan tujuan untuk menciptakan perangkat lunak yang bermutu dan memenuhi kebutuhan pengguna. Proses ini meliputi analisis kebutuhan, perancangan sistem, pelaksanaan, pengujian, dan pemeliharaan. Mulyawan juga menekankan pentingnya menggunakan metodologi pengembangan yang tepat untuk meningkatkan efisiensi proses tersebut.

Menurut Rini Anggraeni (2018), Pengembangan aplikasi didefinisikan sebagai proses yang meliputi perancangan, pengembangan, dan pengujian aplikasi untuk memverifikasi kesesuaian dengan kebutuhan pengguna. Anggraeni menyoroti pentingnya iterasi dalam proses pengembangan untuk mengakomodasi perubahan dan umpan balik yang diterima dari pengguna.

2.1.2 Pengertian Sistem

Sistem, menurut Mulyadi (2016), adalah kumpulan dari bagian-bagian yang bekerja sama secara harmonis untuk mencapai tujuan yang telah ditetapkan. Komponen-komponen sistem bekerja sama untuk mencapai tujuan yang telah ditetapkan karena mereka saling berhubungan.

Sutabri Tata (2012) mendefinisikan sistem sebagai sebuah struktur yang terdiri dari bagian-bagian yang saling berhubungan yang bekerja sama untuk mencapai tujuan tertentu. Sistem adalah segala sesuatu yang terdiri dari komponen-komponen yang bekerja sama untuk mencapai tujuan tertentu.

Menurut Azhar Susanto (2013), Sistem adalah sebuah entitas yang terbentuk dari berbagai elemen yang saling berinteraksi secara terstruktur

untuk mencapai tujuan spesifik. Elemen-elemen ini meliputi individu, data, proses, teknologi, dan kontrol yang bekerja bersama untuk menghasilkan data yang berguna dalam proses pengambilan keputusan.

2.1.2 Pengertian Informasi

Informasi adalah konsep yang merujuk kepada data yang telah diorganisir atau diolah sedemikian rupa sehingga memiliki nilai dan makna bagi penerima. Dalam konteks ini, informasi tidak hanya sekedar kumpulan data mentah, tetapi juga melibatkan proses pengolahan dan interpretasi untuk menghasilkan pemahaman yang berguna. Teori informasi, yang diperkenalkan oleh Claude Shannon dan Warren Weaver, mengajukan model matematis untuk mengukur jumlah informasi dalam suatu pesan serta mengevaluasi efisiensi komunikasi. Teori ini menyatakan bahwa informasi dapat diukur dalam bentuk bit, yaitu unit terkecil informasi yang merepresentasikan suatu kejadian atau pesan. Informasi juga memiliki sifat-sifat seperti ketepatan, relevansi, kebenaran, dan keterpercayaan, yang menentukan nilai dan kegunaannya dalam konteks tertentu. Dalam era *digital*, pertumbuhan teknologi informasi dan komunikasi telah mengubah cara informasi diproduksi, disimpan, diproses, dan disebar. Dengan pemahaman yang kuat tentang teori informasi, seseorang dapat memahami proses komunikasi, pengambilan keputusan, dan pertukaran pengetahuan dalam berbagai konteks, mulai dari ilmu komputer hingga ilmu sosial.

2.1.3 Pengertian Sistem Informasi

Dalam bukunya "Sistem Informasi Manajemen," Jogiyanto mendefinisikan sistem informasi sebagai kombinasi terstruktur dari orang-orang, perangkat, perangkat lunak, jaringan komunikasi, dan sumber daya data yang bekerja bersama untuk mengumpulkan, memproses, dan mendistribusikan informasi di dalam perusahaan. Sistem informasi terdiri dari berbagai komponen utama, seperti:

- a. Orang: Pengguna sistem informasi, termasuk manajer, analis, pengembang aplikasi, dan operator. Perangkat keras (hardware): Meliputi komputer dan perangkat terkait yang digunakan untuk menjalankan operasional sistem informasi.
- b. Perangkat lunak (software): Aplikasi dan sistem operasi yang digunakan untuk memproses data.
- c. Jaringan komunikasi: Media dan perangkat yang memungkinkan pengiriman dan penerimaan data.
- d. Sumber daya data: Data yang dikumpulkan, disimpan, dikelola, dan diproses oleh sistem informasi.

2.1.4 Pengertian Aplikasi

Menurut A. Nurhayati, (2017). Aplikasi adalah program komputer yang dibuat untuk tujuan spesifik, seperti membantu pengguna menyelesaikan tugas, mengakses informasi, atau memberikan hiburan. Konsep aplikasi didasarkan pada gagasan bahwa teknologi komputer dapat digunakan untuk memenuhi berbagai kebutuhan pengguna dalam konteks pekerjaan, pendidikan, hiburan, atau komunikasi. Fungsionalitas sebuah aplikasi mencakup berbagai fitur dan kemampuan yang memungkinkan pengguna mencapai tujuannya, yang ditentukan oleh kebutuhan pengguna dan tujuan aplikasi tersebut. Antarmuka pengguna, sebagai penghubung antara pengguna dan aplikasi, berperan penting dalam memastikan pengalaman pengguna yang baik dan efisien. Dengan antarmuka pengguna yang sederhana dan dapat dimengerti, pengguna dapat berhubungan dengan aplikasi secara nyaman dan efektif. Oleh karena itu, aplikasi tidak hanya sebagai alat teknologi melainkan juga sebagai hal yang memfasilitasi interaksi manusia dengan teknologi dalam berbagai aspek kehidupan sehari-hari.

2.1.5 Pengertian Advokasi

Menurut Rizal, M. C. Advokasi adalah proses atau tindakan yang bertujuan untuk mendukung, memperjuangkan, atau memperjuangkan suatu isu, tujuan, atau kepentingan tertentu. Landasan teoritis advokasi melibatkan konsep-konsep dan prinsip-prinsip yang diterapkan untuk merencanakan, melaksanakan, dan mengevaluasi upaya advokasi. Salah satu landasan teoritis utama dari advokasi adalah teori perubahan sosial, yang mengajukan bahwa perubahan dalam masyarakat dapat dicapai melalui aksi kolektif yang terorganisir. Teori ini menyoroti pentingnya mobilisasi massa, pemilihan strategi yang tepat, dan pembangunan kekuatan politik untuk mencapai tujuan advokasi. Selain itu, teori komunikasi juga memiliki peran penting dalam advokasi, dengan menekankan pentingnya pesan yang jelas, strategi komunikasi yang efektif, dan penggunaan media dan narasi untuk mempengaruhi opini publik dan pembuat kebijakan. Advokasi juga diperkaya oleh konsep-konsep seperti pemahaman kepentingan (interest), keadilan sosial, hak asasi manusia, dan kekuasaan politik. Dengan menggunakan landasan teoritis yang kuat, praktisi advokasi dapat merencanakan dan melaksanakan strategi yang efektif untuk mengubah kebijakan, praktek, atau sikap masyarakat terkait isu-isu yang diadvokasi.

2.1.6 Pengertian *Helpdesk*

Menurut Manulu, W. U. S. *helpdesk* adalah sebuah layanan atau departemen dalam sebuah organisasi yang bertanggung jawab untuk memberikan dukungan teknis dan bantuan kepada pengguna akhir atau pelanggan terkait dengan penggunaan produk atau layanan yang disediakan oleh organisasi tersebut. Fungsi utama dari *Helpdesk* adalah untuk memberikan solusi atas masalah, pertanyaan, atau permintaan bantuan teknis yang diajukan oleh pengguna, baik melalui telepon, email, atau sistem tiket *online*. *Helpdesk* sering kali menjadi titik kontak pertama bagi pengguna yang mengalami masalah atau memiliki pertanyaan terkait dengan produk atau layanan tertentu. Tim *Helpdesk* biasanya terdiri dari staf yang terlatih dan memiliki pengetahuan yang cukup luas tentang produk atau layanan yang mereka dukung, sehingga mereka dapat memberikan bantuan yang efektif dan solusi yang memuaskan kepada pengguna. Fungsi-fungsi umum dari *Helpdesk* meliputi penerimaan dan penanganan permintaan dukungan teknis, pelaporan masalah ke departemen terkait, pemantauan dan pelacakan status permintaan dukungan, serta dokumentasi dan pelaporan kinerja *Helpdesk*. Selain itu, *Helpdesk* juga dapat memberikan pelatihan kepada pengguna terkait dengan penggunaan produk atau layanan tertentu agar mereka dapat mengoptimalkan penggunaannya secara mandiri.

2.2 Teori Khusus

2.2.1 Siklus Hidup Pengembangan Sistem (*System Development Live Cycle*)

Menurut Roger S. Pressman, SDLC adalah suatu kerangka kerja yang mendefinisikan serangkaian tahapan untuk memproduksi perangkat lunak yang memiliki kualitas tinggi dan sesuai dengan kebutuhan pengguna. Pressman menjelaskan tahapan-tahapan dalam SDLC meliputi:

1. **Perencanaan (Planning)**

Dalam tahap perencanaan SDLC, langkah pertama adalah menetapkan tujuan proyek serta ruang lingkupnya. Selanjutnya, risiko-risiko yang mungkin timbul diidentifikasi, dan strategi mitigasi dikembangkan untuk menghadapinya. Terakhir, dalam tahap ini, jadwal proyek, anggaran, serta alokasi sumber daya disusun agar proyek dapat berfungsi secara optimal dan memenuhi semua kriteria dan batasan yang telah ditentukan.

2. **Analisis Kebutuhan (*Requirements Analysis*)**

Pada tahap analisis kebutuhan dalam SDLC, langkah awal adalah mengumpulkan dan menganalisis kebutuhan dari pengguna dan sistem itu sendiri. Informasi yang terkumpul kemudian disusun dalam bentuk dokumen spesifikasi kebutuhan, yang akan menjadi panduan untuk tahap-tahap pengembangan berikutnya. Dokumen ini menjadi fondasi yang sangat penting dalam proses selanjutnya untuk menjamin bahwa aplikasi yang dibangun mampu memenuhi kriteria dan ekspektasi pengguna serta mencapai tujuan bisnis yang telah ditetapkan.

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

Dalam tahap desain sistem SDLC, tim pengembangan merancang arsitektur sistem dan detail teknis, termasuk desain antarmuka pengguna, basis data, dan komponen perangkat lunak lainnya. Mereka juga menyiapkan rencana implementasi dan strategi pengujian untuk memastikan kualitas dan keandalan perangkat lunak sebelum diluncurkan secara resmi. Rencana ini penting untuk memandu proses pengembangan selanjutnya dan memastikan produk memenuhi standar kualitas yang diinginkan.

4. **Pengembangan (*Development*)**

Dalam tahap pengembangan SDLC, tim pengembangan menulis kode sumber sesuai dengan rencana desain yang telah ditetapkan dan menggunakan metodologi serta teknik pemrograman yang sesuai untuk membangun perangkat lunak. Tahap ini memungkinkan implementasi konsep yang telah dirancang menjadi sebuah produk perangkat lunak yang fungsional.

5. **Pengujian (*Testing*)**

Dalam tahap pengujian SDLC, tim melakukan pengujian unit, integrasi, dan sistem untuk memastikan kualitas dan keandalan perangkat lunak. Setelah itu, dilakukan validasi dan verifikasi perangkat lunak untuk memastikan bahwa hasil pengujian memenuhi spesifikasi kebutuhan yang telah ditetapkan. Tahap ini penting untuk memastikan bahwa perangkat lunak siap untuk digunakan dan beroperasi sesuai dengan yang diharapkan.

6. **Implementasi (*Implementation*)**

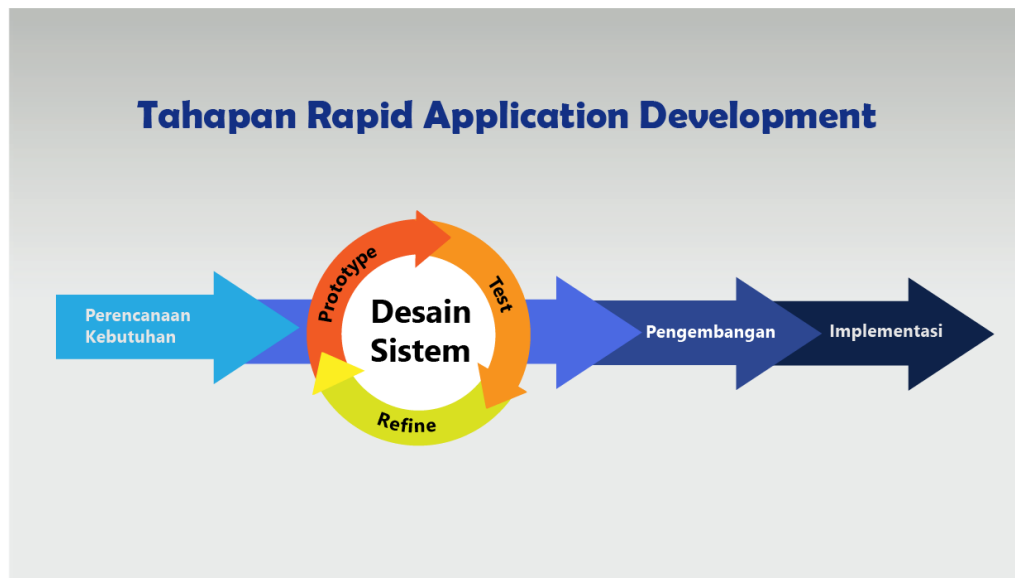
Dalam tahap implementasi SDLC, perangkat lunak diinstal dan dikonfigurasi di lingkungan produksi, sementara pengguna akhir dilatih dan diberikan dukungan teknis awal. Tahap ini penting untuk memastikan bahwa perangkat lunak siap digunakan oleh pengguna akhir dan bahwa mereka memiliki pengetahuan dan dukungan yang cukup untuk menggunakannya secara efektif.

7. **Pemeliharaan (*Maintenance*)**

Dalam tahap pemeliharaan SDLC, perangkat lunak dipelihara untuk memastikan kinerjanya tetap optimal setelah implementasi. Ini melibatkan pembaruan, perbaikan bug, dan penambahan fitur sesuai kebutuhan pengguna. Tahap ini penting untuk memantau agar perangkat lunak tetap beroperasi secara efektif dan memenuhi tuntutan yang berkembang dari pengguna.

2.2.2 *Rapid Application Development (RAD)*

Seperti yang dinyatakan oleh Kevin Pradana (2022). Pengembangan perangkat lunak menggunakan Rapid Application Development (RAD) menekankan pada pembuatan prototipe yang cepat dan berulang. Dengan melibatkan pengguna secara aktif dalam proses pengembangan, metode Rapid Application Development (RAD) berusaha untuk mempercepat langkah-langkah yang terlibat dalam pengembangan perangkat lunak.



Gambar 2. 1 Tahapan RAD

Sumber : Nurman dan Kusuma (2021)

Tahapan dalam *Rapid Application Development* (RAD) dapat bervariasi sesuai dengan metodologi yang diterapkan dan kebutuhan spesifik proyek. Namun, secara umum, ada beberapa langkah yang umumnya ada dalam pendekatan RAD:

A. Perencanaan Kebutuhan

Identifikasi kebutuhan dan persyaratan proyek secara rinci. Ini melibatkan pengumpulan informasi dari pemangku kepentingan (stakeholders) dan pemahaman mendalam tentang tujuan bisnis.

B. User Design

Fokus pada partisipasi aktif pengguna. Dalam tahap ini, tim pengembang bekerja bersama pengguna untuk merancang antarmuka pengguna dan fungsi-fungsi utama aplikasi.

C. Pengembangan

Tahap ini melibatkan pembangunan prototipe atau versi awal aplikasi dengan cepat. Pengembang menggunakan alat-alat pengembangan

yang cepat dan efisien untuk membuat aplikasi yang dapat diuji oleh pengguna.

D. Implementasi

Tahap pengiriman atau implementasi aplikasi ke lingkungan produksi. Ini melibatkan migrasi data, pelatihan pengguna, dan peluncuran aplikasi.

2.2.3 Object Oriented Analysis and Design (OOAD)

Menurut Dennis (2015), Analisis dan Desain Berorientasi Objek (OOAD) adalah suatu metodologi yang memadukan prinsip pemrograman berorientasi objek dengan pendekatan sistematis untuk menganalisis, merancang, dan mengembangkan perangkat lunak. Metodologi ini menitikberatkan pada pemodelan sistem sebagai himpunan objek yang berinteraksi satu sama lain untuk melaksanakan fungsi tertentu. OOAD menekankan pada identifikasi objek dalam domain masalah, penentuan karakteristik dan perilaku objek tersebut, serta pemodelan interaksi antar objek untuk merancang solusi perangkat lunak yang akurat mencerminkan sistem dunia nyata. Teknik ini mendukung pembuatan perangkat lunak yang modular, dapat digunakan kembali, dan mudah dipelihara.

Aspek utama OOAD mencakup:

1. Analisis Berorientasi Objek (OOA): Mengidentifikasi objek, hubungan antar objek, dan persyaratan yang menentukan perilaku sistem.
2. Desain Berorientasi Objek (OOD): Mengaplikasikan model desain dengan bahasa pemrograman berbasis objek seperti Java, C++, atau Python.
3. Pemrograman Berorientasi Objek (OOP): Mengimplementasikan model desain menggunakan kelas, antarmuka, dan interaksi untuk membangun sistem.

OOAD mendorong pendekatan pengembangan perangkat lunak yang terstruktur dan iteratif, menekankan pemahaman domain masalah,

penggunaan prinsip berorientasi objek dalam desain solusi, serta implementasi desain tersebut untuk menciptakan sistem perangkat lunak yang stabil dan dapat diukur. Selain itu, proses desain yang menggunakan OOAD memerlukan penerapan teknik tertentu yang bergantung pada *Unified Modeling Language* (UML) berfungsi sebagai bahasa standar untuk mendesain sistem perangkat lunak yang menggunakan konsep berorientasi objek.

2.2.4 *Unified Modelling Language* (UML)

Menurut Rosa dan Shalahuddin (2018), *Unified Modeling Language* (UML) adalah standar yang luas diterima di industri untuk merancang, menggambarkan, dan mendokumentasikan sistem perangkat lunak berorientasi objek, melakukan pemodelan dan perancangan, serta menggambarkan struktur dalam pengembangan berbasis objek. UML berperan sebagai sistematis visual yang digunakan menggambarkan dan berkomunikasi tentang sistem melalui penggunaan diagram serta dokumen pendukung. Namun, UML hanya digunakan untuk keperluan pemodelan.

1. *Function Model*



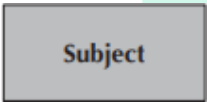

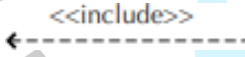
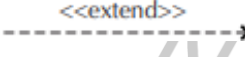

Diagram model fungsional UML digunakan untuk memperlihatkan proses bisnis serta interaksi sistem informasi dengan lingkungannya. Dalam pendekatan OOAD, terdapat dua jenis diagram yang digunakan untuk menampilkan model fungsional, yakni diagram *Use Case* dan diagram Aktivitas.

- *Use Case Diagram*

Gambaran dasar tentang dinamika antara sistem dan lingkungannya ditunjukkan oleh aktor dalam diagram *Use Case*. Untuk memudahkan pemahaman diagram *Use Case*, disarankan agar jumlah *Use Case* dalam satu model tidak melebihi tiga hingga sembilan. Selain itu, aktor harus ditempatkan dekat dengan *Use Case* yang relevan untuk

mengurangi kompleksitas. Diagram *Use Case* ini menggunakan notasi sebagai berikut.

Tabel 2. 1 Notasi Use Case Diagram

Notasi	Deskripsi
 Actor/Role Actor/Aktor	Yang menggambarkan manusia atau sistem yang mendapatkan manfaat namun berada di luar subjek (sistem).
 Use Case	Yang menunjukkan fungsionalitas utama dalam sistem.
 Subject	Yang menunjukkan cakupan dari suatu subjek (sistem) atau proses bisnis.
 Association	Yang mengaitkan aktor dengan <i>Use Case</i> yang terlibat dalam interaksi.
 Include	Yang merepresentasikan inklusi fungsional dari satu <i>Use Case</i> menuju <i>Use Case</i> lainnya.
 Extend	Yang menggambarkan perluasan dari suatu <i>Use Case</i> untuk mencakup perilaku opsional.
 Generalization	Yang menunjukkan spesialisasi <i>Use Case</i> atau aktor ke bentuk yang lebih umum (generalisasi).


Sumber : (Rosa dan Shalahuddin., 2018)

Untuk menyediakan informasi yang dibutuhkan dalam pembuatan diagram struktural dan perilaku, deskripsi *Use Case* disertakan dalam diagram *Use Case*. Deskripsi ini mencakup aliran atau urutan peristiwa yang terjadi dalam satu *Use Case*, dimulai dari proses pemicu hingga kemungkinan alur alternatif. Semua deskripsi tersebut disusun dalam format yang ringkas agar memudahkan pengguna dalam memahami *Use Case* tersebut.

- *Activity diagram*

Activity diagram memiliki kapabilitas untuk menggambarkan alur kerja atau urutan kegiatan dalam sistem. menggambarkan dan merancang berbagai proses serta alur kerja. Diagram ini dapat mencakup mulai dari tingkat yang lebih tinggi yang melibatkan berbagai *Use Case* hingga detailnya. Saat membuat *Activity diagram*, penting untuk mempertimbangkan kebutuhan sistem yang sedang dikembangkan dan dokumentasi yang telah ada, seperti *Use Case* diagram. Hal ini membantu menentukan apa yang diperlukan untuk mendukung proses bisnis yang relevan. Berikut adalah notasi-notasi yang digunakan dalam *Activity diagram*.

Tabel 2. 2 Notasi *Activity diagram*

Notasi	Deskripsi
 <i>Activity</i>	Yang menunjukkan orang atau sistem yang menerima keuntungan tetapi beroperasi di luar lingkup subjek (sistem).



Control Flow

Yang mengilustrasikan urutan pelaksanaan aktivitas.



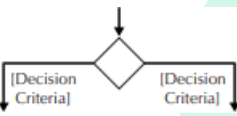
Initial node

Yang menunjukkan permulaan dari serangkaian aktivitas dalam diagram aktivitas.



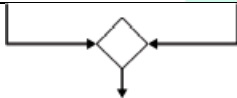
Final activity node

Yang menggambarkan berakhirnya aliran kontrol dalam sebuah diagram aktivitas.



Decision node

Yang menggambarkan kondisi pengujian untuk memverifikasi bahwa aliran hanya mengikuti satu jalur.



Merge Node

Yang digunakan untuk menggabungkan berbagai jalur keputusan yang berasal dari node keputusan.



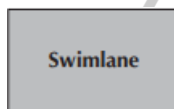
Fork Node

Yang digunakan untuk mempartisi perilaku menjadi aliran yang berjalan secara simultan atau parallel.



Join node

Yang digunakan untuk menggabungkan alur yang berbeda yang berlangsung secara simultan.



Swimlane

Yang digunakan untuk membagi diagram aktivitas menjadi baris dan kolom, di mana ini membantu menetapkan individu atau objek yang bertanggung jawab dalam menjalankan aktivitas tertentu.

Sumber : (Rosa dan Shalahuddin., 2018)

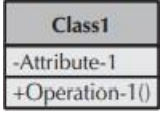
2. Structure Model



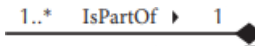
Diagram model struktur UML menggambarkan objek-objek yang mendukung proses bisnis dalam sebuah perusahaan. Diagram ini mencerminkan perilaku sistem dalam model fungsional, namun juga penting untuk memahami informasi yang digunakan dan dihasilkan oleh sistem, seperti data pengguna dan transaksi. Oleh karena itu, pemodelan struktur menampilkan objek-objek yang digunakan dan dihasilkan oleh sistem bisnis. Diagram kelas merupakan representasi dari model struktur dalam pendekatan Analisis dan Desain Berorientasi Objek (OOAD).

- *Class Diagram*

Diagram kelas adalah visualisasi entitas dan koneksi yang menghubungkan mereka konsisten dalam suatu sistem. Diagram ini juga menggambarkan perilaku kelas, termasuk metode atau operasi. Notasi tersebut digunakan untuk membuat visualisasi dari diagram kelas.

Tabel 2. 3 Notasi Class Diagram

Notasi	Deskripsi
 <p><i>Class</i></p>	Yang menggambarkan entitas seperti orang, tempat, atau objek yang diperlukan oleh sistem untuk menangkap dan mengumpulkan informasi.
<p>attribute name /derived attribute name</p> <p><i>Attribute</i></p>	Yang menggambarkan atribut yang menyatakan karakteristik dari suatu objek.
<p>operation name ()</p> <p><i>Operation</i></p>	Yang menggambarkan fungsi atau metode yang ada dalam sebuah kelas.
<p>AssociatedWith</p> <p>0..* 1</p> <p><i>Association</i></p>	Yang menggambarkan hubungan atau relasi antara kelas-kelas atau dengan kelas itu sendiri.

	Yang menggambarkan jenis hubungan yang sama terhadap beberapa kelas.
<i>Generalization</i>	
 0..* IsPartOf ▶ 1	Yang menggambarkan logika dari hubungan antara beberapa kelas, termasuk hubungan dengan kelas itu sendiri.
<i>Aggregation</i>	
 1..* IsPartOf ▶ 1	Yang menggambarkan secara fisik bagaimana hubungan antara beberapa kelas, termasuk hubungan dengan kelas itu sendiri.
<i>Composition</i>	

Sumber : (Rosa dan Shalahuddin., 2018)





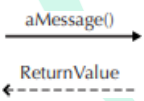
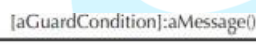

3. Behavior Mode

Menurut Dennis, 2015 diagram model perilaku UML dibuat untuk menggambarkan aspek internal yang dinamis dari sistem informasi yang mendukung proses bisnis perusahaan. Dalam konteks model fungsional, pembuatan model perilaku sering kali mengacu pada diagram *Use Case* karena fokusnya pada interaksi antara sistem dan pengguna sistem informasi terkait. Diagram interaksi seperti diagram urutan dan diagram komunikasi digunakan untuk memodelkan perilaku. *Sequence Diagram* adalah alat yang digunakan untuk memodelkan perilaku dalam studi ini.

- *Sequence Diagram*

Sequence diagram adalah bentuk visualisasi yang menampilkan objek-objek dan interaksi mereka Mereka yang terlibat dalam situasi penggunaan dan rangkaian pesan yang disampaikan di antara mereka selama proses tertentu. Dengan demikian, diagram ini berfungsi sebagai model dinamis yang mengilustrasikan urutan pesan antar objek dalam interaksi yang telah ditentukan. Selain itu, pembuatan *sequence diagram* bisa disesuaikan dengan struktur diagram *Use Case* yang telah didefinisikan sebelumnya. Diagram urutan menggunakan notasi khusus untuk menunjukkan komponen proses.

Tabel 2. 4 Notasi Sequence Diagram

Notasi	Deskripsi
 anActor Actor/Aktor	Notasi yang mengilustrasikan entitas manusia atau sistem yang mendapatkan manfaat dari sistem, tetapi beroperasi di luar dari <i>domain</i> atau lingkup sistem itu sendiri.
 Object	Yang terlibat dalam proses mengirim dan menerima pesan.
 Lifeline	Yang menggambarkan rentang waktu kehidupan suatu objek selama urutan proses tertentu.
 Execution	Yang menunjukkan saat suatu objek sedang mengirim atau menerima pesan adalah bagian dari <i>Sequence Diagram</i> .
 Message	Yang menggambarkan proses penyampaian informasi hubungan antara objek yang satu dengan objek yang lain dalam <i>Sequence Diagram</i> .
 Guard Condition	Yang menggambarkan kondisi yang harus dipenuhi agar suatu pesan dapat terkirim.
 Frame	Yang menggambarkan konteks dari suatu <i>Sequence Diagram</i> .

Sumber : (Rosa dan Shalahuddin., 2018)

2.2.5 Database

Menurut Dennis, 2015 Istilah "basis data" merujuk pada rangkaian data terstruktur yang tersimpan dan diorganisir sedemikian rupa sehingga

memfasilitasi pengambilan, pengelolaan, dan pembaruan informasi dengan efisien. Basis data biasanya digunakan dalam berbagai aplikasi dan sistem untuk menyimpan dan mengakses data dalam format yang terstruktur. Mereka menyediakan metode yang terpusat dan terorganisir untuk mengelola volume data yang besar, sehingga mempermudah akses dan manipulasi informasi sesuai kebutuhan.

Dalam konteks teknologi informasi, sebuah basis data biasanya terdiri dari tabel-tabel yang menyimpan data dalam bentuk baris dan kolom, serta hubungan antara tabel-tabel ini. Basis data dirancang untuk memastikan integritas data, keamanan, dan skalabilitas, memungkinkan beberapa pengguna atau aplikasi berinteraksi dengan data secara bersamaan.

Basis data hadir dalam berbagai bentuk, termasuk basis data relasional, berorientasi objek, hirarkis, dan berbasis grafik. NoSQL, dan lainnya, masing-masing dengan sifat dan aplikasi yang unik. Secara umum, basis data sangat penting untuk sistem informasi kontemporer karena menawarkan metode yang dapat diandalkan dan efektif untuk menangani dan menyimpan data.

2.2.6 Web

Menurut Dennis, 2015 Istilah "*web*" biasanya mengacu pada *World Wide Web*, yang merupakan sistem dokumen dan sumber daya yang terhubung secara global dan diakses melalui internet menggunakan peramban *web*. *Web* adalah jaringan informasi yang luas yang memungkinkan pengguna untuk mengunjungi halaman *web*, situs *web*, dan konten online melalui hyperlink. Tim Berners-Lee adalah pencipta *World Wide Web* pada tahun 1989, sejak saat itu menjadi bagian integral dari internet, memfasilitasi berbagi dan mengakses informasi secara global. Halaman *web* ditulis dalam bahasa seperti HTML, CSS, dan JavaScript, dan dihosting di server *web* yang mengirimkan konten kepada peramban pengguna saat permintaan dilakukan.

Selain halaman *web* statis, *web* juga mencakup konten dinamis, elemen multimedia, aplikasi interaktif, dan layanan seperti email, media sosial, dan belanja online. *Web* telah mengubah cara komunikasi, perdagangan, pendidikan, dan hiburan dilakukan, menjadikan informasi lebih dapat dijangkau oleh pengguna di seluruh dunia.

2.2.7 Testing

Testing dalam konteks pengembangan perangkat lunak merujuk pada proses evaluasi suatu sistem atau aplikasi untuk mengidentifikasi kesalahan, bug, atau cacat. Tujuan utama dari pengujian adalah untuk memverifikasi bahwa perangkat lunak beroperasi dengan benar, memenuhi persyaratan yang telah ditentukan, dan berkinerja seperti yang diharapkan dalam berbagai kondisi.

Pengujian perangkat lunak meliputi menjalankan sistem untuk mencari bug dan mengonfirmasi bahwa sistem tersebut berfungsi sebagaimana mestinya. Sepanjang siklus pengembangan, pengujian dapat dilakukan pada fase yang berbeda. Termasuk dalam proses validasi adalah pengujian perangkat lunak, yang meliputi pengujian unit, integrasi, sistem, dan penerimaan. Setiap jenis pengujian membantu menjamin ketergantungan dan kualitas perangkat lunak dengan berkonsentrasi pada fitur yang berbeda dari program.

Pengujian sangat krusial dalam proses pengembangan perangkat lunak untuk mendeteksi dan memperbaiki masalah secara dini, mengurangi risiko kegagalan perangkat lunak, dan meningkatkan kualitas keseluruhan produk. Pengujian juga membantu memvalidasi bahwa perangkat lunak memenuhi harapan pengguna, mematuhi persyaratan yang ada, dan berfungsi dengan benar di berbagai lingkungan.

Secara keseluruhan, pengujian memainkan peran krusial dalam memastikan kualitas, keandalan, dan kinerja aplikasi perangkat lunak sebelum dideploy ke pengguna

a) **Black Box Testing**

Black Box Testing adalah metode pengujian perangkat lunak yang menitikberatkan pada penilaian kinerja eksternal suatu program tanpa mempertimbangkan struktur internal atau rancangan logis kode tersebut. Konsep dasar *Black Box Testing* didasarkan pada prinsip bahwa pengujian harus dilakukan dari perspektif pengguna akhir, di mana evaluasi dilakukan terhadap input dan output yang diharapkan dari sistem atau perangkat lunak.

Metode Pengujian Black Box mengacu pada spesifikasi dan persyaratan fungsional. Pengujian dilakukan dengan menginput berbagai jenis input ke dalam sistem dan memeriksa output yang dihasilkan untuk memastikan bahwa sistem berperilaku sesuai dengan harapan. Tujuan utama tujuan dari Black Box Testing adalah untuk mengidentifikasi kesalahan atau bug dalam fungsionalitas sistem tanpa memerlukan pengetahuan detail tentang implementasi internalnya.

Konsep *Black Box Testing* juga terintegrasi dalam teori pengujian perangkat lunak secara umum, yang mencakup strategi, teknik, dan metode untuk mengidentifikasi serta mengurangi risiko kegagalan perangkat lunak. Dalam praktiknya, *Black Box Testing* sering digunakan bersama dengan *white box Testing* (pengujian struktural) dan *grey box Testing* (kombinasi black box dan white box Testing) untuk memastikan cakupan pengujian yang komprehensif.

Dengan mengadopsi dasar teoritis *Black Box Testing*, para pengembang perangkat lunak dapat merencanakan serta menjalankan serangkaian tes yang efektif untuk memvalidasi dan memverifikasi fungsionalitas sistem. Hal ini penting untuk memverifikasi bahwa perangkat lunak yang dibangun sesuai dengan kebutuhan dan ekspektasi pengguna dengan baik.

2.3 Literature Review

Penulis melakukan literatur *review* sebagai pendukung dalam merancang riset penelitian, yang mencakup referensi-referensi yang relevan dengan topik yang dibahas oleh penulis. Beberapa sumber literatur yang digunakan dalam literatur *review* ini mencakup:

- Jurnal berjudul “SISTEM INFORMASI PENGADUAN MAHASISWA STIE ADHI UNGGUL BHIRAWA SURAKARTA BERBASIS *WEB*” Tulisan oleh Yusuf Sutanto dan Rat Aditya dalam Indonesian Journal of Network & Security, Vol 10, No 3, September 2021 membahas pengembangan Sistem Informasi Pengaduan Mahasiswa STIE Adhi Unggul Bhirawa berbasis website, dengan tujuan memfasilitasi interaksi antara mahasiswa dan pihak kampus. Melalui platform pengaduan berbasis web ini, mahasiswa dapat menyampaikan pandangan, aspirasi, rekomendasi, dan kritik mereka secara fleksibel. Penelitian ini menggunakan metode analisis dan observasi: analisis untuk mengidentifikasi masalah dalam penelitian dan observasi untuk memahami interaksi antara mahasiswa dan pihak kampus. Hasil dari kedua metode tersebut menjadi landasan untuk pengembangan sistem pengaduan berbasis website, diharapkan dapat meningkatkan komunikasi di lingkungan kampus.
- Jurnal berjudul “SISTEM INFORMASI PENGADUAN SISWA BERBASIS *WEBSITE* DENGAN FRAMEWORK LARAVEL” Tulisan oleh Winda Uli Sari Manulu, Lukman Hakim, dan Cindi Wulandari dalam Journal of Information System Research, Vol 4, No 3, April 2023 membahas pengembangan sistem pengaduan siswa berbasis web untuk SMK Yadika Lubuklinggau sebagai solusi yang lebih efisien daripada prosedur manual yang masih digunakan saat ini. Saat ini, siswa harus menulis keluhannya di atas kertas dan memasukkannya ke dalam kotak umpan balik, yang dinilai kurang efisien. Untuk mengatasi masalah ini, penelitian ini mengusulkan penggunaan teknologi berupa website sebagai sarana untuk siswa menyampaikan pengaduan dengan lebih mudah dan cepat. Sistem ini dikembangkan

menggunakan bahasa pemrograman PHP dengan pendekatan Model View Controller (MVC) dan menggunakan Laravel Framework. Pemodelan sistem melibatkan diagram Use Case, skenario Use Case, diagram aktivitas, diagram urutan, diagram kelas, serta desain input dan output. Dengan penerapan teknologi ini, diharapkan siswa dapat lebih efektif dalam menyampaikan pengaduan mereka secara online di luar jam sekolah yang telah ditentukan. Jurnal ini mengemukakan manfaat berupa peningkatan aksesibilitas bagi siswa dalam menyampaikan pengaduan secara elektronik.

- Jurnal berjudul “PERANCANGAN SISTEM INFORMASI LAYANAN PENGADUAN BADAN EKSEKUTIF MAHASISWA BERBASIS *WEB*” Pembuatan sistem pengaduan online untuk Badan Eksekutif Mahasiswa (BEM) di Universitas PGRI Sumatera Barat dibahas dalam sebuah penelitian oleh Elda Sulistiawati Elda, Heri Mulyono, dan Anggri Yulio Pernanda yang dipublikasikan di Jurnal Pendidikan Teknologi Informasi, Vol. 3, No. 1, Maret 2023. Tujuan utama dari sistem ini adalah untuk membantu BEM dalam menangani laporan masalah mahasiswa agar dapat terorganisir, diarsipkan secara efektif, dan sesuai dengan hukum yang berlaku. Selain itu, sistem ini juga dimaksudkan untuk menyederhanakan prosedur pelaporan dan memungkinkan BEM dan mahasiswa untuk menindaklanjuti laporan yang diajukan.
- “METODE RAD PADA PERANCANGAN *WEBSITE* PENGADUAN MASYARAKAT DESA KARANG SATRIA DENGAN UI/UX BERBASIS UCD” Penelitian Elda Sulistiawati Elda, Heri Mulyono, dan Anggri Yulio Pernanda diterbitkan dalam Jurnal Pendidikan Teknologi Informasi, Vol. 3, No. 1, Maret 2023. Penelitian ini membahas tentang pembuatan sistem pengaduan berbasis web untuk Badan Eksekutif Mahasiswa (BEM) Universitas PGRI Sumatera Barat. Tujuan utama dari sistem ini adalah untuk membantu BEM dalam menangani laporan masalah mahasiswa agar dapat terorganisir, diarsipkan secara efektif, dan sesuai dengan hukum yang berlaku. Selain itu, sistem ini juga dimaksudkan untuk menyederhanakan prosedur pelaporan dan memungkinkan BEM dan mahasiswa untuk menindaklanjuti laporan yang diajukan. Analisis menyeluruh dilakukan di bidang desain

antarmuka pengguna dengan menggunakan Metode User Centered Design (UCD) untuk memahami kebutuhan dan pengalaman pengguna. Menurut penelitian tersebut, prototipe situs web mendapat skor lebih dari 78 untuk kegunaan, kemudahan penggunaan, kemudahan belajar, dan kepuasan, yang menunjukkan tingkat persetujuan yang tinggi di antara pengguna bahwa situs web tersebut bermanfaat, mudah digunakan, mudah dipelajari, dan memuaskan.

- Jurnal berjudul “RANCANG BANGUN APLIKASI PENGADUAN PELANGGAN BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER DI INDOTECHNO PURWOKERTO” Casro Casro, Yuli Purwati, Gustin Setyaningsih, dan Adam Prayogo Kuncoro menulis artikel ini untuk Jurnal Sains dan Informatika pada bulan Desember 2020. Artikel ini mengeksplorasi bagaimana Perusahaan Indotechno Purwokerto menggunakan kerangka kerja CodeIgniter untuk membangun aplikasi pengaduan pelanggan berbasis web, dengan penekanan pada peningkatan layanan dan penanganan data pengaduan pelanggan. PC anggota staf dukungan pelanggan berisi perangkat lunak Microsoft Excel, yang digunakan untuk menangani data yang mereka kumpulkan dari pelanggan melalui pengumpulan keluhan secara manual menggunakan formulir kertas. Pelanggan yang ingin mengajukan keluhan harus menelepon call center perusahaan atau mengunjungi kantor secara langsung karena saat ini tidak ada jaringan komputer yang menghubungkan staf. Tujuan dari penelitian ini adalah untuk membuat solusi berbasis web untuk keluhan pelanggan. Melalui studi literatur, wawancara, dan observasi, data dikumpulkan. Metode pengembangan sistem waterfall diterapkan. Salah satu hasil dari penelitian ini adalah aplikasi berbasis web untuk pengaduan konsumen, yang diharapkan dapat mempercepat penyelesaian keluhan konsumen. Pengujian blackbox menunjukkan bahwa fungsionalitas aplikasi telah berjalan sebagaimana mestinya. Diharapkan dengan adanya aplikasi ini, layanan pelanggan dapat ditingkatkan dan penanganan data keluhan konsumen menjadi lebih efektif.