BAB II
TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Pengertian Perancangan

Perancangan adalah prosedur yang melibatkan penentuan struktur, fungsi, dan estetika suatu produk atau sistem. Perancangan sering kali mencakup penulisan, analisis, dan pengujian untuk memastikan hasil yang optimal dan memenuhi kebutuhan pengguna. Menurut Setiawan, B. (2023) dalam artikel yang diterbitkan di Jurnal Teknologi Informasi "perancangan proses iteratif yang melibatkan identifikasi kebutuhan, perancangan konsep, dan implementasi solusi untuk menciptakan produk atau sistem yang efisien dan efektif." Definisi ini menekankan aspek iteratif dan kebutuhan untuk menyesuaikan solusi berdasarkan umpan balik pengguna. Berikut adalah karakteristik perancangan yaitu :

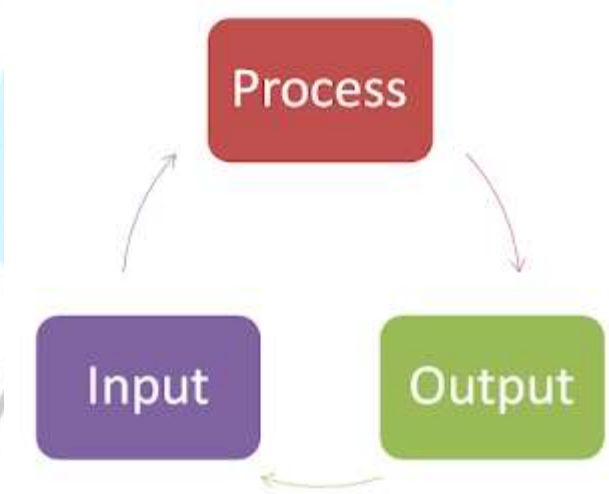
a. Iteratif: Proses perancangan bersifat iteratif, artinya terus berulang sampai solusi optimal tercapai. Setiap iterasi melibatkan pengujian dan revisi berdasarkan umpan balik.
b. Sistematis: Perancangan mengikuti langkah-langkah yang terstruktur mulai dari identifikasi kebutuhan hingga evaluasi akhir.
c. Pengguna Sentris: Fokus utama perancangan adalah memenuhi kebutuhan dan harapan pengguna akhir.
d. Kreatif dan Analitis: Menggabungkan kreativitas dalam perancangan konsep dengan analisis untuk memastikan kelayakan dan efisiensi.

2.1.2 Pengertian Sistem

Sistem merupakan suatu gagasan yang digunakan untuk menggambarkan suatu substansi yang terdiri dari bagian-bagian yang berkolaborasi dan bekerja sama untuk mencapai suatu visi dan tujuan tertentu.. Menurut Chrisdianti (2023) menyatakan *system* ialah sekelompok aspek yang saling bahu-membahu dan berkaitan untuk menggapai visi. Pengertian dan penjelasan ini menegaskan utamanya dari esesnsi hubungan antar satu sama lain dan bekerja bersatu antar unsur pada sistem untuk

menggapai visi sistem. Adapun menurut Ladewi Puspa, Erlangga (2023) dalam Jurnal Teknologi Informasi Sistem dapat dijelaskan sebagai rangkaian unsur yang terhubung dan miliki antarhubungan yang terkait mencapai hasil yang diinginkan. Definisi ini menegaskan interaksi antar unsur untuk menghasilkan output yang diharapkan. Beberapa karakteristik utama dari sistem antara lain:

a. Tujuan atau Sasaran: Setiap sistem memiliki tujuan atau sasaran yang ingin dicapai.
b. Komponen atau Elemen: *System* dibagi menjadi beberapa komponen / unsur, berupa manusia, mesin, perangkat lunak, dan prosedur.
c. Interaksi: Unsur dalam sistem saling bekerja dan berkaitan untuk menggapai hasil yang diharapkan.
d. Lingkungan: Sistem beroperasi dalam suatu lingkungan yang saling memberikan pengaruh satu sama lain.
e. Keterkaitan: Komponen dalam sistem saling terkait dan perubahan pada satu komponen dapat mempengaruhi komponen lainnya.



Gambar 2. 1 Model Sistem

(*Arvindz Eclass Tuesday, November 01, 2022 Class 9 IT 402*)

2.1.3 Pengertian Perancangan Sistem

Perancangan sistem adalah prosedur yang melibatkan pengembangan dan implementasi solusi teknologi informasi untuk memenuhi kebutuhan khusus

pengguna atau organisasi. Proses ini mencakup beberapa kegiatan sistematis, diawali dengan identifikasi kebutuhan hingga penerapan dan perawatan sistem. Tujuan perancangan sistem adalah menciptakan kerangka program yang berkaitan dengan kegunaan yang akan dipakai oleh masyarakat yang akan menjadi target dalam program dan visi bisnis organisasi. Langkah-langkahnya mencakup identifikasi sebuah kebutuhan, desain sistem, perancangan perangkat lunak, pengujian, dan implementasi (Shelly dan Rosenblatt, 2019). Konsep ini menekankan pentingnya pendekatan *iterative* dalam perancangan sistem untuk memastikan fleksibilitas dan adaptabilitas terhadap perubahan kebutuhan.

Sementara itu, Whitten dan Bentley (2020) mengatakan bahwa perancangan sistem juga melibatkan aspek manajemen proyek untuk memastikan bahwa setiap tahap dilakukan sesuai jadwal dan anggaran yang telah ditetapkan. Mereka juga menekankan pentingnya Komunikasi yang efektif antara tim pengembang dan pihak-pihak terkait untuk menghindari kesalahpahaman dan memastikan bahwa kebutuhan pengguna tercermin dalam desain sistem. Adapun tahapan-tahapan dalam perancangan sistem meliputi:

1. Analisis Kebutuhan: Mengidentifikasi dan mendokumentasikan kebutuhan fungsional dan non-fungsional sistem. Tujuan tahap ini adalah memahami apa yang dibutuhkan pengguna dan bagaimana sistem akan digunakan.
2. Desain Sistem: Merancang arsitektur, komponen, antarmuka, dan *database* sistem. Desain ini mencakup pembuatan diagram alur data, diagram *entity-relationship*, dan spesifikasi desain lainnya yang diperlukan untuk membangun sistem.
3. Perancangan dan Pengujian: Membuat perangkat lunak sesuai dengan desain dan kebutuhan yang telah disusun, lalu mengujinya untuk memastikan sistem berjalan sesuai spesifikasi dan tidak memiliki kesalahan.
4. Implementasi dan Pemeliharaan: Mengintegrasikan sistem ke dalam lingkungan operasional, melatih pengguna, serta melakukan pemeliharaan dan perbaikan yang diperlukan untuk menjaga kinerja sistem tetap optimal.

2.1.4 Pengertian *Booking*

Booking adalah proses pemesanan atau reservasi yang dilakukan oleh individu atau kelompok untuk memastikan ketersediaan suatu layanan atau fasilitas pada waktu tertentu. Proses ini seringkali dilakukan untuk berbagai kebutuhan seperti perjalanan (misalnya tiket pesawat, kereta, atau bus), akomodasi (hotel, apartemen, atau villa), restoran, acara (konser, seminar, atau pertunjukan), serta layanan lainnya seperti penyewaan mobil atau fasilitas olahraga. Berikut adalah beberapa poin penting mengenai *booking* yaitu mengamankan tempat atau layanan tertentu agar tidak kehabisan slot atau kapasitas yang tersedia, terutama jika layanan tersebut memiliki permintaan tinggi. Adapun beberapa proses *booking* melibatkan langkah-langkah berikut yaitu :

1. Pencarian: Mencari informasi mengenai ketersediaan layanan atau fasilitas yang diinginkan.
2. Pemilihan: Memilih opsi yang paling sesuai dengan kebutuhan, seperti tanggal, waktu, jenis layanan, dan lokasi.
3. Pemesanan: Mengisi formulir pemesanan atau menghubungi penyedia layanan untuk melakukan reservasi.
4. Konfirmasi: Menerima konfirmasi dari penyedia layanan yang berisi detail pemesanan dan informasi penting lainnya.
5. Pembayaran: Tergantung pada kebijakan penyedia layanan, pembayaran bisa dilakukan di muka (sebagian atau seluruhnya), atau pada saat menggunakan layanan. Beberapa penyedia pelayanan juga memberikan opsi untuk pembatalan dan pengembalian dana sesuai dengan kebijakan yang berlaku.
6. Keuntungan: Dengan melakukan *booking*, pengguna bisa memastikan mereka mendapatkan layanan atau fasilitas yang diinginkan pada waktu yang telah direncanakan, menghindari ketidaknyamanan karena keterbatasan kapasitas atau ketersediaan.
7. Teknologi: Saat ini, proses booking banyak dilakukan melalui platform digital seperti situs *web* dan aplikasi, yang mempermudah pengguna untuk melakukan pencarian, pemesanan, dan pembayaran secara cepat dan efisien.

2.1.5 Pengertian *Coworking Space*

Coworking Space adalah ruangan yang digunakan untuk bekerja bersama di mana individu dan kelompok dari berbagai latar belakang profesional dapat bekerja dalam lingkungan yang terbuka dan kolaboratif. Konsep ini menawarkan alternatif fleksibel terhadap kantor tradisional, memberikan fasilitas lengkap dengan biaya yang lebih efisien dan lingkungan yang lebih dinamis. *Coworking Space* adalah lingkungan kerja di mana pekerja dari berbagai perusahaan atau latar belakang profesional berbagi ruang fisik dan fasilitas. *Coworking Space* menyediakan lingkungan yang mendukung kolaborasi, inovasi, dan pertukaran ide, sambil memberikan fleksibilitas bagi individu dan perusahaan kecil untuk memilih tempat dan waktu kerja yang paling sesuai dengan kebutuhan mereka (Spinuzzi, 2019).

Karakteristik utama dari *Coworking Space* meliputi:

a. Lingkungan Kerja yang Kolaboratif: *Coworking Space* dirancang untuk mendorong interaksi dan kolaborasi antara anggotanya, sering kali menyediakan area terbuka, ruang meeting, dan fasilitas untuk acara atau workshop.
b. Fleksibilitas: Anggota *Coworking Space* dapat memilih paket keanggotaan yang sesuai dengan kebutuhan mereka, mulai dari penggunaan harian hingga bulanan, dengan opsi untuk menyewa meja, ruangan pribadi, atau area kerja bersama.
c. Fasilitas Lengkap: *Coworking Space* biasanya dilengkapi dengan fasilitas modern seperti koneksi internet cepat, ruang rapat, area *lounge*, dapur, serta fasilitas tambahan seperti gym atau ruang bermain, tergantung pada lokasi.
d. Komunitas dan Jaringan: *Coworking Space* sering kali menyelenggarakan acara jaringan, *workshop*, dan seminar untuk membantu anggotanya mengembangkan keterampilan dan membangun jaringan profesional.

Adapun beberapa pendapat para ahli mengenai *Coworking Space* didefinisikan sebagai berikut.

1. Menurut penulisan (Stumpf, 2013) *Coworking Space* merupakan sebuah tempat untuk pelaksanaan suatu pekerjaan dalam organisasi ataupun institusi yang memberikan kenyamanan dalam bekerja dengan

mengutamakan kerja sama, luwes, dan mandiri, yang didasarkan kepercayaan, dan poin-poin antara anggotanya. *Coworking Space* menyediakan fasilitas seperti meja kerja, ruang pertemuan, akses internet, dan fasilitas lainnya yang memungkinkan individu dan perusahaan untuk bekerja secara fleksibel dan ekonomis.

2. (Gandini, 2021) menyatakan *Coworking Space* juga menawarkan manfaat psikososial yang signifikan bagi anggotanya. Selain memberikan fleksibilitas kerja, *Coworking Space* menciptakan rasa komunitas dan kesempatan untuk belajar dari sesama anggota yang mungkin memiliki keahlian dan pengalaman yang berbeda.

2.1.6 Pengertian Aplikasi

Aplikasi adalah program yang dibuat untuk membantu Masyarakat yang menjadi target aplikasi dalam menyelesaikan pekerjaan tertentu seperti produktivitas, komunikasi, dan hiburan. Aplikasi dapat berjalan pada berbagai *platform*, termasuk *desktop*, *web*, dan *mobile*. Pada konsep teknologi informasi, Aplikasi mengacu pada program yang dibuat untuk membantu masyarakat target aplikasi dalam menyelesaikan tugas-tugas khusus pada perangkat komputasi seperti komputer, *tablet*, atau *smartphone*. Aplikasi dapat berupa program sederhana hingga perangkat lunak yang kompleks dengan berbagai fungsi. Dalam jurnal yang diterbitkan oleh Universitas Sam Ratulangi, (Koloay, Klaudio 2023) mendefinisikan aplikasi sebagai program yang dirancang untuk menjalankan tugas-tugas tertentu pada perangkat komputer atau *mobile*, dengan tujuan memudahkan pengguna dalam berbagai aktivitas sehari-hari.

Aplikasi memiliki karakteristik sebagai berikut.

a. Spesifik terhadap tugas : Aplikasi dirancang untuk tugas tertentu, misalnya aplikasi pengolah kata untuk menulis dokumen, aplikasi pesan untuk komunikasi, atau aplikasi reservasi untuk memesan layanan.
b. Interaksi pengguna: Umumnya, aplikasi dilengkapi dengan antarmuka yang memungkinkan pengguna berinteraksi dengan perangkat lunak, seperti melalui menu, tombol, dan input lainnya.

   Berikut kategori aplikasi yaitu :

a. Aplikasi *Desktop*: Perangkat lunak yang diinstal dan dijalankan di komputer desktop atau laptop, seperti *Microsoft Word* atau *Adobe Photoshop.*
b. Aplikasi Web: Perangkat lunak yang dijalankan melalui web browser, seperti *Google Docs* atau *Facebook.*
c. Aplikasi *mobile*: Program yang dibangun untuk smartphone dan tablet, seperti *Instagram* atau *Gojek*.
d. Penggunaan aplikasi menawarkan berbagai manfaat, di antaranya:
e. Efisiensi: Mempermudah dan mempercepat proses yang biasanya memakan waktu lebih lama jika dilakukan secara manual.
f. Aksesibilitas: Memudahkan pengguna untuk mengakses layanan atau informasi secara fleksibel melalui perangkat mobile.
g. Pengalaman Pengguna menjadi Lebih Baik: Antarmuka yang dirancang dengan baik meningkatkan kepuasan pengguna dalam berinteraksi dengan perangkat lunak.

2.1.7 Pengertian *Mobile*

Mobile secara umum memiliki arti dapat bergerak atau berpindah. Dalam konteks teknologi, mobile mengacu pada perangkat yang portabel, kecil, dan memiliki kemampuan komputasi, seperti smartphone, tablet, laptop, dan jam tangan pintar. Perangkat mobile ini tidak memerlukan kabel untuk terhubung ke internet atau jaringan lain, sehingga memungkinkan penggunanya untuk berkomunikasi, mengakses informasi, dan melakukan berbagai aktivitas di mana pun mereka berada. Berikut beberapa karakteristik utama perangkat mobile:

a. Portabel: Mudah dibawa karena memiliki ukuran dan berat yang kecil.
b. Nirkabel: Terhubung ke internet atau jaringan lain tanpa kabel, seperti melalui Wi-Fi atau jaringan seluler.
c. Komputasi: Memiliki kemampuan untuk menjalankan aplikasi dan melakukan berbagai tugas komputasi.
d. Interaktif: Pengguna dapat berinteraksi dengan perangkat melalui layar sentuh, keyboard, atau suara.
e. Multifungsi: Dapat digunakan untuk berbagai aktivitas seperti korespondensi, browsing web, main-main, menonton video, dan masih banyak lagi.

2.1.8 Pengertian Sistem Informasi

Sistem informasi adalah sistem yang mendukung pengambilan keputusan dan operasi bisnis dengan mengelola dan menyebarkan informasi. Mereka terdiri dari teknologi, manusia, dan proses. Sistem informasi mencakup peralatan, pemrograman, organisasi, informasi, serta teknik dan individu yang menggunakannya untuk mengumpulkan, memproses, dan mengedarkan data. Sesuai dengan penjelasan dari (Laudon, K. C., dan Laudon, J. P. (2023)), Sistem Informasi merupakan sekumpulan unsur yang terhubung dan memiliki tugas untuk menyatukan, proses unsur, mengamankan, dan mendistribusikan informasi yang berfungsi untuk menentukan pilihan pada situasi maupun kondisi tertentu dalam sebuah organisasi. Mereka menekankan bahwa sistem informasi modern memanfaatkan teknologi digital untuk meningkatkan kemudahan dan fleksibelitas pada aktivitas disuatu organisasi. Penerapan Sistem Informasi dalam dunia IT, yaitu :

a. *Database Management System* (DBMS): Pengelolaan data yang dilakukan *user* seperti menyimpan, memproses, dan mengambil data secara struktur. Contoh DBMS meliputi MySQL, Oracle, dan Microsoft SQL Server.
b. *Enterprise Resource Planning* (ERP) *System*: ERP *System* menggabungkan semua bidang operasional suatu perusahaan, termasuk perencanaan, produksi, penjualan, pemasaran, inventaris, pengiriman, dan akuntansi. Contoh ERP termasuk SAP, Oracle ERP, dan Microsoft Dynamics.
c. Sistem Informasi Manajemen *Development Life Cycle* (SDLC) menguraikan serangkaian tahapan atau fase yang harus dilalui mulai dari ide awal hingga implementasi dan pemeliharaan sistem tersebut. Setiap tahap dalam SDLC memiliki tujuan dan aktivitas yang terdefinisi dengan baik, dan sering kali proses ini diatur secara hierarkis, sehingga memungkinkan pengembang untuk mengelola proyek dengan lebih terstruktur dan efisien.- MIS): MIS menyediakan informasi yang dibutuhkan oleh manajemen untuk membuat keputusan strategis. Contoh MIS adalah dashboard yang memberikan laporan kinerja perusahaan dan analisis data operasional.
d. *Decision Support System*s atau disingkat DSS: Program ini berguna untuk pengelola dalam menjawab sebuah keputusan dengan menyediakan data

analitik dan model keputusan. Contoh DSS mencakup sistem yang digunakan untuk analisis risiko, perencanaan keuangan, dan pemodelan bisnis. Ini merupakan pendekatan sistematis dalam desain program. Berikut adalah manfaat Sistem Informasi, yaitu:

a. Meningkatkan Kegiatan Operasional dengan Optimal: Program membantu mengotomatisasi prosedur transaksi dalam organisasi, mengurangi pengeluaran, dan memajukan kegiatan produksi.
b. Mendukung Pengambilan Keputusan: Dengan memberikan data tepat dan relevan. Program membantu pengelola mengambil keputusan akurat.
c. Meningkatkan Layanan Pelanggan: Sistem informasi memungkinkan perusahaan mengelola interaksi dengan pelanggan secara lebih efektif, meningkatkan kepuasan dan loyalitas pelanggan.
d. Mendukung Strategi Bisnis : Sistem informasi membantu organisasi dalam merencanakan dan melaksanakan strategi bisnis dengan memberikan wawasan yang diperlukan untuk mengambil tindakan yang tepat.

2.2 Teori Khusus

2.2.1 *System Development Life Cycle* (SDLC)

Merupakan pendekatan sistematis guna membangun program. SDLC menjelaskan kumpulan tahap dari konsepsi awal ide hingga implementasi dan pemeliharaan sistem tersebut. Setiap tahap dalam SDLC memiliki tujuan dan aktivitas yang jelas, dan sering kali proses ini diatur secara hierarkis untuk memungkinkan pengembang mengelola proyek dengan lebih terstruktur dan efisien. SDLC memiliki empat fase utama: perencanaan, analisis, desain, dan pelaksanaan. Meskipun setiap proyek mungkin menekankan suatu proses dengan cara beragam, tetapi setiap proyek mencakup dari keempat tahap ini. Setiap kegiatan pada level tertentu memiliki kumpulan langkah yang bergantung pada metode yang sudah dipilih untuk menghasilkan dokumen dan file yang memberikan pemahaman lebih dalam tentang proyek tersebut (Dennis, Wixom, & Tegarden, 2015).

2.2.2 Metode *Rapid Application Development* (RAD)

RAD merupakan pendekatan mengembangkan program yang mengutamakan dalam pembuatan prototipe yang dapat diuji dengan cepat dan berulang kali. Tujuannya adalah untuk menghasilkan aplikasi yang berkualitas tinggi dalam waktu singkat dengan melibatkan pengguna secara aktif dalam proses pengembangan. Contoh penggunaan RAD yaitu diantara: mengembangkan aplikasi *web* sederhana, membuat *prototype* untuk aplikasi baru, mengembangkan aplikasi yang perlu menyesuaikan dengan persyaratan yang berubah dengan cepat. Karakteristik utama RAD yaitu:

a. Berfokus pada *prototype*: RAD menggunakan prototipe untuk mendemonstrasikan fungsionalitas aplikasi kepada pengguna dan mendapatkan umpan balik mereka.
b. Iteratif: RAD adalah proses berulang di mana prototipe diperbaiki dan ditingkatkan berdasarkan umpan balik pengguna.
c. Berpusat pada pengguna: RAD melibatkan pengguna secara aktif dalam proses pengembangan untuk memastikan bahwa aplikasi memenuhi kebutuhan mereka.
d. Menggunakan alat dan teknik khusus: RAD menggunakan berbagai alat dan teknik untuk mempercepat proses pengembangan, seperti pemrograman berpasangan, pemrograman *ekstrim*, dan alat *Computer-Aided Software Engineering* (CASE).

Adapun Manfaat RAD adalah sebagai berikut:

a. Cepat: RAD dapat menghasilkan aplikasi dalam waktu yang lebih singkat daripada metodologi pengembangan tradisional.
b. Berkualitas tinggi: RAD membantu memastikan bahwa aplikasi berkualitas tinggi dengan melibatkan pengguna dalam proses pengembangan.
c. Berfokus pada pengguna: RAD membantu memastikan bahwa aplikasi memenuhi kebutuhan pengguna.
d. Fleksibel: RAD adalah proses yang dapat menyesuaikan diri dengan kebutuhan khusus dari sebuah proyek.

Adapun Kekurangan RAD sebagai berikut:

a. Tidak ideal dalam produk dengan kerumitan yang tinggi: Tidak sesuai dengan rancangan sebuah produk / pekerjaan yang kerumitannya tinggi dan memiliki kebutuhan keamanan yang sangat penting.
b. Membutuhkan keterlibatan pengguna yang tinggi: RAD membutuhkan keterlibatan pengguna yang tinggi agar berhasil.
c. Dokumentasi mungkin kurang: RAD mungkin menghasilkan dokumentasi yang kurang lengkap daripada metodologi pengembangan tradisional.

2.2.3 Object-Oriented Analysis and Design (OOAD)

Merupakan strategi dengan konsep objek dan kelas untuk menganalisis dan merancang sistem perangkat lunak. OOAD memungkinkan desainer untuk membuat model yang mendekati situasi dunia nyata. Hal ini mempermudah pengembang dan pemangku kepentingan untuk memahami desain sistem. OOAD juga sering menggunakan use case untuk mengidentifikasi kebutuhan dan interaksi pengguna dengan sistem, sehingga memastikan sistem yang dikembangkan dapat memenuhi persyaratan pengguna secara efektif.

OOAD sangat cocok digunakan dalam perancangan aplikasi booking *Coworking Space* berbasis *mobile* karena beberapa alasan berikut Contoh Penerapan OOAD Dalam perancangan aplikasi booking *Coworking Space* berbasis *mobile*, OOAD dapat digunakan untuk:

a. Membuat Kelas seperti *User, Reservation, Workspace, dan Payment.*
b. Mendefinisikan Atribut seperti nama pengguna, ID reservasi, jenis ruang kerja, dan metode pembayaran.
c. Merancang Metode untuk tindakan seperti membuat reservasi, membatalkan reservasi, dan memproses pembayaran.
d. Menggambarkan Interaksi menggunakan diagram urutan untuk menunjukkan bagaimana pengguna membuat dan mengelola reservasi melalui aplikasi.

2.2.4 *Unified Modelling Language* (UML)

Dalam rekayasa perangkat lunak, UML adalah bahasa untuk membangun, memodelkan, dan pengarsipan program berbasis objek. UML memberikan

dokumentasi realistis yang dinormalisasi, termasuk garis besar, misalnya bagan kelas, grafik kasus penggunaan, garis besar pergerakan, grafik susunan, dan lain-lain. Berbagai komponen sistem perangkat lunak digambarkan melalui diagram ini. UML memungkinkan para insinyur pemrograman untuk berkomunikasi secara lebih nyata dalam grup rencana karena dokumentasinya bersifat umum dan lugas. Pengembang dapat menggunakan UML untuk mendeskripsikan struktur statis sistem, seperti kelas, hubungan antar kelas, dan properti kelas. Mereka juga dapat menggambarkan perilaku dinamis sistem, seperti bagaimana proses bisnis bekerja sama atau bagaimana objek berinteraksi satu sama lain.

Salah satu keunggulan utama dari UML adalah fleksibilitas nya yang memungkinkan adaptasi untuk berbagai kebutuhan proyek. Meskipun UML memiliki standar notasi yang telah ditetapkan, ia juga memungkinkan penggunaan ekstensi dan penyesuaian sesuai kebutuhan proyek tertentu. UML tidak hanya berguna selama fase perancangan perangkat lunak, tetapi juga selama fase analisis, implementasi, dan dokumentasi proyek. Dengan menggunakan UML, para pengembang dapat secara visual menganalisis dan merancang sistem secara holistik, membantu dalam memahami kompleksitas sistem dengan lebih baik, serta mengurangi kesalahan dalam desain dan implementasi.

a) *Use Case*

Jenis bagan UML yang digunakan untuk menggambarkan hubungan antara kerangka kerja dan penghibur adalah *use case*. Bagan ini terdiri dari beberapa komponen, termasuk penghibur, kasus penggunaan, kerangka kerja, dan koneksinya, misalnya, digabungkan dan diperluas.Diagram ini juga menunjukkan tindakan-tindakan yang bisa dilakukan oleh aktor terhadap use case. Jenis diagram ini sangat bermanfaat untuk menggambarkan hubungan antara sistem dan aktor. Menurut Valacich dan George (2016), use case adalah deskripsi aktivitas yang menggambarkan bagaimana sistem merespons permintaan dari pengguna utama dalam berbagai situasi. Berikut komponen-komponen pada Use Case Diagram antara lain:

1. Aktor

Merupakan entitas yang melakukan kegiatan dengan sistem. Pengguna dapat berupa manusia, sistem, atau perangkat yang memainkan peran penting dalam menjalankan operasi sistem.

2. *Use Case*

   Merupakan representasi fungsional dari sistem, menjelaskan kegiatan aktor terhadap sistem.

3. Sistem

   Program membatasi sejauh mana aktor yang memanfaatkan kerangka (di luar kerangka) dan unsur-unsur yang harus tersedia (di dalam kerangka).

4. *Association*

   Merupakan hubungan antara aktor dan use case, dan menggambarkan partisipasi aktor.

5. *Include*

   Merupakan hubungan yang membutuhkan unsur lain untuk menyudahi tugas.

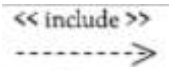
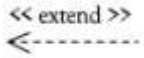
6. *Exclude*

   Merupakan hubungan antar unsur yang bisa menghapus salah satu unsur untuk menyudahi tugas.

Elemen-elemen ini dimanfaatkan sebagai penggambaran kegiatan aktor dan sistem pada program. Dengan menggunakan Diagram Use Case, pengguna dapat memahami operasi sistem dan keterlibatan aktor dalam sistem tersebut.

Tabel 2. 1 Komponen Use Case Diagram

Gambar	Nama Notasi	Keterangan
	Aktor	Pihak-pihak yang terlibat dalam sistem

	*Use Case*	Mewakili kegunaan utama sistem. Dapat memperluas (*Extend*) atau menyertakan (*Include*). Diletakkan dalam batas program dan dilabel kan dengan kata kerja/benda.
	*Boundary*	Digunakan untuk menggambarkan ruang lingkup sistem. Kotak dengan nama subjek di dalam atau di atasnya.
	Association	Menghubungkan aktor dengan kasus penggunaan. Menunjukkan bagaimana aktor berinteraksi dengan kasus penggunaan.
	*Include*	Menggambarkan perluasan kasus penggunaan untuk fungsi opsional. Dapat digambarkan sebagai panah dengan label <<*extend*>> dari kasus penggunaan ekstensi ke kasus penggunaan dasar.
	*Extend*	Menggambarkan perluasan kasus penggunaan untuk fungsi opsional. Dapat digambarkan sebagai panah dengan label <<*extend*>> dari kasus penggunaan ekstensi ke kasus penggunaan dasar.

(Sumber : contoh tabel use diagram, Michael Elkan, 2022)

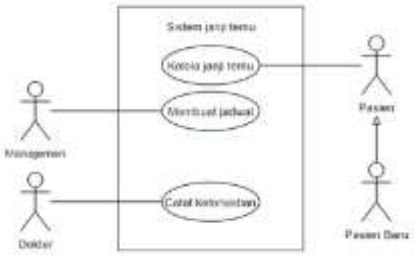
Berikut adalah penjelasan dari tabel di atas:

a. *Actor*

Actor dalam diagram use case merupakan representasi dari peran atau perangkat dalam lingkungan yang berinteraksi dengan sistem, bukan pengguna individu. *Actor* bisa mewakili pengguna sistem atau sistem lain yang berinteraksi dengan sistem saat ini. Seperti yang dijelaskan pada tabel 2.4, *Actor* bisa mengisi *form* pada sistem, menerima hasil atau melakukan keduanya.

b. *Use* Case

Mewakili fungsionalitas utama sistem. Dapat memperluas (Extend) atau menyertakan (Include) kasus penggunaan lain. Diletakkan di dalam lingkup sistem dan diberi judul dengan frasa kata kerja/ benda.



Gambar 2. 2 contoh use case diagram sistem janji temu

(Dennis, Wixom, & Tegarden, 2015)

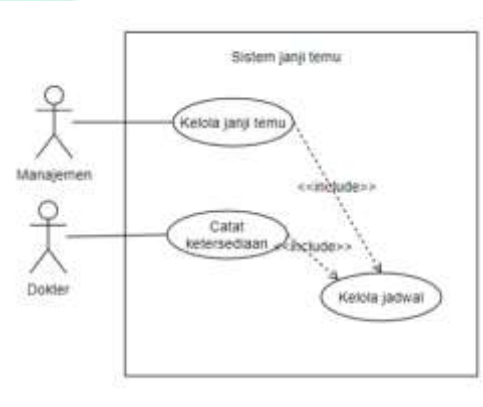
c. *Boundary* atau batasan subjek dalam *use case* diagram merupakan sebuah notasi yang dimodel kan dengan bentuk kotak, digunakan untuk menggambarkan ruang lingkup sistem dengan jelas, baik yang bersifat eksternal maupun internal. satu keputusan penting adalah menentukan batasan subjek. Hal tersebut dapat digunakan untuk memisahkan perangkat lunak dari lingkungan eksternal nya, memisahkan sub sistem dalam perangkat lunak, atau bahkan memisahkan proses individual dalam sistem. Batasan subjek juga dapat digunakan untuk membedakan system informasi, seperti perangkat lunak dan aktor internal, dari lingkungannya.

d. *Association*

Hubungan antara kasus penggunaan dan aktor dalam diagram Kasus Penggunaan dijelaskan dengan menggunakan istilah "hubungan." Hubungan ini menggambarkan cara Kasus Penggunaan berinteraksi dengan aktor, yang menghubungkan aktor dengan kasus penggunaan mencerminkan hubungan, yang biasanya menunjukkan pertukaran informasi dua arah antara Kasus Penggunaan dan aktor. Apabila pertukaran informasi hanya satu arah, kita dapat menggunakan panah penuh untuk menunjukkan arah aliran informasi.

e. *Include Relationship*

Hubungan "*Include*" adalah sebuah jenis ikatan yang digunakan dalam diagram use case untuk menggambarkan bagaimana sebuah kasus penggunaan dapat mencakup atau memasukkan fungsionalitas dari kasus penggunaan lain. Ini berarti bahwa kasus penggunaan yang memiliki relasi "*Include*" akan selalu menjalankan kasus penggunaan yang tercakup dalam ikatan tersebut. Hubungan "*Include*" berguna ketika terdapat fungsi yang sering digunakan dan dapat diakses melalui beberapa kasus penggunaan yang berbeda. Dengan memanfaatkannya, kita dapat menghindari pengulangan kode dan menjaga keseragaman dalam sistem.



Gambar 2. 3 Contoh menggunakan Include Relationship

(Dennis, Wixom, & Tegarden, 2015)

Dari contoh tersebut, terlihat bahwa notasi include ditunjukkan dengan panah berlabel "<<include>>", dan use case yang termasuk ditempatkan di bawah use case yang memasukkannya. Dalam kasus ini, "Catat Ketersediaan" dan "Membuat Jadwal" memanfaatkan fitur yang disertakan dari "Kelola Jadwal".

f. *Extend Relationship*

Menurut penulisan yang dilakukan oleh (Dennis, Wixom, & Tegarden, 2015) *Extend Relationship* merupakan salah satu jenis asosiasi yang digunakan dalam *use case* sebagai penggambaran *use case* dapat memperluas atau menambah perilaku dari use case lain. Hubungan "Extend" digunakan untuk menjelaskan skenario opsional atau alternatif yang mungkin terjadi dalam suatu kasus penggunaan. Ini membantu dalam menjelaskan bagaimana suatu

kasus penggunaan dapat melengkapi atau memperluas kasus penggunaan lain dalam respons terhadap kondisi khusus.

Gambar 2. 4 Contoh penggunaan Extend Relationship (Dennis, (Wixom, & Tegarden, 2015)

Pada contoh gambar 2.4.3, jika kita perlu menambahkan fitur verifikasi atau pengaturan pembayaran dalam use case "Kelola Janji Temu," Oleh karena itu, kita dapat membuat sebuah use case terpisah yang disebut "Pengaturan Pembayaran" untuk memperluas fungsi dari "Kelola Janji Temu". Dalam diagram use case, hubungan ini biasanya digambarkan dengan panah yang bertulisan “<<extend>>" yang menghubungkan "Pengaturan Pembayaran" ke "Kelola Janji Temu." Selain itu, posisi vertikal dari use case dalam diagram dapat mencerminkan hierarki dan tingkat detail yang berbeda dalam fungsionalitas sistem, dengan beberapa use case berada dalam level hierarki tertentu.

Adapun menurut penulisan yang dilakukan oleh (Sukamto & Shalahuddin, 2018:155), cara menggambarkan use case adalah sebagai berikut.

a. Identifikasi tujuan sistem

Mengidentifikasi alasan pembuatan sistem adalah tahap pertama dalam membuat Use Case Diagram. Pengguna sistem juga akan dapat diidentifikasi dengan tujuan yang jelas.

b. Identifikasi aktor

Cari tahu siapa yang berpartisipasi dalam sistem. Individu atau organisasi yang berinteraksi dengan sistem dapat dianggap sebagai aktor.

c. Identifikasi *use case*

Tentukan kemampuan aktor. Semua hal ini disebut "use case", dan dibuat sesuai dengan tujuan dan kebutuhan sistem.

d. Gambarkan diagram

Gambarkan aktor dan use case dalam diagram. Simbol orang dan lingkaran mewakili aktor. Gunakan garis untuk menghubungkan aktor dengan use case untuk menggambarkan hubungan dan interaksi antara keduanya.

e. Tambahkan Detail

Tambahkan nama dan deskripsi use case ke *Use Case Diagram.*

2.2.5 *Database dan Database Management System* (DBMS)

Program yang berguna menata data pada penyimpanan. Dalam konteks sistem komputer, basis data adalah rangkaian data terstruktur dan dapat diakses oleh berbagai aplikasi atau pengguna. DBMS dan *Relational Database Management System* (RDBMS) merupakan program dengan tujuan dalam metode relasi. Dalam model ini, data disimpan pada tabel yang memiliki baris dan kolom.

Tiap tabel mempunyai kunci utama unik untuk mengidentifikasi tiap baris dengan spesifik. RDBMS menggunakan *Structured Query Language* untuk mengakses, menata, dan memanipulasi data pada penyimpanan. Sistem ini menawarkan keunggulan dalam keandalan, konsistensi, dan kemampuan untuk menjaga integritas data. RDBMS banyak digunakan dalam berbagai aplikasi, mulai dari sistem perbankan, *e-commerce,* hingga aplikasi perusahaan yang kompleks dan beragam, mulai dari sistem basis data sederhana untuk penggunaan perorangan hingga sistem basis data yang sangat canggih untuk organisasi besar dengan kebutuhan pemrosesan data yang kompleks dan skalabilitas yang tinggi.

2.2.6 *Relational Database Management System (RDBMS)*

RDBMS merupakan salah satu program mengatur basis data yang menyimpan data dalam tabel-tabel yang saling terkait. Data dalam tabel diatur dalam baris dan kolom, dengan tiap baris menaungi sebuah entri data di mana setiap kolom merepresentasikan atribut dari data tersebut. RDBMS menggunakan *Structured Query Language* guna menata dan melakukan *query* sebuah data. Dalam buku "Database System Concepts" edisi ke-7, Silberschatz, Korth, dan Sudarshan menjelaskan bahwa RDBMS adalah program untuk menegaskan, menciptakan, mengelola, serta memanipulasi basis data relasional. Sistem ini menggunakan SQL sebagai bahasa standar untuk mengakses dan memodifikasi data, serta memastikan bahwa transaksi basis data memenuhi properti ACID (Atomicity, Consistency, Isolation, Durability).

Karakteristik Utama RDBMS adalah :

1. Tables: Data disimpan dalam bentuk baris (rekaman) dan kolom (atribut).
2. Primary Keys: Kolom unik untuk mengidentifikasi setiap baris pada tabel.
3. Foreign Keys: salah satu kunci pada kolom yang digunakan untuk menghubungkan tabel-tabel yang berbeda.
4. SQL *(Structured Query Language)*: Bahasa yang dipakai untuk pengelolaan dan mengquery data pada RDBMS.
5. ACID *Properties*: Menjamin transaksi basis data yang aman dan andal dengan memastikan atomisitas, konsistensi, isolasi, dan daya tahan.

2.2.7 Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa mengetahui struktur internal atau kode sumbernya. Pendekatan ini sangat penting dalam memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna. Pengujian dilakukan dengan memberikan input tertentu dan memeriksa output yang dihasilkan, tanpa memperhatikan bagaimana hasil tersebut diperoleh.

Teknik pengujian dalam black box testing meliputi berbagai metode seperti Equivalence Class Partitioning, yang membagi domain input menjadi kelas-kelas data untuk mengidentifikasi kasus uji yang representatif; Boundary Value Analysis (BVA), yang fokus pada pengujian batas input untuk menemukan kesalahan pada titik ekstrem; State Transition Testing, yang menguji perubahan status aplikasi berdasarkan input yang berbeda; dan Cause-Effect Graphing, yang menggambarkan hubungan sebab-akibat antara kondisi input dan hasil yang diharapkan (Serupa.id, 2020).

2.3 Tinjauan Studi

1. Jurnal yang ditinjau adalah **"Perancangan Aplikasi Perangkat Bergerak Rekomendasi *Coworking Space* di Kota Malang Berbasis Android"** Asyhar, K. (2019) tentang Memahami Coworking Space (Ruang Kerja

Bersama) Sebagai Konsep Baru Tempat Bekerja (Studi Pada Coworking Space di Kota Malang). Jurnal Ilmiah Mahasiswa Fakultas Ekonomi dan Bisnis Universitas Brawijaya, 7(2) ini diambil kembali dari https://jimfeb.ub.ac.id/index.php/jimfeb/article/view/5899 oleh Dwi Ayu Permata Sari pada tahun 2021 ini membahas tentang pengembangan aplikasi Android yang dirancang untuk memberikan rekomendasi *coworking space* di Kota Malang. Aplikasi ini menggunakan dua metode utama, yaitu TOPSIS dan Haversine, untuk memberikan rekomendasi terbaik bagi pengguna. Proses bisnis dimulai dengan pengguna mengunduh aplikasi dan mendaftar dengan mengisi data pribadi. Setelah login, pengguna dapat memasukkan preferensi seperti lokasi, fasilitas yang diinginkan, harga, dan jarak maksimal. Aplikasi ini mengumpulkan data *coworking spac*e yang tersedia dan menggunakan metode Haversine untuk menghitung jarak antara lokasi pengguna dan setiap coworking space. Kemudian, metode TOPSIS diterapkan untuk menilai dan memberi peringkat pada setiap *coworking space* berdasarkan preferensi pengguna. Hasilnya, aplikasi menampilkan daftar rekomendasi yang paling sesuai. Pengguna dapat melihat detail *coworking space*, termasuk fasilitas, harga, dan jarak, serta melakukan pemesanan dan pembayaran online yang aman. Setelah menggunakan *coworking space*, pengguna dapat memberikan feedback dan ulasan, yang membantu meningkatkan kualitas rekomendasi di masa depan dan memberikan informasi tambahan bagi pengguna lainnya. Aplikasi ini tidak hanya membantu pengguna menemukan *coworking space* terbaik, tetapi juga memfasilitasi seluruh proses dari pencarian hingga pemesanan dan pembayaran.

2. Tinjauan studi dari jurnal yang berjudul **"Sistem Informasi dan Pemesanan *Coworking Space* Berbasis Web (Studi Kasus EZO Space Malang)"** yang ditulis oleh M. Ridwan pada tahun 2020 ini mengulas perancangan sistem informasi dan pemesanan *coworking space* berbasis web untuk EZO Space Malang. Sistem ini dirancang untuk memungkinkan pengguna mencari, memesan, dan membayar *coworking space* secara online. Pengguna memulai dengan mendaftar di situs web dan login

menggunakan akun yang terdaftar. Setelah masuk, pengguna dapat melakukan pencarian *coworking space* dengan kriteria seperti lokasi, jenis ruangan, dan fasilitas. Sistem mengumpulkan data yang sesuai dengan kriteria pencarian dan menampilkan daftar *coworking space* beserta detailnya. Pengguna dapat melihat informasi lengkap seperti fasilitas, harga, dan ketersediaan, dan melakukan pemesanan langsung melalui situs web. Sistem pembayaran online yang terintegrasi memudahkan transaksi yang cepat dan aman. Setelah pembayaran, pengguna menerima konfirmasi pemesanan. Pengelola *coworking space* dapat mengelola reservasi dan memastikan ketersediaan ruangan. Pengguna dapat memberikan feedback dan ulasan setelah menggunakan *coworking space*, yang membantu meningkatkan kualitas layanan dan memberikan informasi tambahan bagi pengguna lainnya. Sistem ini memfasilitasi seluruh proses dari pencarian hingga pembayaran dan pengelolaan reservasi dengan efisiensi tinggi.

3. Tinjauan studi dari jurnal **"Aplikasi *Mobile* Pencarian dan Pemesanan *Coworking Space* Berbasis Android"** yang ditulis oleh Muhammad Iqbal Ghifari pada tahun 2019 ini mengulas perancangan aplikasi Android untuk mencari dan memesan coworking space. Aplikasi ini memanfaatkan Google Maps untuk menampilkan lokasi coworking space dan menyediakan fitur filter untuk membantu pengguna menemukan coworking space yang sesuai dengan kebutuhan mereka. Proses dimulai dengan pengguna mengunduh aplikasi dan mendaftar dengan mengisi data pribadi, lalu login menggunakan akun yang terdaftar. Setelah masuk, pengguna dapat memasukkan kriteria pencarian seperti lokasi, fasilitas, harga, dan jenis ruangan. Aplikasi mengumpulkan data yang sesuai dan menampilkan lokasi coworking space menggunakan Google Maps. Pengguna dapat melihat daftar coworking space beserta detail seperti fasilitas, harga, dan jarak dari lokasi pengguna. Pengguna dapat melakukan pemesanan langsung melalui aplikasi, memilih tanggal dan durasi penggunaan, dan menyelesaikan transaksi dengan sistem pembayaran online yang aman. Setelah pembayaran, pengguna menerima konfirmasi pemesanan dan dapat menggunakan coworking space sesuai jadwal. Pengguna dapat memberikan

feedback dan ulasan setelah menggunakan coworking space, yang membantu meningkatkan kualitas rekomendasi di masa depan. Aplikasi ini memfasilitasi seluruh proses dari pencarian hingga pemesanan dan pembayaran, membantu pengguna menemukan coworking space yang paling sesuai dengan kebutuhan mereka.

4. Jurnal yang ditinjau dari **"Pemanfaatan Aplikasi *mobile coworking space* untuk Meningkatkan Produktivitas Kerja Karyawan di Era Pandemi Covid-19"** yang ditulis oleh Dimas Bayu Saputra pada tahun 2021 mengevaluasi manfaat aplikasi *mobile coworking space* dalam meningkatkan produktivitas kerja karyawan selama pandemi COVID-19. Aplikasi ini dirancang untuk memungkinkan karyawan bekerja dari mana saja dan tetap terhubung dengan rekan kerja secara online. Proses bisnis dimulai dengan karyawan mengunduh aplikasi dan mendaftar dengan memasukkan data pribadi, lalu login menggunakan akun yang terdaftar. Setelah masuk, karyawan dapat mencari *coworking space* sesuai preferensi seperti lokasi, fasilitas, dan harga. Aplikasi mengumpulkan data dan menampilkan informasi detail seperti fasilitas, harga, dan jarak. Karyawan dapat memilih *coworking space* yang diinginkan dan melakukan pemesanan langsung melalui aplikasi, memilih tanggal dan durasi penggunaan. Sistem pembayaran online yang terintegrasi memudahkan transaksi yang cepat dan aman. Setelah pembayaran, karyawan menerima konfirmasi pemesanan dan dapat menggunakan *coworking space* sesuai jadwal. Aplikasi ini juga menyediakan fitur komunikasi dan kolaborasi online, memungkinkan karyawan tetap produktif meskipun bekerja dari lokasi berbeda. Karyawan dapat memberikan feedback dan ulasan setelah menggunakan *coworking space*, yang membantu meningkatkan kualitas layanan. Aplikasi ini mendukung produktivitas dan kolaborasi di era pandemi, memfasilitasi seluruh proses dari pencarian hingga pemesanan dan pembayaran.