

BAB II

LANDASAN TEORI

2.1 Teori Dasar

2.1.1 Pengertian Analisis dan Perancangan

Analisis adalah aktivitas berpikir yang dilakukan untuk menguraikan atau memecahkan suatu masalah dari bagian yang besar menjadi bagian-bagian terkecil (Febriana, 2020).

Analisis sistem merupakan metode pengetahuan dan penetapan secara menyeluruh tentang apa saja yang harus dicapai oleh sistem informasi (Tiara Waty S et al., 2021).

Dalam melakukan analisis sistem terdapat langkah-langkah sebagai berikut :

1. *Identify* : dilakukan untuk mengidentifikasi masalah yang ada.
2. *Understand* : proses memahami kerja dari sistem yang ada.
3. *Analyze* : fase menganalisis sistem.
4. *Report* : fase terakhir yaitu membuat laporan dari hasil analisis yang telah dilakukan.

Perancangan merupakan suatu cara yang digunakan untuk menciptakan dan mendesain sistem yang baru (Nur & Suyuti, 2017).

Perancangan sistem adalah gabungan dari beberapa kegiatan yang mendeskripsikan dengan detail bagaimana sistem akan bekerja (Hidayat et al., 2022).

Analisis dan perancangan sistem dapat disimpulkan sebagai aktivitas berpikir yang dilakukan untuk menguraikan masalah dari bagian besar menjadi bagian kecil untuk menetapkan secara menyeluruh tentang apa saja yang harus dicapai oleh sistem informasi. Untuk mencapai tujuan yang diharapkan maka dibuatlah metode untuk membuat desain sistem yang baru yang berisikan gabungan dari beberapa kegiatan yang mendeskripsikan secara detail bagaimana sistem baru tersebut akan bekerja.

2.1.2 Pengertian Sistem

Sistem memanifestasikan sekelompok komponen yang saling berkaitan dan saling berkolaborasi demi meraih suatu tujuan yang dikehendaki (Satzinger et al., 2016).

2.1.3 Karakteristik Sistem

Agar dapat dikategorikan sebagai sistem ada beberapa karakteristik yang perlu dipenuhi. Berikut adalah karakteristik dari sistem (Fahlevi & Erlansyah, 2022) :

1. Komponen (*Component*)

Suatu sistem haruslah memiliki beberapa komponen atau unit-unit tersendiri maupun komponen atau unit-unit yang akan terintegrasi dengan sistem.

2. Batasan sistem (*Boundary*)

Boundary merupakan batasan sistem yang harus terpisah dengan sistem lain atau lingkungan di luar sistem.

3. Lingkungan luar sistem (*Environment*)

Environment merupakan lingkungan di luar sistem yang tidak berkaitan dengan bagian dalam sistem.

4. Penghubung sistem (*Interface*)

Interface atau disebut juga *conector* merupakan penghubung elemen luar dengan sistem.

5. Masukan sistem (*Input*)

Input adalah masukan yang akan dikelola oleh sistem.

6. Proses (*Process*)

Sistem harus memiliki unit pengolahan untuk mengelola *input* dan menghasilkan *output*.

7. Keluaran sistem (*Output*)

Hasil yang di dapat dari unit pengolahan atau proses.

8. Sasaran sistem (*Objective*)

Sistem haruslah mempunyai suatu tujuan yang ingin di capai.

2.1.4 Klasifikasi Sistem

Sistem dapat diklasifikasikan berdasarkan persamaan dan perbedaannya, berikut adalah klasifikasi sistem menurut (Fahlevi & Erlansyah, 2022) :

1. Sistem diklasifikasikan sebagai :

1. Sistem Abstrak

Berisikan sebuah gagasan atau konsep. Contoh : sistem teologi yang berisikan gagasan hubungan antara manusia dan tuhan.

2. Sistem Fisik

Sistem yang wujudnya dapat dilihat. Contoh : sistem komputer.

2. Sistem diklasifikasikan sebagai :

1. Sistem Deterministik

Sistem yang dapat diprediksi secara tepat operasinya. Contoh : sistem komputer.

● 2. Sistem Probabilistik

Sistem yang tidak dapat diprediksi secara tepat operasinya. Contoh : arisan.

3. Sistem diklasifikasikan sebagai :

1. Sistem Terbuka

Sistem yang tidak melakukan pertukaran materi, energi dan informasi lingkungan. Contoh : reaksi kimia pada tabung yang terisolasi.

2. Sistem Tertutup

Sistem yang dipengaruhi oleh lingkungan dan berhubungan dengan lingkungan tersebut. Sistem ini dapat menerima masukan yang diketahui dan bersifat acak. Contoh : perusahaan dagang.

2.1.5 Pengertian Informasi

Informasi merupakan hasil yang didapat dari pemrosesan data. Data yang dihasilkan merupakan data yang bermakna dan memiliki manfaat bagi pengguna data (Putra, 2020). Sementara (Maydianto & Rasid, 2021) mengungkapkan bahwa Informasi adalah data yang telah diolah kembali menjadi sesuatu yang bernilai tinggi bagi penerima data dan akan digunakan sebagai dasar pembuatan keputusan.

Berdasarkan pendapat dari penelitian di atas mengenai pengertian informasi dapat disimpulkan bahwa informasi adalah hasil dari pemrosesan data yang sangat penting bagi pengguna data untuk membuat suatu keputusan berdasarkan data yang diterimanya. Maka dari itu kualitas dari informasi sangatlah penting. Kualitas sebuah informasi dipengaruhi oleh tiga hal (Tajul et al., 2020) yaitu :

1. Akurat (*Accuracy*)

Informasi bebas dari kesalahan, dapat digunakan sesuai kebutuhan dan kondisi serta dapat disajikan secara lengkap.

2. Tepat Waktu (*Timeliness*)

Proses pengolahannya dapat diselesaikan tepat waktu agar dapat menghasilkan laporan yang tepat waktu.

3. Relevan

- Informasi yang disajikan sesuai dengan kenyataan.

2.1.6 Pengertian Sistem Informasi

Sistem Informasi merupakan kumpulan komponen yang saling berkaitan satu dengan yang lain untuk mencapai suatu tujuan (Maydianto & Rasid, 2021).

Menurut (Seah et al., 2020) sistem informasi adalah gabungan dari berbagai komponen dari teknologi informasi yang saling berkaitan untuk menghasilkan sebuah informasi dan memperoleh jalur komunikasi dalam organisasi.

Berdasarkan pendapat ahli di atas dapat disimpulkan bahwa sistem informasi adalah komponen-komponen yang saling berkaitan dan digunakan untuk menghasilkan informasi dan jalur komunikasi organisasi.

Menurut (Seah et al., 2020) komponen-komponen dari sistem disebut dengan blok bangunan (*building block*) yang diuraikan sebagai berikut :

1. Blok masukan (*input block*)

Blok yang bertugas dalam bagian input data. Memastikan data masuk ke dalam sistem informasi dan merekam data apa saja yang dimasukkan. Data yang dimasukkan berupa dokumen- dokumen dasar.

2. Blok model (*model block*)

Terbentuk dari kombinasi prosedur, logika dan model matematik yang

memproses data masukan (*input*) dan data yang tersimpan di basis data dengan cara yang telah diatur agar mendapatkan keluaran (*output*) yang sesuai.

3. Blok keluaran (*output block*)

Menghasilkan informasi yang berkualitas dan berguna untuk para pengguna sistem.

4. Blok teknologi (*technology block*)

Blok teknologi digunakan untuk menerima data, menjalankan model, menyimpan data, mengakses data, mengelola keluaran (*output*) berupa informasi yang berkualitas dan berguna bagi pengguna sistem. Blok teknologi dibagi menjadi dua bagian yaitu bagian perangkat keras (*hardware*) dan bagian perangkat lunak (*software*).

5. Blok basis data (*database block*)

- Media yang disediakan untuk menyimpan data yang saling berkaitan satu sama lain. Tersimpan dalam perangkat keras (*hardware*) dan dimanipulasi dalam perangkat lunak (*software*).

6. Blok kendali (*control block*)

Blok untuk menanggulangi gangguan-gangguan dalam sistem.

2.1.7 Pengertian Penjualan

Penjualan adalah proses administratif sosial dengan kondisi individu dan kelompok mendapatkan apa yang mereka butuhkan dan inginkan, menciptakan, menawarkan, dan mempertukarkan produk yang bernilai dengan pihak lain (Iskandar, 2021).

Menurut (S. Setiawan, 2019) penjualan adalah proses yang terjadi ketika penjual dan pembeli melakukan pertukaran barang. Tujuan dari penjualan itu sendiri sebagai berikut :

1. Mencapai angka penjualan tertentu.
2. Mendapatkan keuntungan atau laba.
3. Membantu pertumbuhan perusahaan.

4. Mendapatkan persona positif terhadap produk yang dijual perusahaan.
5. Menentukan target pasar yang pas dan sesuai dengan kebutuhan pelanggan.

Tentunya untuk mencapai tujuan di atas bukan hanya menjadi tugas dan tanggung jawab para tenaga penjual di perusahaan. Diperlukan kolaborasi dari beberapa pihak diantaranya fungsionaris dari perusahaan yang menyediakan dana seperti bagian keuangan, bagian produksi yang menghasilkan produk, bagian personalia sebagai penyedia SDM, bagian personalia yang menyediakan tenaga kerja, bagian marketing sebagai pemasaran produk dan pengumpulan data konsumen agar menjadi konsumen potensial bagi perusahaan.

2.1.8 Jenis Dan Bentuk Penjualan

Menurut (S. Setiawan, 2019) jenis- jenis penjualan sebagai berikut :

1. Trade selling

- Penjualan yang terjadi bila pengecer dipersilakan untuk melakukan perbaikan jaringan distribusi produk produsen dan pedagang besar. Hal ini melibatkan penyalur melakukan promosi, persediaan dan produk baru. Pada dasarnya *trade selling* adalah penjual menjual barangnya melalui penyalur sebelum ke konsumen.

2. Missionary selling

Penjualan yang terjadi pada produsen yang sudah memiliki penyalur untuk produknya sendiri. *Missionary selling* mengarahkan pembeli untuk melakukan pembelian produk pada penyalurnya.

3. Technical selling

Penjualan yang di dalamnya penjual melakukan rekomendasi, saran dan nasihat dari produk yang dijualnya dengan tujuan agar pembeli mendapatkan produk yang sesuai.

4. New bussines selling

Penjualan yang menggunakan teknik mencari pembeli baru dengan tujuan menaikkan penjualan. Seperti yang dilakukan perusahaan asuransi.

5. Responsive selling

Setiap tenaga penjual dapat memberikan respons terhadap permintaan konsumen dengan *route driving and retailing*.

Menurut (S. Setiawan, 2019) bentuk-bentuk penjualan sebagai berikut :

1. Penjualan tunai

Penjualan yang begitu sudah ada kesepakatan harga antara penjual dan pembeli, maka pembeli melakukan pembayaran secara tunai.

2. Penjualan kredit

Penjualan yang sistem pembayarannya dapat diangsur dengan jangka waktu tertentu.

3. Penjualan secara tender

Penjualan yang dilakukan secara lelang. Penjual akan mempresentasikan produknya kepada pembeli, kemudian pembeli akan memilih produk terbaik yang diinginkan.

4. Penjualan ekspor

- Penjualan yang dilakukan dengan pembeli luar negeri yang mengimpor barang dan biasanya menggunakan fasilitas *letter of credit*.

5. Penjualan secara konsinyasi

Penjualan yang dilakukan dengan sistem titip barang kepada pembeli yang nanti pembeli tersebut juga akan menjual barangnya, namun hanya barang yang terjual saja yang perlu pembeli bayar. Jika ada barang yang tidak laku maka akan dikembalikan pada penjual.

6. Penjualan secara grosir

Penjualan barang yang tidak langsung kepada pembeli tetapi melalui mediator. Biasanya melakukan pembelian dalam jumlah banyak.

2.1.9 Pengertian Sistem Informasi Penjualan

Menurut (Purwaningtias & Agustini, 2022) sistem informasi penjualan adalah sistem yang menjembatani penjual dan pembeli menggunakan beberapa metode dan juga menggunakan fitur pendukung, berupa sistem yang dapat membuat informasi pesanan, membuat catatan serta perhitungan penjualan dan menghasilkan informasi penjualan.

2.1.10 Aplikasi Berbasis Web

Website adalah sekumpulan halaman web yang mempunyai pokok saling berhubungan antara satu halaman dengan yang lainnya, biasanya ditempatkan pada sebuah server web yang dapat di akses melalui jaringan internet maupun LAN atau jaringan wilayah lokal (Wijayanti, 2022).

Aplikasi berbasis web adalah sebuah sistem perangkat lunak yang mengikuti teknologi dan standar World Wide Web Consortium (W3C). Menyediakan sumber daya web spesifik seperti konten dan layanan melalui sebuah *user interface* dan browser web (Hatmoko, 2019).

2.1.11 Metode Pengembangan Sistem *Waterfall*

Merupakan model pengembangan sistem tradisional yang memiliki perencanaan pengembangan sistem secara tertata atau dilakukan sesuai urutannya (Wahid, 2020). Sementara menurut (Kadim et al., 2022) metode pengembangan sistem *waterfall* adalah metode pengembangan sistem yang mempunyai alur seperti air terjun karena proses pengerjaannya yang berurutan dari atas kebawah.

Dapat disimpulkan bahwa metode pengembangan sistem *waterfall* adalah metode yang digunakan dalam merancang perangkat lunak (*software*) dengan sistem pengerjaan yang berurutan dari proses yang paling atas hingga proses yang paling bawah sesuai dengan urutannya.

Berikut adalah beberapa tahapan dalam merancang sistem menggunakan metode *waterfall* menurut (Novitasari, 2020):

1. *Requirement Analyst*

Tahapan yang dilakukan untuk mengetahui kebutuhan sistem. Pada tahap ini dilakukan proses wawancara ataupun diskusi untuk menentukan kebutuhan sistem yang di harapkan.

2. *Design*

Tahap yang menggambarkan desain sistem usulan dan proses pengerjaannya.

3. *Implementation*

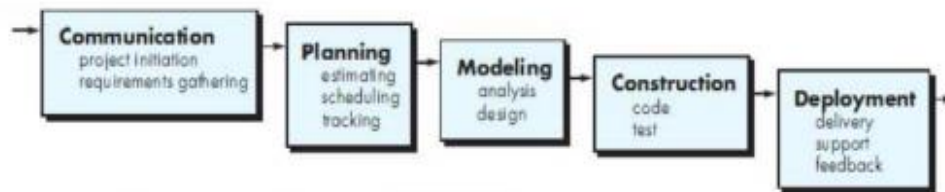
Proses untuk realisasi program menggunakan bahasa pemrograman untuk mendefinisikan kebutuhan yang telah didapat kedalam *software*.

4. *Testing*

Tahapan yang digunakan untuk menguji program. *Testing* dilakukan untuk mencari apakah masih ditemukan kesalahan pada program sebelum digunakan.

5. *Maintenance*

Fase pemeliharaan program yang berupa perbaikan kesalahan, perbaikan unit sistem, dan peningkatan pada sistem sesuai kebutuhan.



Gambar 2. 1 Ilustrasi Model *Waterfall* (Pressman, 2015, p. 42)

2.1.12 *Rich Picture*

Menurut (Barbrook-Johnson & Penn, 2022) *rich picture* merupakan metodologi yang digunakan untuk menggambarkan ide, permasalahan atau suatu konsep. *Rich picture* menyajikan gambaran umum suatu topik dan memaparkan hubungan yang saling berkaitan antara elemen-elemennya, identifikasi aktor dan aktivitas utama dengan jelas.

Rich picture merupakan sarana yang digunakan untuk menggambarkan keadaan yang terjadi dan permasalahan yang timbul di dalamnya. *Rich picture* dibuat dengan gambar yang sederhana agar mudah dipahami baik oleh pelukis ataupun orang yang melihatnya (Bell et al., 2019).

Berdasarkan pendapat ahli di atas dapat disimpulkan bahwa *rich picture* adalah metodologi yang digunakan untuk menggambarkan situasi, ide, permasalahan dan konsep. *Rich picture* memberikan gambaran umum topik dan hubungan antara elemennya, identifikasi aktor dan aktivitas utama dengan gambaran yang mudah dipahami.

2.1.13 *Flowchart*

Menurut (Hanief et al., 2019) *flowchart* merupakan teknik yang digunakan untuk menggambarkan urutan logika prosedur penyelesaian masalah. Dapat dikatakan bahwa *flowchart* merupakan urutan penyelesaian masalah yang

digambarkan dalam bentuk simbol-simbol tertentu. Penggunaan *flowchart* bermanfaat untuk dokumentasi program dan juga sebagai media komunikasi. *Flowchart* bertujuan mempresentasikan langkah-langkah penguraian masalah dengan cara yang sederhana, terurai dan rapi.

1. Menurut (Yuniansyah, 2020) berikut adalah jenis-jenis dari *flowchart* :

1. *Flowchart* sistem

Flowchart yang menggambarkan alur kerja secara keseluruhan, apa yang sedang dikerjakan pada sistem dan urutan dari prosedur yang ada di dalam sistem. Dapat di katakan bahwa *flowchart* sistem adalah urutan prosedur yang membentuk sistem.

2. *Flowchart* dokumen

Flowchart dokumen disebut juga dengan *flowchart paperwork* menggambarkan proses dari formulir ke laporan-laporan yang digunakan.

3. *Flowchart* skematik

Hampir sama dengan *flowchart* sistem yang membedakan adalah cara penggambaran *flowchart* skematik menggunakan gambar dokumen, peralatan komputer, dan peralatan pendukung lainnya untuk memudahkan pengguna dalam memahami alur prosesnya.

4. *Flowchart* program

Flowchart yang menggambarkan secara rinci langkah-langkah yang ada pada program.

5. *Flowchart* proses

Flowchart yang menggambarkan secara detail sebuah proses dengan cara memecah dan menganalisa setiap proses dan langkah pada suatu sistem.

2. Menurut (R. Setiawan, 2021) berikut adalah fungsi *flowchart* :

1. Mengembangkan proyek baru

Flowchart dapat digunakan untuk mengembangkan proyek baru. Seperti mendesain pekerjaan dan metode yang digunakan untuk menganalisa kepastian secara bersamaan.

2. Mengatur alur kerja

Flowchart merupakan pengatur alur kerja yang paling strategi.

3. Memodelkan proses bisnis




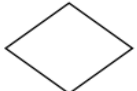
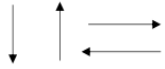

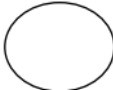


Flowchart dapat memodelkan proses bisnis dari yang paling sederhana hingga yang paling rumit. *Flowchart* digunakan untuk melakukan penyampaian perkiraan hingga kepastian hasil dari proses bisnis.

4. Mengaudit proses

Penggunaan *flowchart* dapat digunakan untuk melihat kesalahan pada proses yang berjalan.

Berikut adalah simbol-simbol yang digunakan dalam *flowchart* :

Tabel 2. 1 Tabel simbol-simbol *flowchart*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Terminator</i>	Merupakan simbol awal dan akhir kegiatan pada <i>flowchart</i> .
2		<i>Process</i>	Digunakan sebagai penghubung halaman.
3		<i>Input-Output</i>	Menggambarkan proses input dan out dan tidak tergantung pada jenis peralatannya.
4		<i>Decision</i>	Menggambarkan kondisi yang menghasilkan beberapa kemungkinan.
5		Simbol arus	Penghubung antar simbol dan menyatakan arah arus proses.
6		<i>Off-Line Storage</i>	menyatakan penyimpanan data pada simbol ini.
7		<i>Connector</i>	simbol yang digunakan untuk masuk atau menyambung proses pada halaman yang sama.
8		Dokumen	menyatakan input yang didapat berasal dari dokume kertas.
9		Predefined	menggambarkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam Storage.

Sumber : Jhon W Satzinger, 2016

2.1.14 Object Oriented Analysis and Design (OOAD)

Menurut (Ristias et al., 2021) *Object Oriented Analysis and Design* (OOAD) adalah cara yang digunakan untuk menganalisa dan merancang sistem berbasis komputer dengan menggunakan pendekatan yang berorientasi objek.

Dalam tahapannya OOAD dibagi menjadi 2 yaitu *Object Oriented Analysis* (OOA) dan *Object Oriented Design* (OOD). *Object Oriented Analysis* (OOA) menerapkan teknik pemodelan objek yang menganalisis persyaratan fungsional untuk sistem, sedangkan *Object Oriented Design* (OOD) menjabarkan bentuk sistem berdasarkan hasil dari analisis OOA (*Object Oriented Analysis*). OOA berfokus pada “apa yang sistem butuhkan”, sedangkan OOD berfokus pada “apa yang sistem dapat lakukan”. Yang selanjutnya pemodelan berorientasi objek digambarkan dalam teknik *Unified Modeling Language* (UML).

2.1.15 Unified Modeling Language (UML)

Sejarah awal UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997. UML 1.0 memiliki banyak standar untuk tetap mempertahankan konsistensinya di semua diagram. Kemudian untuk lebih menyempurnakan prinsip metodologi tersebut, aturan kardinalitas dimasukkan dalam bahasa dan gugus tugas revisi dibuat. Akhirnya pada tahun 2005, UML 2.0 dirilis dan sebagian besarnya didasarkan pada pendekatan object-oriented. Untuk tetap mengikuti perkembangan yang ada, versi UML 2.x diperbarui dan disempurnakan.

Menurut (Ansori, 2022) UML (*Unified Modelling Language*) adalah sebuah standar bahasa yang digunakan untuk menggambarkan batasan sistem yang akan dibangun, membuat analisis dan desain sistem, dan arsitektur yang digunakan dalam pemrograman berorientasi objek. UML paling banyak digunakan pada dunia industri sekarang ini.

UML adalah sekumpulan diagram, susunan dan teknik yang digunakan untuk menggambarkan dan merancang program pada sebuah aplikasi yang menggunakan *object oriented programming* (Ritonga, 2018).

Dari pendapat ahli di atas dapat disimpulkan bahwa UML digunakan untuk mendefinisikan perancangan awal sistem yang akan dibangun. UML mempunyai banyak pemodelan karena satu model saja tidak cukup untuk menggambarkan sistem secara keseluruhan. Maka dibutuhkan lebih dari satu model yang saling berkaitan untuk merancang sistem menggunakan UML.

A. Use case Diagram

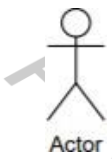
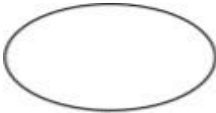

Perilaku suatu sistem yang memanifestasikan bentuk aktivitas dalam berbagai keadaan ketika permintaan dari pelaku utama direspon oleh sistem disebut dengan *use case*. Sistem akan merespon permintaan dari pelaku utama terkait suatu tujuan yang ingin dicapai (Ansori, 2022).


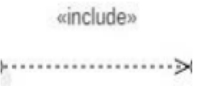

Penamaan *use case diagram* dibuat sesederhana mungkin dan tepat serta mudah dipahami. Dua hal utama pada saat mendefinisikan suatu *use case* adalah *actor* dan *use case*. Dengan pengertian sebagai berikut :

1. *Actor* melambangkan orang, proses, atau sistem yang berinteraksi dengan sistem informasi yang nantinya akan dibuat. Meskipun digambarkan orang, *actor* belum tentu selalu orang.
2. *Use case* merupakan fungsi dari sistem yang telah disediakan oleh sistem sebagai sarana bertukar pesan antar unit atau *actor*.

Berikut ini simbol-simbol yang digunakan dalam pembuatan *use case diagram*:

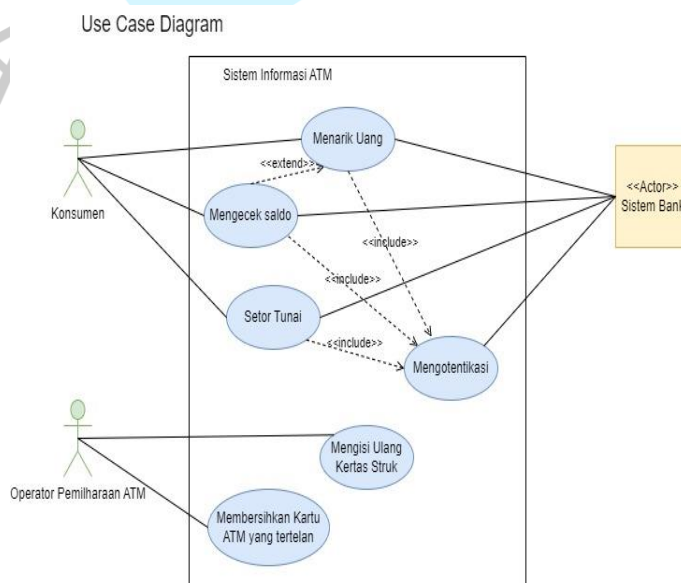
Tabel 2. 2 Tabel simbol-simbol *usecase diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Meskipun dilambangkan dengan orang, <i>actor</i> tidak selalu orang. Biasanya dinyatakan dengan kata benda.
2.		<i>Use Case</i>	Fungsi yang disediakan oleh sistem sebagai unit yang saling bertukar pesan antara unit atau <i>actor</i> . Biasanya dinyatakan dengan kata kerja.
3.		<i>Association</i>	Penghubung yang menggambarkan Komunikasi yang terjadi antara <i>actor</i> dengan <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .

4.		<i>Generalization</i>	Hubungan umum ke khusus. (generalisasi dan spesialisasi). Antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.		<i>Include</i>	Relasi tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsi.
6.		<i>Extend</i>	Relasi tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri. Konsepnya sama seperti inheritance pada oop. Biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.

Sumber : Jhon W Satzinger, 2016

Gambar 2.2 menggambarkan proses yang terjadi pada sistem informasi ATM. Konsumen sebagai *actor* yang dapat melakukan pengecekan saldo, menarik uang dan melakukan setor tunai. Penarikan uang dan pengecekan saldo dapat dilakukan sekaligus ataupun terpisah, maka dari itu notasi *use case* yang digunakan antara keduanya *extend*. Untuk melakukan penarikan uang, mengecek saldo dan melakukan setoran tunai harus melakukan proses otentikasi yang akan dilakukan oleh *actor* sistem bank. *Use case* menarik uang, mengecek saldo dan setor tunai tidak bisa berdiri sendiri harus melalui *use case* mengotentikasi maka dari itu notasinya *include*. Sementara *actor* operator pemeliharaan ATM dapat melakukan pengisian ulang kertas struk dan membersihkan kartu ATM konsumen yang tertelan mesin. Detail dari gambaran prosesnya dapat dilihat sebagai berikut :



Gambar 2. 2 Contoh Use Case Diagram (Slideplayer)

Use Case Description

Teks yang digunakan untuk berinteraksi dengan langkah-langkah terperinci antara pengguna dengan sistem (Satzinger et al., 2016). *Use case description* mempunyai dua kunci utama yaitu *use case* dan *scenario*. Tujuannya untuk memberikan gambaran umum tentang fungsionalitas proses bisnis yang melibatkan sistem di dalamnya.

1. *Use case name* : berisikan nama *use case* yang sedang dideskripsikan.
2. *Scenario* : penjelasan yang detail dan singkat dari *use case* yang sedang dideskripsikan.
3. *Triggering event* : digunakan untuk mengidentifikasi kegiatan apa yang dapat memicu *use case* berjalan.
4. *Brief description* : Penjelasan singkat mengenai *use case* yang akan dibahas.
5. *Actors* : siapa saja yang terlibat dalam *use case*.
6. *Related use case* : digunakan untuk mengidentifikasi *use case* lain yang terkait dengan *use case* yang sedang dibahas dan apa yang membuat berkaitan.
7. *Stakeholders* : memberikan informasi mengenai siapa saja yang membutuhkan informasi dari *use case*.
8. *Preconditions* : persyaratan yang dibutuhkan untuk memulai *use case*. Seperti objek-objek yang harus ada, informasi yang harus tersedia, dan aktor harus dalam keadaan seperti apa untuk memulai *use case*.
9. *Postconditions* : kondisi apa yang terjadi setelah *use case* dijalankan. Diantaranya informasi apa yang diperbaharui, objek apa yang terbentuk, ataupun bagaimana interaksi antar objek terjadi setelah *use case* selesai dijalankan.
10. *Flow of activities* : berisikan apa yang perlu dilakukan aktor untuk membuat *use case* berjalan dan sistem dapat memberikan respons balik kepada aktor.
11. *Exception Conditions* : berisikan identifikasi mengenai kejadian-kejadian apa saja yang membuat sistem tidak berjalan sesuai dengan yang telah direncanakan sehingga dibutuhkan cara lain untuk menyelesaikan *use case*.

Tabel 2. 3 Contoh *Use Case Description* (Jhon W Satzinger, 2016, p. 437)

<i>Use case name :</i>	<i>Create customer account.</i>	
<i>Scenario :</i>	<i>Create online customer account.</i>	
<i>Triggering event :</i>	<i>New customer wants to set up account online.</i>	
<i>Brief description :</i>	<i>Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.</i>	
<i>Actors :</i>	<i>Customer.</i>	
<i>Related use cases :</i>	<i>Might be invoked by the Check out shopping cart use case.</i>	
<i>Stakeholders :</i>	<i>Accounting, Marketing, Sales.</i>	
<i>Preconditions :</i>	<i>Customer account subsystem must be available.</i>	
	<i>Credit/debit authorization services must be available.</i>	
<i>Postconditions :</i>	<i>Customer must be created and saved.</i>	
	<i>One or more Addresses must be created and saved.</i>	
	<i>Credit/debit card information must be validated.</i>	
	<i>Account must be created and saved.</i>	
	<i>Address and Account must be associated with Customer.</i>	
<i>Flow of activities :</i>	<i>Actor</i>	<i>System</i>
	<ol style="list-style-type: none"> <i>1. Customer indicates desire to create customer account and enters basic customer information.</i> <i>2. Customer enters one or more addresses.</i> <i>3. Customer enters credit/debit card information.</i> 	<ol style="list-style-type: none"> <i>1.1 System creates a new customer.</i> <i>1.2 System prompts for customer addresses.</i> <i>2.1 System creates addresses.</i> <i>2.2 System prompts for credit/debit card.</i> <i>3.1 System creates account.</i> <i>3.2 System verifies authorization for credit/debit card.</i> <i>3.3 System associates customer, address, and account.</i> <i>3.4 System returns valid customer account details.</i>
<i>Exception conditions:</i>	<i>1.1 Basic customer data are incomplete.</i>	
	<i>2.1 The address isn't valid.</i>	
	<i>3.1 Credit/debit information isn't valid.</i>	





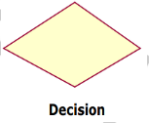


Sumber : *Jhon W Satzinger, 2016*

B. Activity Diagram

Activity diagram adalah diagram yang menerangkan *workflow* (aliran kerja) atau kegiatan dari sebuah sistem, proses bisnis atau menu yang ada pada sebuah perangkat lunak. Penting untuk diketahui bahwa *activity diagram* menerangkan kegiatan yang dilakukan oleh sistem bukan kegiatan yang dilakukan oleh aktor, jadi kegiatan atau aktivitas yang dilakukan adalah aktivitas atau kegiatan sistem (Ansori, 2022).

Berikut adalah komponen-komponen yang terdapat pada *activity diagram*:

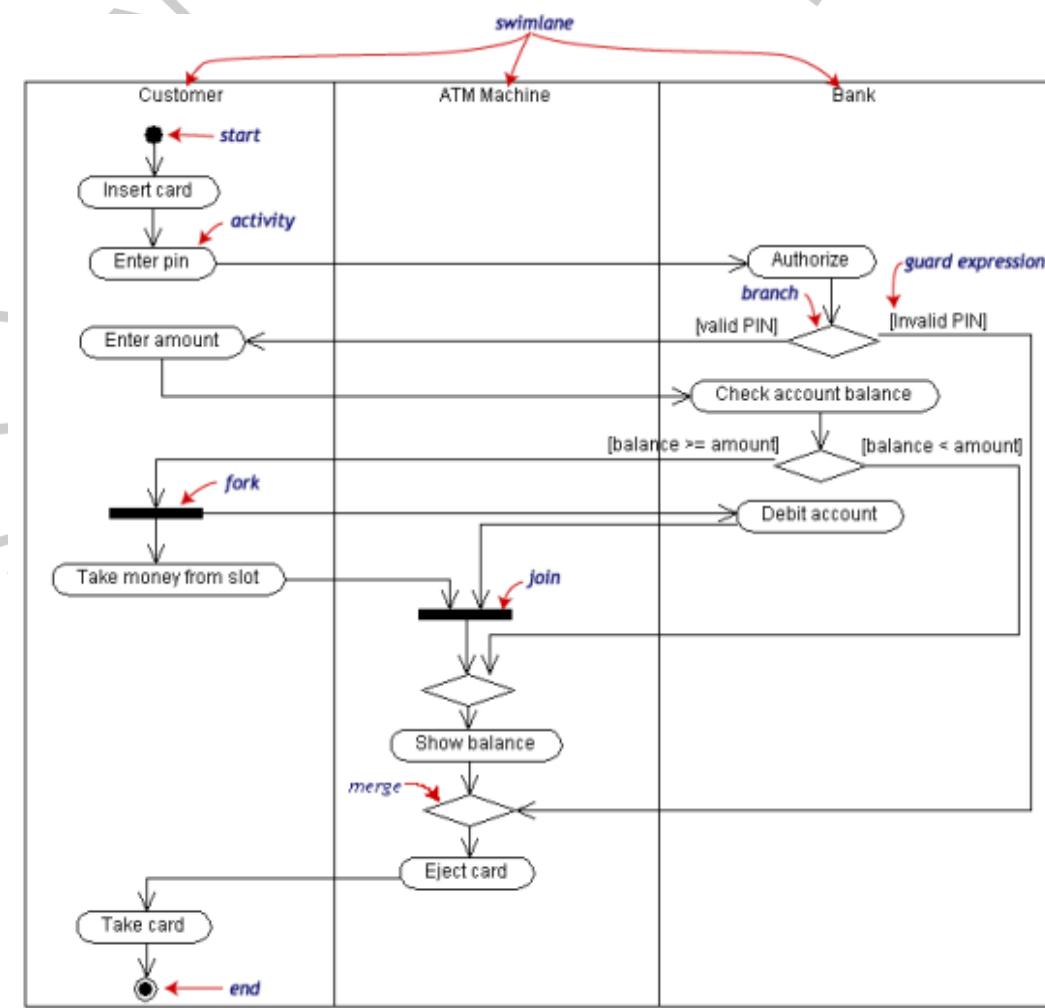
Tabel 2. 4 Tabel Simbol-simbol *activity diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Activity Symbol</i>	Menggambarkan sebuah tugas atau aktivitas dalam aliran kerja (<i>workflow</i>).
2.		<i>Start State</i>	<i>Start state</i> menggambarkan dimulainya aliran kerja (<i>work flow</i>) pada sebuah <i>activity diagram</i> . Hanya ada satu <i>start state</i> pada sebuah aliran kerja (<i>work flow</i>). <i>Start state</i> pada UML 2.0 digambarkan dengan gambar lingkaran solid.
3.		<i>End State</i>	Menggambarkan akhir pada sebuah <i>activity diagram</i> . Bisa terdapat lebih dari satu <i>end state</i> pada sebuah <i>activity diagram</i> .
4.		<i>State Transition</i>	<i>State Transition</i> menggambarkan kegiatan apa yang selanjutnya dilakukan setelah kegiatan sebelumnya selesai.
5.		<i>Decision</i>	Titik atau point pada <i>activity diagram</i> yang menunjukkan kondisi di mana kemungkinan ada perbedaan transisi atau terdapat pilihan kondisi.
6.		<i>Swimlane</i>	<i>Swimlane</i> digunakan untuk membuat pembatas <i>activity diagram</i> agar lebih mudah memahami siapa atau apa yang melakukan aktivitas.
7.		<i>Fork atau Join</i>	<i>Fork</i> digunakan untuk menggambarkan kegiatan yang dilakukan secara paralel. Sementara <i>join</i> digunakan untuk menggambarkan kegiatan paralel yang digabungkan menjadi satu.

Sumber : Jhon W Satzinger, 2016

Gambar 2.3 menggambarkan *activity diagram* proses pengambilan uang pada mesin ATM. *Customer* memasukkan kartu ATM pada mesin ATM beserta PIN. Sistem bank akan melakukan verifikasi jika PIN yang dimasukan *customer* benar maka akan muncul opsi jumlah yang ingin diambil. Jika PIN salah maka kartu akan dikeluarkan oleh mesin ATM. Setelah *customer* memasukkan nominal yang ingin diambil sistem bank akan mengurangi saldo rekening dan menampilkan sisa saldo pada mesin ATM. Mesin ATM mengeluarkan kartu tanda transaksi selesai dan *customer* dapat mengambil kartunya kembali.

Detail dari gambaran prosesnya dapat dilihat sebagai berikut :




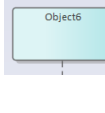
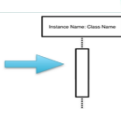
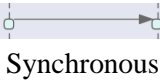





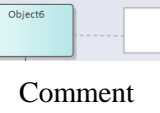
Gambar 2. 3 Contoh Activity Diagram (binus.ac.id)






C. Sequence Diagram

Sequence diagram merupakan satu di antara model diagram pada UML (*Unified Modelling Language*) yang bertujuan untuk menguraikan hubungan antar objek yang berprinsipkan pada urutan waktu (Satzinger et al., 2016).

Berikut adalah komponen-komponen yang terdapat pada *sequence diagram*:

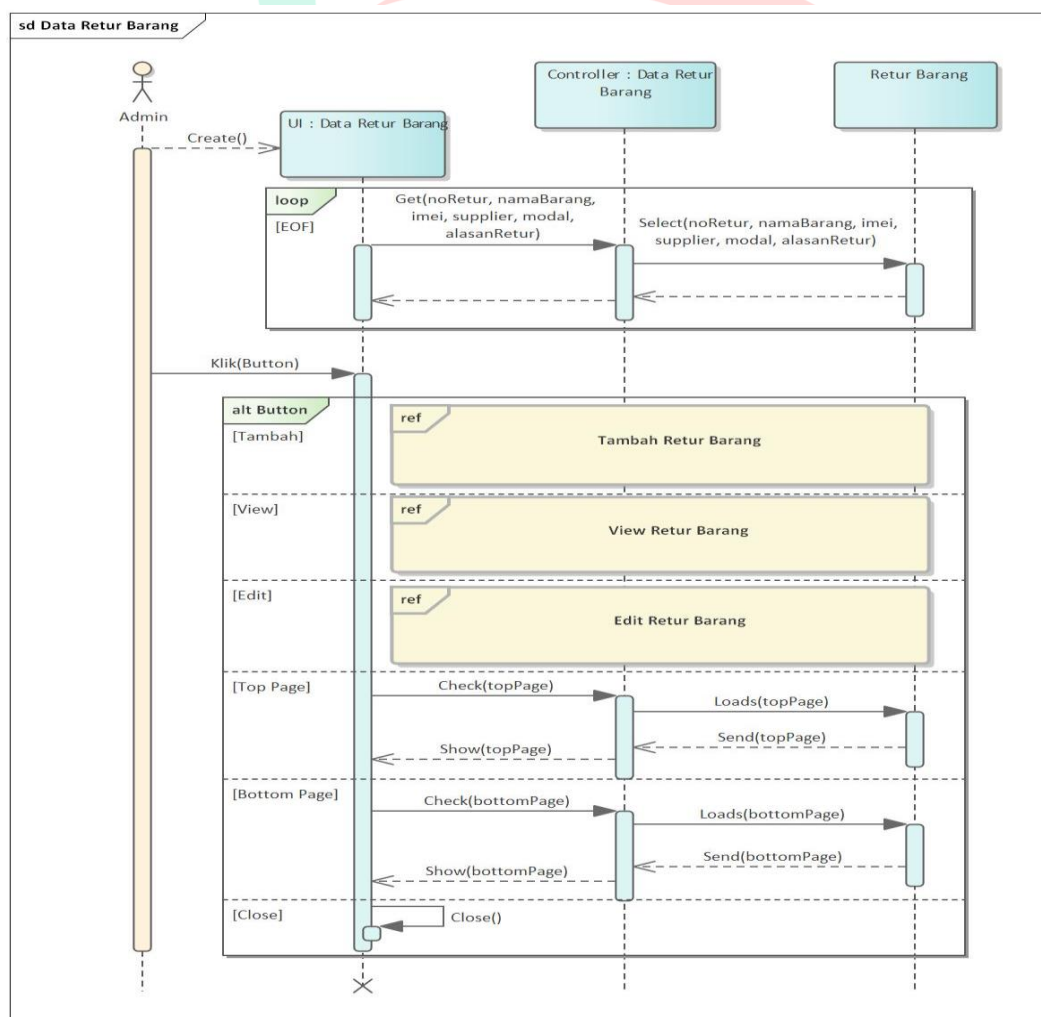
Tabel 2. 5 Tabel simbol-simbol *sequence diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menggambarkan siapa atau apa yang menjalankan atau menggunakan sistem.
2.		<i>Lifeline</i>	Mewakili peserta dalam <i>sequence diagram</i> . Memiliki persegi panjang yang berisi nama dari objeknya. Penggambaran dari lifeline diikuti garis putus-putus ke bawah.
3.		<i>Activation Box</i>	Menggambarkan waktu yang dibutuhkan objek untuk menyelesaikan tugas atau aktivitas.
4.	 Synchronous  Asynchronous  Return Message  Create Message  Self Message  Destroy Message  Comment	<i>Message</i>	<p><i>Message</i> menggambarkan bagaimana objek saling berinteraksi dengan saling mengirim dan menerima pesan. Berikut jenis-jenisnya:</p> <ul style="list-style-type: none"> • Synchronous : Digunakan pada saat pengirim pesan harus menunggu respons untuk melanjutkan interaksi. • Asynchronous : Digunakan pada saat pengirim pesan tidak perlu menunggu respons untuk melanjutkan interaksi. • Return Message : Untuk menginformasikan bahwa pesan sebelumnya telah selesai diproses dan dikembalikan kepada yang mengirimkan pesan. • Create Message : digunakan saat membuat objek baru. Digunakan bersamaan dengan pesan <i>Create ()</i>. • Self Message : Digunakan pada saat objek sebagai pengirim dan penerima pesan pada <i>lifeline</i> yang sama, • Destroy Message : untuk mengakhiri siklus hidup <i>lifeline</i>. • Comment : untuk menambahkan komentar.

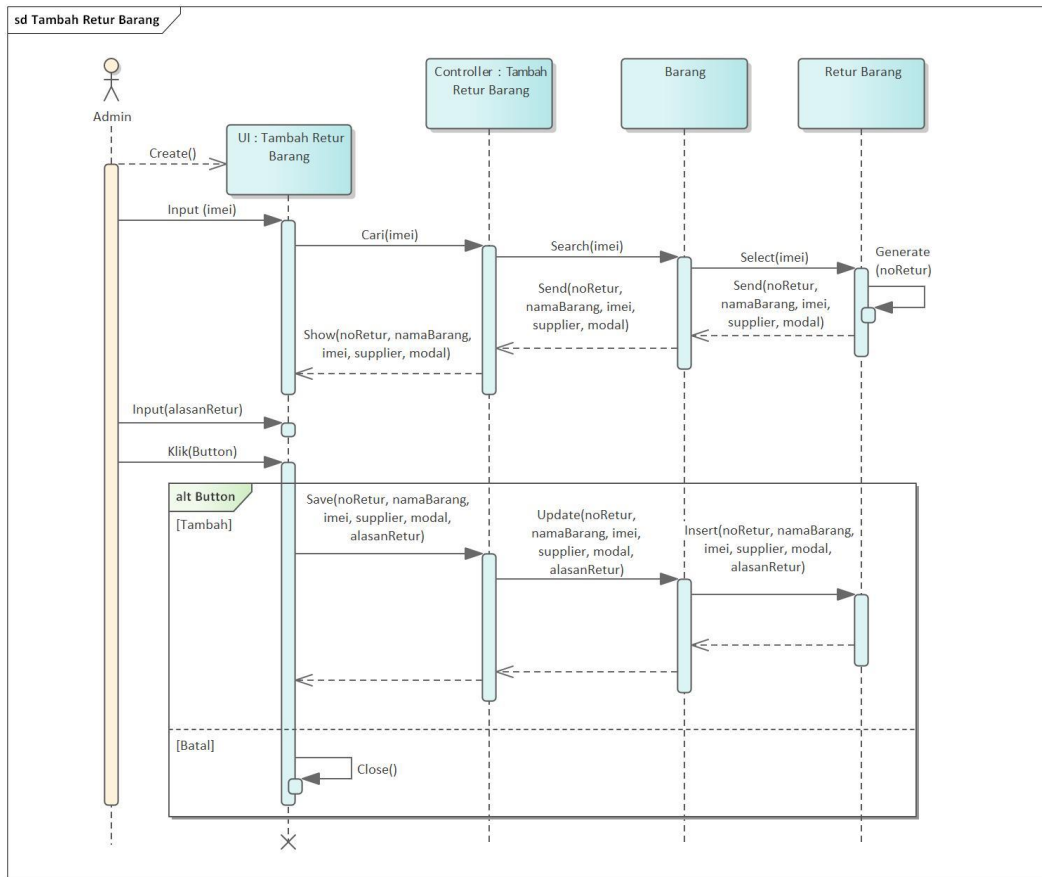
5.		<i>Focus Of Control</i>	Menunjukkan waktu yang berlangsung saat pemrosesan aktivitas di dalam objek.
6.		<i>Control Object</i>	Mengatur arus informasi dalam skenario sistem.
7.		<i>Boundary</i>	Biasanya berupa <i>interface</i> ataupun alat yang digunakan untuk berinteraksi antar sistem.
8.		<i>Entity</i>	Entity digunakan untuk menggambarkan hubungan dari kegiatan yang akan dilakukan.
9.		<i>Fragmen</i>	<p>Pada UML 2.0 fragmen telah dipergunakan. Dengan penggunaan fragmen maka akan lebih mudah dalam membuat dan menjaga keakuratan <i>sequence</i> diagram. Jenis fragmen terletak pada bagian kiri atas, adapun jenis-jenisnya sebagai berikut :</p> <ul style="list-style-type: none"> • Alt (<i>alternative</i>) : alternatif dari banyak fragmen. hanya kondisi <i>True</i> (benar) yang akan di eksekusi. • Opt (<i>optional</i>) : fragmen akan dijalankan jika kondisi yang mendukungnya <i>true</i> (benar). • Par (<i>parallel</i>) : setiap fragmen dijalankan secara paralel. • Loop (<i>loop</i>) : fragmen dapat dieksekusi beberapa kali, dan <i>guard</i> menunjukkan dasar iterasi. • Region (<i>critical region</i>) : fragmen hanya dapat mengeksekusi satu urutan sekaligus. • Neg (<i>negative</i>) : fragmen menunjukkan interaksi yang tidak valid. • Ref (<i>reference</i>) : mengacu pada interaksi yang diartikan pada diagram lain. Frame atau bingkai ditarik untuk menutupi lifelines yang terlibat dalam interaksi. Parameter dan nilai pengembaliannya dapat ditentukan. • Sd (<i>sequence diagram</i>) : untuk mengelilingi seluruh <i>sequence</i> diagram. <p>Notes : - diperbolehkan untuk menggabungkan beberapa fragmen.</p>

Gambar 2.4 sampai dengan gambar 2.8 menggambarkan *sequence* diagram mengelola data retur barang pada sistem informasi penjualan PT. Ekuator Putra Indonesia. Terdapat satu aktor yaitu admin dan tiga objek di dalamnya yaitu UI: data retur barang, Controller: data retur barang dan Barang. Langkah pertama admin membuka UI data retur barang dan sistem akan menampilkan list data retur barang yang telah diinput dalam bentuk tabel. Admin mempunyai enam pilihan tombol yang dapat digunakan yaitu tombol tambah untuk menambah data retur barang, tombol view untuk melihat data retur barang dan mencetak form retur yang akan ditempel pada barang, tombol edit untuk melakukan modifikasi pada data retur barang yang salah input, terdapat tombol scroll untuk membaca data bagian atas dan bawah tabel, serta tombol close untuk menutup layar. Pada tombol tambah, view dan edit terdapat referensi yang jika diklik maka akan menampilkan *sequence* diagram lainnya.

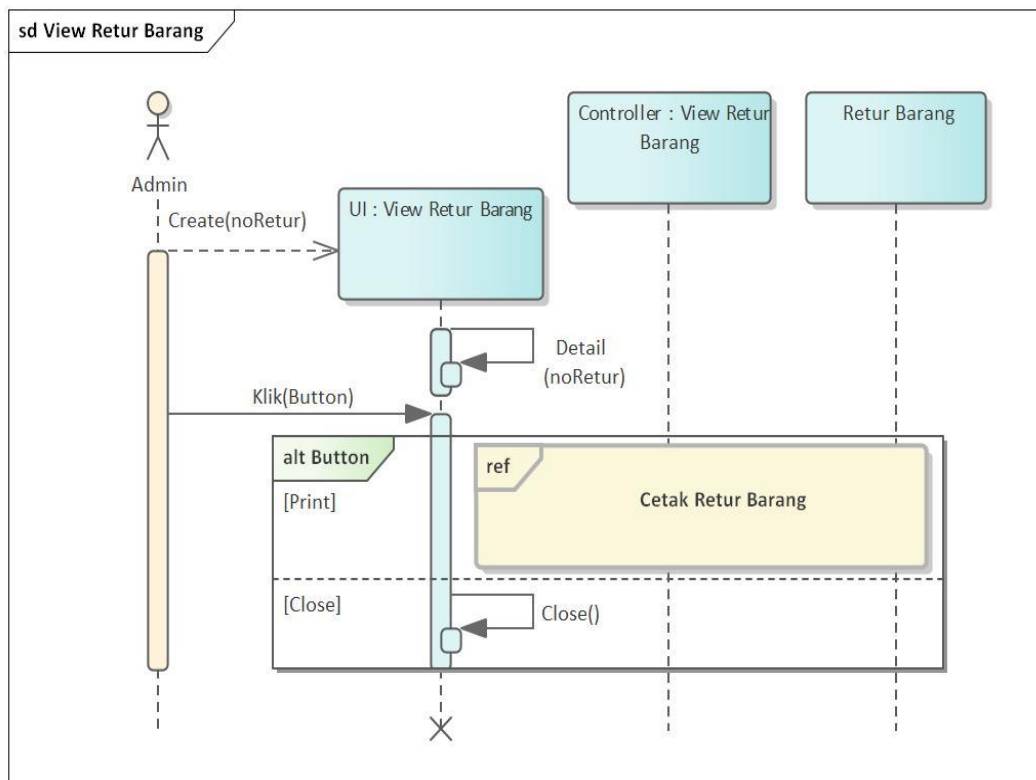
Detail dari gambaran prosesnya dapat dilihat sebagai berikut :



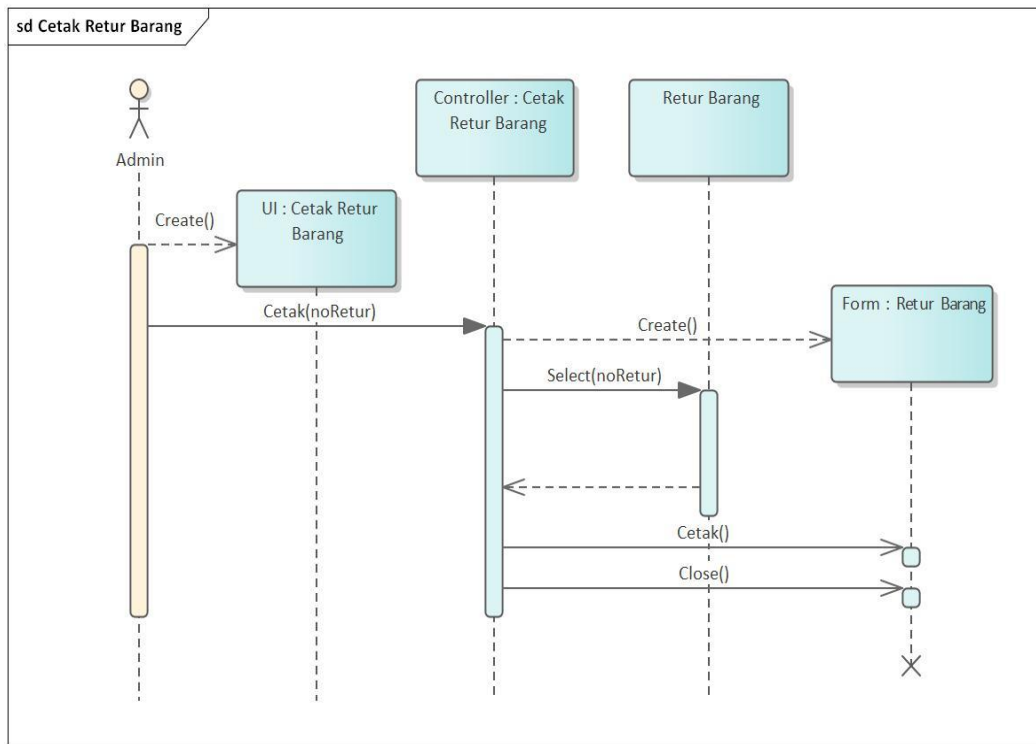
Gambar 2. 4 Contoh *Sequence Diagram* Proses Retur Barang



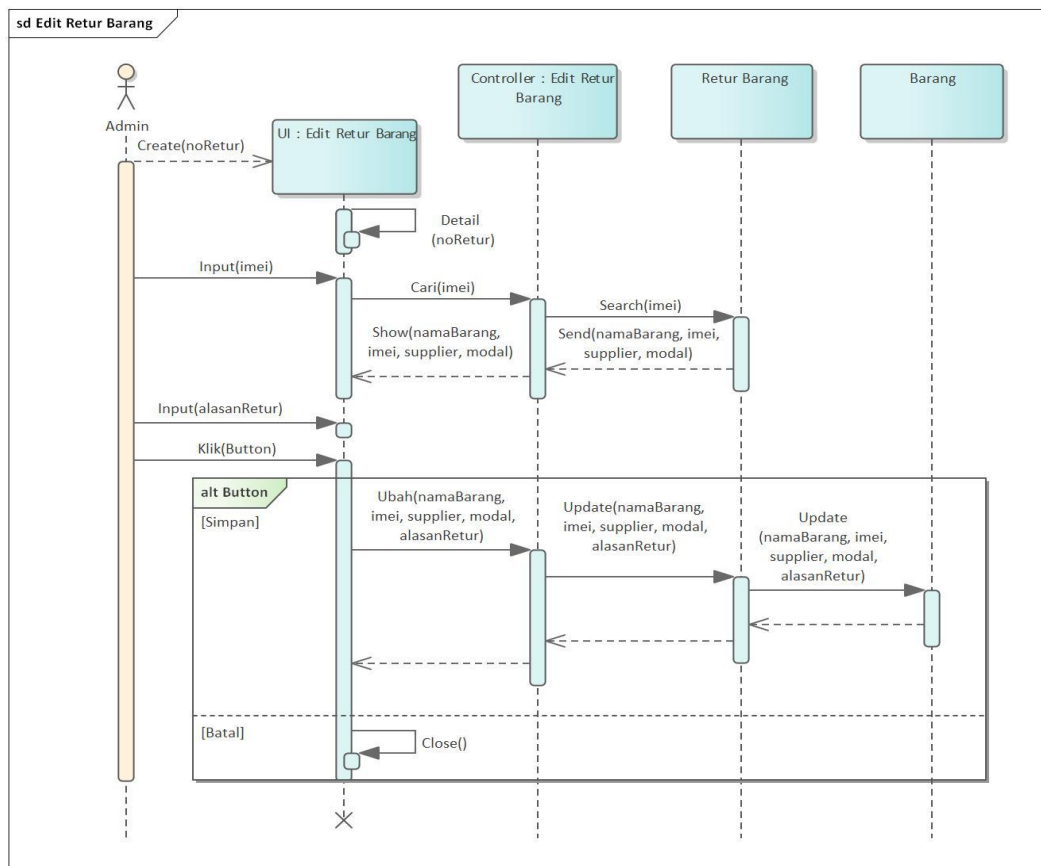
Gambar 2. 5 Contoh Sequence Diagram Tambah Retur Barang



Gambar 2. 6 Contoh Sequence Diagram View Retur Barang



Gambar 2. 7 Contoh *Sequence Diagram* Cetak Retur Barang



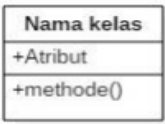





Gambar 2. 8 Contoh *Sequence Diagram* Edit Retur Barang

D. Class Diagram

Class diagram atau disebut juga diagram kelas adalah diagram yang menerangkan bentuk sistem dari sisi penjabaran *class-class* yang nantinya digunakan untuk membangun sebuah sistem (Satzinger et al., 2016).

Berikut adalah komponen-komponen yang terdapat pada *class diagram*:

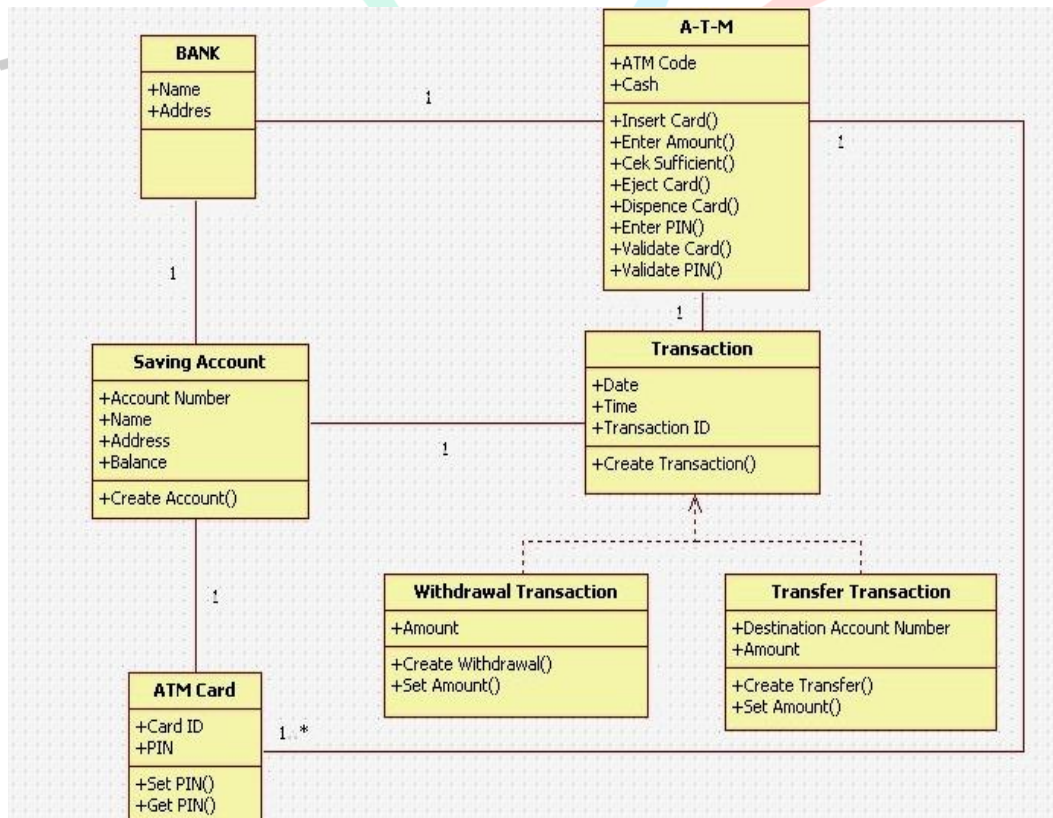
Tabel 2. 6 Tabel simbol-simbol *class diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Class</i>	Mendefinisikan kelas pada sistem yang terbagi menjadi tiga bagian. Bagian atas berisikan nama kelas. Bagian tengah berisikan atribut dari kelas tersebut. Sementara bagian paling bawah berisikan <i>methode</i> dari kelas.
2.		<i>Association</i>	Hubungan statis antar kelas. Mendefinisikan kelas yang memiliki atribut berupa kelas lain. Atau dapat juga mendefinisikan kelas yang perlu mengetahui keberadaan kelas lainnya.
3.		<i>Agregation</i>	Hubungan yang menerangkan bahwa ada suatu kelas yang menjadi atribut bagi kelas lainnya.
4.		<i>Composition</i>	Gambaran khusus dari <i>agregation</i> di mana kelas yang menjadi bagian diciptakan setelah kelas <i>whole</i> dibuat.
5.		<i>Generalization</i>	Hubungan antar kelas yang berupa pengertian hubungan umum-khusus (generalisasi – spesialisasi)
6.		<i>Directed Association</i>	Asosiasi dengan pengertian kelas tersebut digunakan oleh kelas yang lain.

Sumber : Jhon W Satzinger, 2016

Gambar 2. 9 menggambarkan *Class diagram* transaksi pada mesin ATM. Terdapat tujuh kelas yang masing-masing terhubung dengan notasi *association* yang menandakan hubungan statis antara kelas-kelas tersebut. Sementara untuk kelas *withdrawal transaction* dan *transfer transaction* terhubung dengan notasi *Directed Association* yang menandakan bahwa kelas *transaction* digunakan secara bersamaan oleh kelas *withdrawal transaction* dan *transfer transaction*. Pada kelas *Bank* hanya berisikan atribut nama dan alamat karena tidak ada metode yang dapat dilakukan pada kelas tersebut. Pada kelas *ATM* terdapat atribut *ATM code* dan *cash* untuk *method*-nya berisikan *insert card*, *enter amount*, *cek sufficient*, *eject card*, *dispenche card*, *enter PIN*, *validate card* dan *validate PIN*. Pada kelas *saving account* terdapat atribut *account number*, *name*, *addres*, *balance* dengan *method* *create account*. Pada kelas *ATM card* berisikan atribut *card id* dan *PIN* serta *method* *set PIN* dan *Get PIN*. Pada kelas *transaction* terdapat atribut *date*, *time* dan *transaction ID* serta *method* *create transaction*. Pada kelas *withdrawal transaction* terdapat atribut *amount* dan *method* *create withdrawal*, *set amount*. Terakhir pada kelas *transfer transaction* terdapat atribut *destination account number*, *amount* dengan *method* *create transfer* dan *set amount*.

Detail dari gambaran prosesnya dapat dilihat sebagai berikut :



Gambar 2. 9 Contoh *Class Diagram* (binus.ac.id)

2.1.16 User Interface

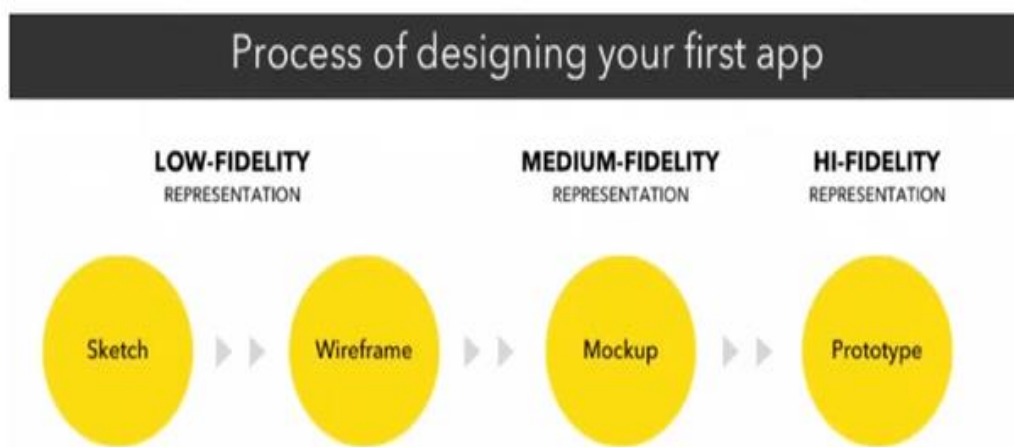
Pengertian dari *user interface* adalah prosedur perolehan dan pengiriman kembali informasi kepada pengguna (*user*). Untuk mendukung proses penyusunan alur penelusuran masalah sehingga dapat ditemukan penyelesaian dari masalah tersebut (Aprilia, 2020).

Pengertian dari *user experience* adalah reaksi atau pengetahuan seseorang dan penerimaannya dari pemakaian sebuah produk, sistem, atau jasa. Seberapa puas dan nyaman seseorang terhadap pemakaian sebuah produk, sistem, atau jasa adalah apa yang dinilai dari *user experience* (Adani, 2022).

Berdasarkan pendapat para ahli di atas maka dapat disimpulkan bahwa *user interface (UI) design* adalah proses pembuatan antar muka pada aplikasi, agar terlihat menarik secara visual. Yang membuat *UI design* disebut dengan *UI designer*. Tugas *UI designer* adalah mendesain segala sesuatu yang berhubungan dengan tampilan. Seperti tata letak atau *layout*, warna, jenis *font* dan gambar. Dengan tujuan membuat aplikasi yang menarik dan *user friendly*.

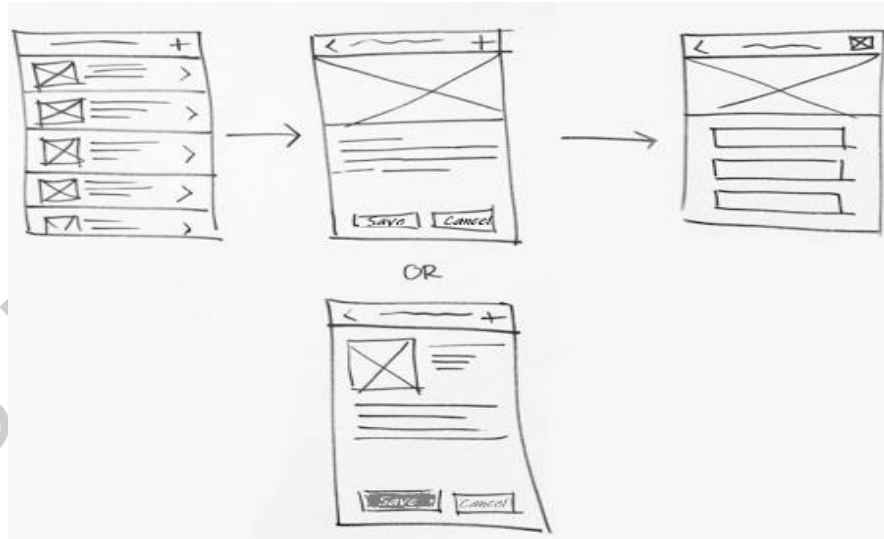
Merujuk dari pendapat ahli di atas dapat disimpulkan bahwa *UX Design (Interaction Designer)* proses dalam pencarian, pembentukan dan menciptakan interaksi *user* dalam penggunaan aplikasi untuk mendapatkan kepuasan pengguna (*user*) yang lebih baik. *UX designer* bertugas untuk meneliti bagaimana kebiasaan dan keperluan para pengguna kemudian mengimplementasikannya kedalam sebuah *interface* yang menarik.

Proses desain aplikasi :



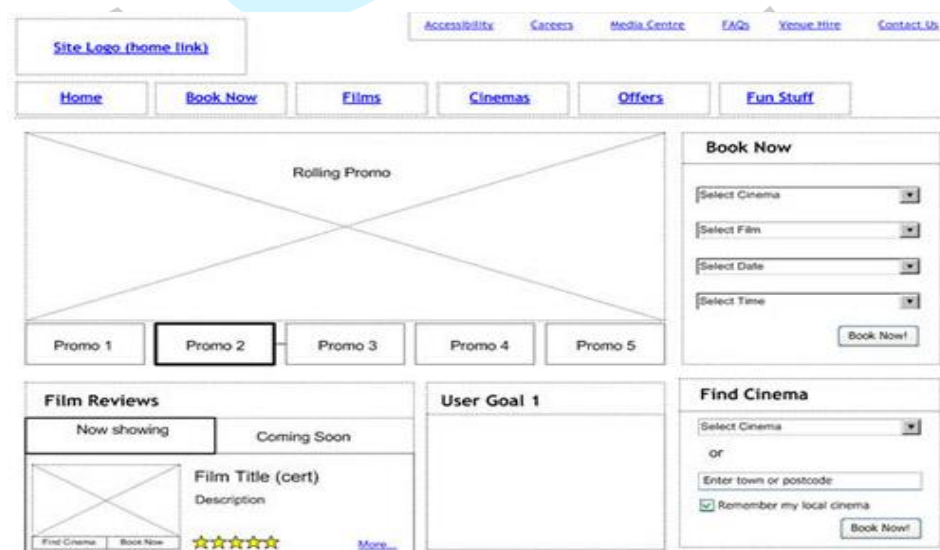
Gambar 2. 10 Proses Desain Aplikasi (TechforID)

1. Sketsa (*sketch*) : menggunakan kertas dan pensil. Gambar tangan bebas untuk membuat sketsa aplikasi yang dilakukan pada selembar kertas. Teknik ini masuk kedalam tingkat presisi rendah (*low fidelity*). Berikut contoh dari sketsa :



Gambar 2. 11 Contoh Gambar Sketsa (TechforID)

2. *Wireframe* : ilustrasi dua dimensi dari *interface* halaman yang secara khusus berfokus pada alokasi ruang dan prioritas konten, fungsi yang tersedia dan aktivitas yang diinginkan. Biasanya hanya berupa rangka tanpa warna ataupun gambar. Teknik ini masuk kedalam tingkat presisi rendah (*low fidelity*). *Tools* yang digunakan untuk membuat *wireframe* diantaranya *balsamiq*, *wireframe sketcher*, *lucid chart* dll. Berikut contoh dari *wireframe* :



Gambar 2. 12 Contoh Gambar *Wireframe* (TechforID)

3. *Mockup* : media visual yang digunakan untuk melihat *preview* sebuah konsep desain. Karena diberikan efek visual membuat hasil gambar menyerupai wujud yang sebenarnya. *Mockup* masuk kedalam kategori *medium fidelity*. *Tools* yang digunakan untuk membuat *mockup* diantaranya *sketch*, *figma*, *adobe illustrator* dll. Berikut contoh dari *mockup* :

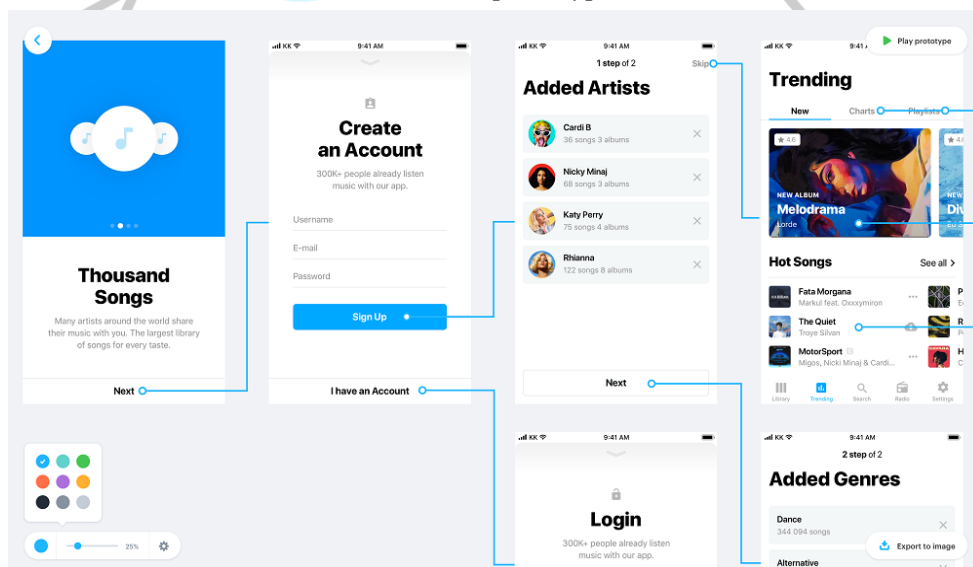
Proses Desain Aplikasi

Mockup (Sketch, Figma, Adobe Illustrator)



Gambar 2. 13 Contoh Gambar *Mockup* (TechforID)

4. *Prototype* : merupakan representasi dengan tingkat presisi yang tinggi *high fidelity*. *Prototype* hampir sama dengan *mockup* hanya saja dilengkapi dengan interaksi, animasi dan reaksi ketika sebuah tombol diklik. *Tools* yang digunakan untuk membuat *prototype* diantaranya *sketch*, *figma*, *adobe illustrator* dll. Berikut contoh *prototype*:



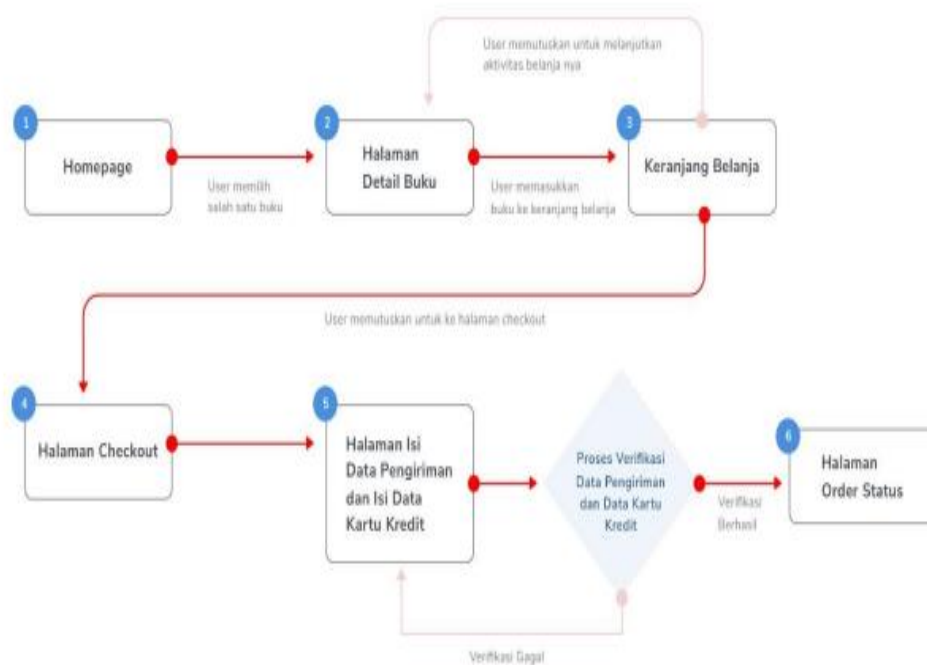
Gambar 2. 14 Contoh Gambar *Prototype* (TechforID)

2.1.17 User Flow

User flow atau alur pengguna merupakan urutan langkah-langkah yang perlu dilakukan oleh pengguna untuk melakukan suatu *task* atau tugas yang perlu dilakukan oleh pengguna. Pembuat rancangan aplikasi menggunakan *user flow* untuk mengetahui apa saja yang sering dilakukan oleh pengguna atau langkah-langkah yang dibuat untuk membuat pengguna merasa nyaman saat melaksanakan *task* atau tugas (Oliveira et al., 2022).

Sementara menurut (Gie, 2021) *user flow* adalah rangkaian langkah yang perlu dilakukan *user* atau pengguna untuk menyelesaikan tugas pada aplikasi atau situs web. Contohnya adalah rangkaian langkah yang dilakukan oleh user saat melakukan belanja di situs ecommerce.

Berdasarkan pendapat di atas dapat disimpulkan bahwa *user flow* adalah urutan langkah-langkah yang dirancang sedemikian rupa untuk membuat pengguna merasa nyaman saat melakukan tugas atau *task* yang perlu dilakukan baik pada aplikasi atau situs web. Dengan contoh *user flow* seperti gambar 2.15.



Gambar 2. 15 Contoh *User Flow* (TechforID)

2.1.18 Perancangan Basis Data

Basis data atau *database* adalah himpunan informasi yang tersusun secara sempurna dan teratur serta tersimpan di dalam komputer sehingga informasi tersebut nantinya dapat diperiksa menggunakan suatu program komputer untuk mendapatkan informasi yang diharapkan (Solichin, 2020).

Database atau basis data tersusun dari dua kata yaitu basis dan data. Basis dapat dikatakan sebagai markas atau gudang sementara data adalah catatan dari sekumpulan fakta yang mewakili objek seperti manusia, barang, hewan, konsep, peristiwa dan lainnya yang digambarkan dalam bentuk huruf, gambar, angka, simbol, teks, bunyi atau kombinasinya (S. Setiawan, 2020).

Contoh basis data yang banyak digunakan dan *open source* diantaranya MySQL, PostgreSQL dan MariaDB.

2.2 Tinjauan Studi

Tinjauan studi dilakukan untuk mendukung tugas akhir, tinjauan studi berkaitan dengan topik yang dibahas oleh dalam tugas akhirnya dan memperkuat hasil penelitian yang telah dilakukan. Berikut adalah beberapa referensi tinjauan studi yang digunakan :

1. Jurnal penelitian yang disusun oleh (Ong, 2022) dengan judul : **“Analisa Perancangan Dan Penggunaan Sistem Penjualan Pada PT. Tirta Varia Inti Pratama”**. Jurnal ini membahas PT. Tirta Varia Inti Pratama sebagai perusahaan yang bergerak pada bidang penjualan yang saat ini mengalami perkembangan yang pesat dalam usahanya. Sistem yang digunakan saat ini tidak lagi dapat memenuhi kebutuhan pencatatan transaksi penjualan yang terjadi beserta pembuatan laporan yang dipisahkan berdasarkan kategori. Beralihnya ke sistem komputerisasi akan meliputi tiga langkah: analisis, desain dan implementasi. Alat yang akan mendukung proses analisis sistem dan desain, sebagai berikut: Data *Flow* Diagram, ER-Diagram, Skema Database dan Pengguna *Sketch Interface Design*. Alat yang akan digunakan untuk mengimplementasikan sistem adalah PHP & MySQL.

2. Jurnal penelitian yang disusun (Ahmadar et al., 2021) dengan judul **“Perancangan Sistem Informasi Penjualan Berbasis Web Pada Rahayu Photo Copy Dengan Database MySQL”**. Jurnal ini membahas Rahayu *photo copy* yang menggunakan pencatatan manual di dalam bisnisnya namun terdapat banyak kendala yang terjadi, seperti kehilangan nota transaksi, nota transaksi tidak tersusun dengan rapi, pencatatan manual yang kurang konsisten mengakibatkan kesulitan dalam pencatatan transaksi hingga kesalahan pencatatan laporan akhir bulan. Adapun upaya pemecahan masalah tersebut yaitu dengan membuat sistem informasi penjualan berbasis web menggunakan MySQL.
3. Jurnal penelitian yang disusun (Nugraha, 2021) dengan judul **“Perancangan Aplikasi Point Of Sales (POS) Pada Apotek Mitra Sejahtera Berbasis Web”**. Jurnal ini membahas tentang pencatatan dan pengelolaan data transaksi pada Apotek Mitra Sejahtera. Sistem informasi *Point Of Sale* (POS) merupakan sebuah sistem informasi yang dirancang dan dibangun untuk mencatat transaksi penjualan dan mengolah data pada apotek mitra sejahtera untuk membantu jalannya kegiatan operasional dan mempercepat proses pelayanan mulai dari proses pendaftaran sampai pembayaran sehingga pelayanan dapat ditingkatkan. Sistem informasi POS ini dibangun dengan menggunakan metode *Waterfall* menggunakan *Linear Sequential Model*. Teknik pengumpulan data yang dilakukan yaitu wawancara, observasi, serta studi pustaka untuk menganalisa kebutuhan dari sistem informasi POS. Perancangan sistem informasi POS ini menggunakan *Unified Modeling Language* (UML) untuk menggambarkan proses-proses yang terjadi pada sistem informasi. Diagram yang digunakan yaitu *use case* diagram, *activity* diagram, *sequence* diagram, dan *class* diagram. Bahasa pemrograman yang digunakan yaitu PHP dengan *framework Codeigniter*, dan MySQL sebagai database. Hasil dari perancangan sistem informasi yaitu *form* pengolahan data master, *form* registrasi *customer*, *form* pengolahan transaksi, serta *form* pengelolaan laporan.
4. Jurnal penelitian yang disusun (Alukadinata et al., 2021) dengan judul **“Perancangan Sistem Informasi Penjualan Handphone di Forza Mufid Perdana Selular”**. Jurnal ini membahas pencatatan laporan penjualan pada

toko forza mufid perdana selular. Program ini memungkinkan admin untuk input data, edit data, hapus data, pencarian data, penyaringan data dan pencetakan data ke media kertas. Data yang diinput berupa data barang, data jenis barang dan penjualan barang. Sedangkan *output* yang dihasilkan sistem informasi ini berupa laporan data barang dan penjualan barang.

5. Jurnal penelitian yang disusun (Reinaldi, 2022) dengan judul **“Perancangan Sistem Penjualan Berbasis Web Pada Toko Smart Jaya Phone”**. Jurnal ini membahas toko *smart jaya phone* dan pencatatan transaksi penjualan yang terjadi di dalamnya mengalami perubahan yang positif setelah menerapkan sistem penjualan berbasis web di dalamnya. Pencatatan dan pembuatan laporan tidak lagi dilakukan secara manual sehingga minim dari salah catat dan kehilangan nota seperti yang sebelumnya.
6. Jurnal penelitian yang disusun (Putri, 2018) dengan judul **“Perancangan Sistem Informasi Penjualan Berbasis Web Pada Toko Muncul Komputer”**. Jurnal ini membahas toko Muncul Komputer yang menggunakan pencatatan manual dalam proses bisnisnya ingin pencatatan laporan penjualan dapat dilakukan dengan lebih cepat dan rapi. Sehingga dibuatlah ini sistem informasi penjualan berbasis web yang dapat mempermudah pembuatan laporan dan nota pada toko.

Hasil tinjauan studi terhadap penelitian yang telah dilakukan sebelumnya akan digunakan sebagai acuan pembuatan Tugas akhir (TA) guna merancang sistem yang mampu memberikan pencatatan laporan penjualan yang cepat dan meminimalisir kesalahan catat. Karena tidak lagi menggunakan pencatatan manual laporan yang dihasilkanpun dapat berupa *hardcopy* ataupun *softcopy*. Penggunaan sistem informasi pada penjualan berbasis web dapat mempercepat proses bisnis terutama pembuatan nota dan laporan untuk toko yang sudah memiliki transaksi cukup banyak. Dalam hal ini PT. Ekuator Putra Indonesia dalam setiap harinya rata-rata mencetak lebih dari 100 nota pembelian tentunya akan mempermudah dan mempercepat proses pembuatan nota serta dengan penerapan sistem informasi di dalamnya dapat meminimalisir kesalahan catat yang mengakibatkan kerugian bagi perusahaan.

Pembuatan laporan dengan menerapkan sistem informasi penjualan berbasis web pada jurnal di atas dapat dilakukan dengan cepat dan mudah karena semua data yang dibutuhkan tidak perlu lagi diinput, pengguna hanya perlu mengklik jenis laporan yang ingin dibuatkan apakah itu laporan penjualan harian, mingguan atau mungkin bulanan. Tersedianya fitur cetak nota juga akan memberikan kesan profesional pada proses bisnis PT. Ekuator Putra Indonesia karena dengan status toko branding Samsung kelas SE (Samsung *Experience*) memberikan nota dengan tulisan tangan akan mengurangi kepercayaan dari konsumen dan tidak seragam dengan standar yang diberikan oleh toko Samsung dengan status SE-lainnya.

