

BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Sistem Informasi

Sistem adalah sekumpulan elemen yang saling berinteraksi mengikuti aturan tertentu dalam suatu struktur, membentuk kesatuan untuk menjalankan fungsi tertentu demi mencapai tujuan yang telah ditetapkan. Sebuah sistem memiliki beberapa karakteristik utama, seperti komponen sistem, batasan, lingkungan eksternal, keterkaitan antar elemen, masukan (input), keluaran (output), proses pengolahan, serta tujuan akhir yang ingin dicapai. Menurut Murdick (1991), sistem dapat didefinisikan sebagai serangkaian elemen atau proses yang saling berhubungan dalam mengelola data dalam periode tertentu untuk menghasilkan informasi, energi, atau produk.

Sementara itu, informasi merupakan data yang telah diproses sehingga memiliki makna dan nilai lebih bagi penerima, membantu dalam mengurangi ketidakpastian saat mengambil keputusan. McLeod (dalam Yakub, 2012:8) menjelaskan bahwa informasi adalah hasil dari pengolahan data yang membuatnya lebih berguna bagi pengguna. Sutabri (2012:22) menambahkan bahwa informasi adalah data yang telah diinterpretasikan atau dikategorikan agar dapat digunakan dalam proses pengambilan keputusan. Sistem informasi sendiri merupakan kombinasi terstruktur yang mencakup manusia, perangkat keras, perangkat lunak, jaringan komunikasi, serta sumber daya data yang berfungsi untuk mengumpulkan, mengolah, dan mendistribusikan informasi dalam suatu organisasi.

2.1.2 Aplikasi

Menurut Kadir (2008:3), program aplikasi adalah perangkat lunak yang telah siap digunakan atau dirancang untuk menjalankan fungsi tertentu, baik bagi pengguna maupun aplikasi lainnya. Secara umum, aplikasi dapat diartikan sebagai implementasi dari suatu konsep dengan tujuan tertentu atau

sebagai program komputer yang dibuat untuk membantu manusia dalam menyelesaikan berbagai tugas. Dalam *Kamus Besar Bahasa Indonesia* (KBBI), aplikasi didefinisikan sebagai penerapan dari desain sistem yang berfungsi untuk mengolah data menggunakan bahasa pemrograman atau aturan tertentu. Aplikasi merupakan program komputer yang dikembangkan untuk menangani tugas-tugas spesifik sesuai kebutuhan penggunanya. Selaras dengan pengertian tersebut, Rachmad Hakim S (2012:38) menjelaskan bahwa aplikasi adalah perangkat lunak yang dirancang untuk keperluan khusus, seperti pengelolaan dokumen, pengaturan sistem operasi Windows, bermain game, serta berbagai fungsi lainnya.

Aplikasi menggambarkan konsep dasar dan prinsip-prinsip yang mendasari pengembangan serta penggunaan aplikasi perangkat lunak. Secara umum, aplikasi adalah perangkat lunak yang dirancang untuk menjalankan tugas-tugas tertentu bagi pengguna atau sistem, bertujuan untuk mempermudah dan mempercepat proses kerja serta membantu pengguna dalam menyelesaikan aktivitas sehari-hari. Dalam pengembangan aplikasi, terdapat beberapa komponen penting yang perlu dipertimbangkan. Antarmuka pengguna (*user interface*) adalah bagian yang berinteraksi langsung dengan pengguna dan harus dirancang agar interaktif, intuitif dan mudah digunakan serta dipahami. Logika bisnis mengacu pada aturan dan proses yang mengatur bagaimana data diproses dan diterapkan dalam aplikasi. Basis data menyimpan data yang diperlukan aplikasi, sedangkan middleware menghubungkan aplikasi dengan sistem lain dan memungkinkan integrasi yang lancar.

2.1.3 *Point of Sales* (POS)

Point of Sales (POS) adalah sistem kasir yang digunakan untuk memproses transaksi penjualan di restoran atau kafe. Selain sekadar menangani transaksi jual beli, sistem ini juga dapat mencakup manajemen kasir dan gudang yang terintegrasi untuk mengelola data penjualan. Ini menghasilkan informasi yang berguna sebagai dasar dalam pengambilan keputusan terkait usaha (Jaber et al., 2021). *Point of Sales* (POS) memainkan peran penting bagi perusahaan karena merupakan titik di mana uang diterima dari pelanggan. Bagi pemilik

perusahaan, uang yang diterima berfungsi sebagai indikator utama untuk mengukur pendapatan perusahaan mereka.

Point of Sales (POS) mencakup pemahaman mendalam tentang konsep dasar dan fungsi dari sistem POS dalam konteks bisnis dan teknologi. POS adalah sistem yang dirancang untuk memproses transaksi penjualan dengan efisiensi tinggi di berbagai jenis usaha, seperti restoran, kafe, dan toko ritel. Sistem ini mengintegrasikan perangkat keras, perangkat lunak, dan jaringan untuk mencatat transaksi, mengelola inventaris, dan menghasilkan laporan yang penting untuk keputusan bisnis. Fungsi dan manfaat utama dari sistem POS termasuk pengelolaan transaksi, pengelolaan inventaris, pembuatan laporan, peningkatan efisiensi, dan peningkatan layanan pelanggan. POS memfasilitasi proses pembayaran dengan berbagai metode, otomatis memperbarui stok barang saat transaksi terjadi, dan menghasilkan laporan yang membantu pemilik usaha dalam perencanaan strategis.

2.1.4 Mobile

Dalam dunia teknologi, istilah "*mobile*" merujuk pada perangkat komputasi yang dapat dibawa dan digunakan di mana saja, yang memungkinkan pengguna untuk melakukan berbagai aktivitas digital seperti komunikasi, akses informasi, dan transaksi secara nirkabel. *Mobile* juga mengacu pada platform yang mendukung pengembangan dan distribusi aplikasi yang dirancang khusus untuk perangkat seluler. Dalam penjelasan Prakarsya (2019), istilah "*mobile*" digunakan untuk menggambarkan aplikasi yang berjalan pada perangkat kecil, portabel, dan nirkabel yang mendukung komunikasi. Konsumen cenderung mencari perangkat yang ringkas demi kenyamanan dan kemudahan mobilitas. Selain itu, perangkat *mobile* memiliki keunggulan dalam efisiensi daya dibandingkan dengan komputer desktop. Istilah "mobile" sendiri mengacu pada sesuatu yang dapat berpindah atau bergerak, sehingga aplikasi *mobile* merujuk pada perangkat lunak yang dirancang untuk dijalankan pada perangkat tersebut. Dengan adanya aplikasi *mobile*, pengguna dapat dengan mudah mengakses berbagai layanan, mulai dari hiburan, transaksi jual beli, pembelajaran, penyelesaian tugas kantor, hingga aktivitas seperti browsing dan lainnya.

Aplikasi mobile, yang dirancang khusus untuk perangkat seluler, dapat mencakup berbagai jenis, termasuk aplikasi produktivitas, permainan, media sosial, dan bisnis seperti *Point of Sales* (POS). Meskipun memiliki banyak keunggulan seperti portabilitas dan kemudahan akses, pengembangan untuk mobile juga menghadirkan tantangan, seperti keterbatasan baterai dan ukuran layar yang lebih kecil. Dalam konteks bisnis, teknologi mobile menjadi alat penting untuk mendukung operasi dan strategi pemasaran, memungkinkan perusahaan untuk meningkatkan interaksi dengan pelanggan, mengotomatisasi proses, dan memperluas jangkauan pasar.

2.2 Teori Khusus

A. Rancang Bangun

Proses perancangan melibatkan penerjemahan hasil analisis sistem ke dalam bentuk yang lebih rinci menggunakan bahasa pemrograman, dengan tujuan mendefinisikan bagaimana setiap komponen dalam sistem akan diimplementasikan. Sementara itu, pembangunan sistem mencakup aktivitas dalam menciptakan, menggantikan, atau menyempurnakan sistem yang telah ada, baik secara menyeluruh maupun sebagian (Pressman, 2016). Menurut *Kamus Besar Bahasa Indonesia* (KBBI), istilah “rancang” merujuk pada proses perencanaan atau pengaturan sesuatu sebelum pelaksanaannya, sedangkan “bangun” mengacu pada sesuatu yang didirikan atau dibuat agar dapat berdiri (Departemen Pendidikan Nasional, 2015).

Dari penjelasan tersebut, dapat disimpulkan bahwa proses rancang bangun adalah serangkaian kegiatan yang dimulai dengan perencanaan berdasarkan hasil analisis, yang kemudian diterjemahkan ke dalam bahasa pemrograman untuk menggambarkan implementasi komponen sistem. Selain itu, pembangunan sistem melibatkan penciptaan, penggantian, atau perbaikan sistem yang ada. Dengan demikian, rancang bangun mencakup perpaduan antara tahap perencanaan awal (rancang) dan pelaksanaan nyata untuk membangun atau memperbaiki sistem (bangun).

B. SDLC (*Software Development Life Cycle*)

Software Development Life Cycle (SDLC) adalah pendekatan yang terstruktur dan sistematis dalam pengembangan perangkat lunak. Proses ini

mencakup serangkaian tahapan yang dirancang untuk merancang, membangun, menguji, serta memelihara perangkat lunak secara terorganisir. Tujuan utama dari SDLC adalah memastikan bahwa perangkat lunak yang dikembangkan tidak hanya memenuhi kebutuhan pengguna, tetapi juga memiliki kualitas tinggi dan dapat diandalkan. SDLC, yang juga dikenal sebagai *System Development Life Cycle*, merupakan proses yang digunakan dalam pengembangan maupun modifikasi sistem perangkat lunak dengan menerapkan model dan metodologi yang telah terbukti efektif berdasarkan praktik terbaik. Pendekatan ini telah digunakan secara luas dalam pengembangan sistem informasi, terutama sejak tahap awal evolusinya (AS & Shalahuddin, 2015; Susanto, 2004).

Proses ini melibatkan berbagai tahapan yang masing-masing memiliki tujuan dan aktivitas spesifik.

1) Perencanaan (*Planning*)

Tahap di mana tujuan proyek ditetapkan dan rencana pengembangan dibuat. Di sini, analisis kebutuhan dilakukan untuk memahami apa yang diperlukan dari sistem, serta penentuan sumber daya dan jadwal proyek. Tahap ini sangat penting karena menetapkan dasar untuk semua langkah berikutnya.

2) Analisis Kebutuhan (*Requirement Analysis*)

Pada tahap ini, informasi dikumpulkan dari pemangku kepentingan untuk memahami kebutuhan dan ekspektasi mereka. Proses ini melibatkan wawancara, survei, dan studi dokumen untuk mengidentifikasi secara rinci apa yang harus dilakukan oleh sistem.

3) Desain (*Design*)

Kemudian masuk ke tahap desain, di mana arsitektur sistem, antarmuka pengguna, dan struktur database dirancang berdasarkan informasi yang diperoleh selama analisis. Desain ini berfungsi sebagai blueprint yang akan diikuti selama pengembangan.

4) Pengembangan (*Development*)

Pada tahap pengembangan, kode perangkat lunak ditulis dan sistem dibangun sesuai dengan desain yang telah ditentukan. Tahap ini melibatkan penerjemahan desain menjadi aplikasi yang berfungsi dengan mengimplementasikan fitur dan fungsi yang direncanakan.

5) Pengujian (*Testing*)

Setelah pengembangan, pengujian dilakukan untuk memastikan bahwa sistem berfungsi dengan baik dan bebas dari kesalahan. Pengujian ini mencakup berbagai teknik untuk memeriksa fungsionalitas sistem, integrasi antar komponen, dan keseluruhan kinerja aplikasi.

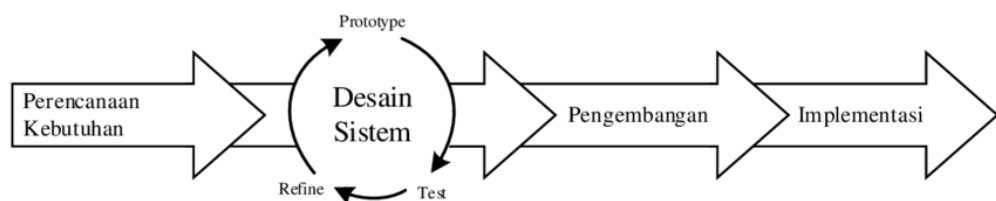
6) Implementasi (*Implementation*)

Setelah sistem diuji dan dinyatakan siap, tahap implementasi dimulai. Pada tahap ini, sistem dipasang dan digunakan di lingkungan produksi. Ini mencakup pelatihan pengguna, instalasi perangkat lunak, dan penyiapan infrastruktur yang diperlukan.

1) **Rapid Application Development (RAD)**

Rapid Application Development (RAD) adalah sebuah pendekatan dalam pengembangan perangkat lunak yang dirancang sebagai respons terhadap keterbatasan model pengembangan perangkat lunak tradisional seperti Waterfall, yang cenderung kaku dan memerlukan waktu lama untuk penyelesaian. RAD menawarkan pendekatan yang lebih fleksibel dan cepat, sehingga lebih sesuai untuk lingkungan yang dinamis dan membutuhkan hasil yang segera. Metodologi RAD menekankan pada kecepatan dan fleksibilitas dalam proses pengembangan perangkat lunak. Salah satu elemen utama dalam RAD adalah penggunaan prototipe untuk mengumpulkan umpan balik dari pengguna. Proses ini dilakukan melalui siklus iteratif yang memungkinkan pengembang untuk memperbaiki dan menyempurnakan sistem berdasarkan masukan pengguna.

Karakteristik utama dari RAD mencakup prototyping cepat, keterlibatan aktif pengguna, iterasi yang berulang, penggunaan komponen yang dapat digunakan kembali, serta fokus pada pengurangan waktu dan biaya pengembangan. Prototyping cepat memungkinkan pengembang untuk membuat versi sementara dari perangkat lunak yang dapat diuji oleh pengguna. Dengan keterlibatan aktif pengguna dalam setiap tahap pengembangan, RAD memastikan bahwa sistem yang dibangun benar-benar sesuai dengan kebutuhan dan harapan pengguna. Proses iteratif dalam RAD juga memainkan peran penting, di mana pengembangan dilakukan dalam siklus cepat yang berulang. Setiap iterasi melibatkan evaluasi dan penyempurnaan berdasarkan umpan balik yang diterima, hingga sistem akhir yang diinginkan tercapai. Selain itu, RAD sering kali memanfaatkan komponen perangkat lunak yang sudah ada dan dapat digunakan kembali, yang membantu mempercepat proses pengembangan.



Gambar 2. 1 Tahapan Metode RAD (Sumber: ResearchGate, Nurdian, F. D., & Rosid, M. A. (2024))

1) Object Oriented Analysis Design (OOAD)

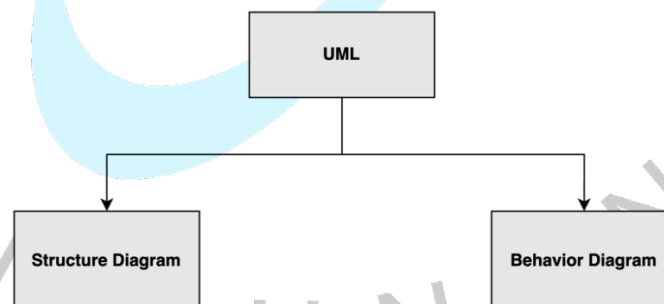
Object Oriented Analysis and Design (OOAD) merupakan pendekatan dalam memahami dan merancang sistem dengan mendasarkan pada konsep objek dari dunia nyata. Dalam pendekatan ini, objek mengintegrasikan data dan perilaku dalam satu kesatuan. OOAD mencakup dua aspek utama, yaitu *Object Oriented Analysis* (OOA) dan *Object Oriented Design* (OOD). OOA berfokus pada analisis kebutuhan sistem dengan mempertimbangkan objek dan kelas dalam ruang lingkup bisnis, sementara OOD lebih menitikberatkan pada perancangan arsitektur perangkat lunak dengan memanfaatkan manipulasi objek dalam sistem.

Dalam proses ini, UML (*Unified Modeling Language*) digunakan sebagai alat untuk memvisualisasikan, memodelkan, dan mendokumentasikan sistem berbasis objek.

2) *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk memodelkan sistem perangkat lunak. UML berfungsi dalam merancang dan mendokumentasikan struktur serta perilaku sistem dengan merepresentasikan berbagai komponen serta interaksi di antara mereka. Dalam proses pengembangan perangkat lunak, UML digunakan untuk memvisualisasikan, mendeskripsikan, dan menetapkan kebutuhan sistem secara lebih efisien.

UML juga merupakan alat yang digunakan untuk merepresentasikan serta mendokumentasikan hasil analisis dan desain sistem melalui sintaks yang memungkinkan pemodelan visual (Braun, et al., 2001). Selain itu, UML terdiri dari serangkaian konvensi pemodelan yang digunakan untuk mendefinisikan serta menggambarkan sistem perangkat lunak berbasis objek (Whitten, et al., 2004).



Gambar 2. 2 Jenis UML (*Unified Modeling Language*)

1. *Structure Diagram*

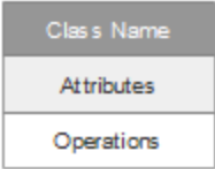

Structure diagram adalah jenis diagram dalam UML yang menggambarkan aspek statis dari sebuah sistem perangkat lunak, fokus pada

elemen-elemen yang membentuk struktur atau arsitektur sistem. Diagram ini memodelkan elemen-elemen seperti kelas, objek, komponen, dan hubungan di antara mereka, yang tidak berubah selama waktu berjalan. Structure diagram membantu memahami bagaimana berbagai bagian dari sistem disusun dan diorganisasikan. Diagram yang digunakan untuk menjabarkan struktur dan arsitektur sistem adalah *Class Diagram*.

a) *Class Diagram*

Class diagram adalah representasi grafis yang menunjukkan hubungan antar kelas serta detail dari setiap kelas dalam model desain suatu sistem. Diagram ini juga mengilustrasikan aturan serta tanggung jawab masing-masing entitas yang berkontribusi dalam menentukan perilaku sistem (Hendini, 2016). Oleh karena itu, class diagram dapat dianggap sebagai visualisasi struktur sistem program yang terdiri dari berbagai jenis kelas. Selain itu, diagram ini juga berperan dalam menggambarkan alur kerja database dalam sistem yang sedang dikembangkan atau akan dibangun.

Tabel 2. 1 Tabel Simbol Class Diagram

Simbol	Notasi	Keterangan
	Class	Merepresentasikan sebuah kelas dalam sistem, yang merupakan blueprint dari objek.
	Operation	Operasi atau metode yang dilakukan oleh kelas



2. Behavior Diagram



Behavior diagram dalam UML merupakan diagram yang merepresentasikan aspek dinamis suatu sistem, dengan menitikberatkan pada bagaimana sistem beroperasi, berinteraksi, serta merespons input atau

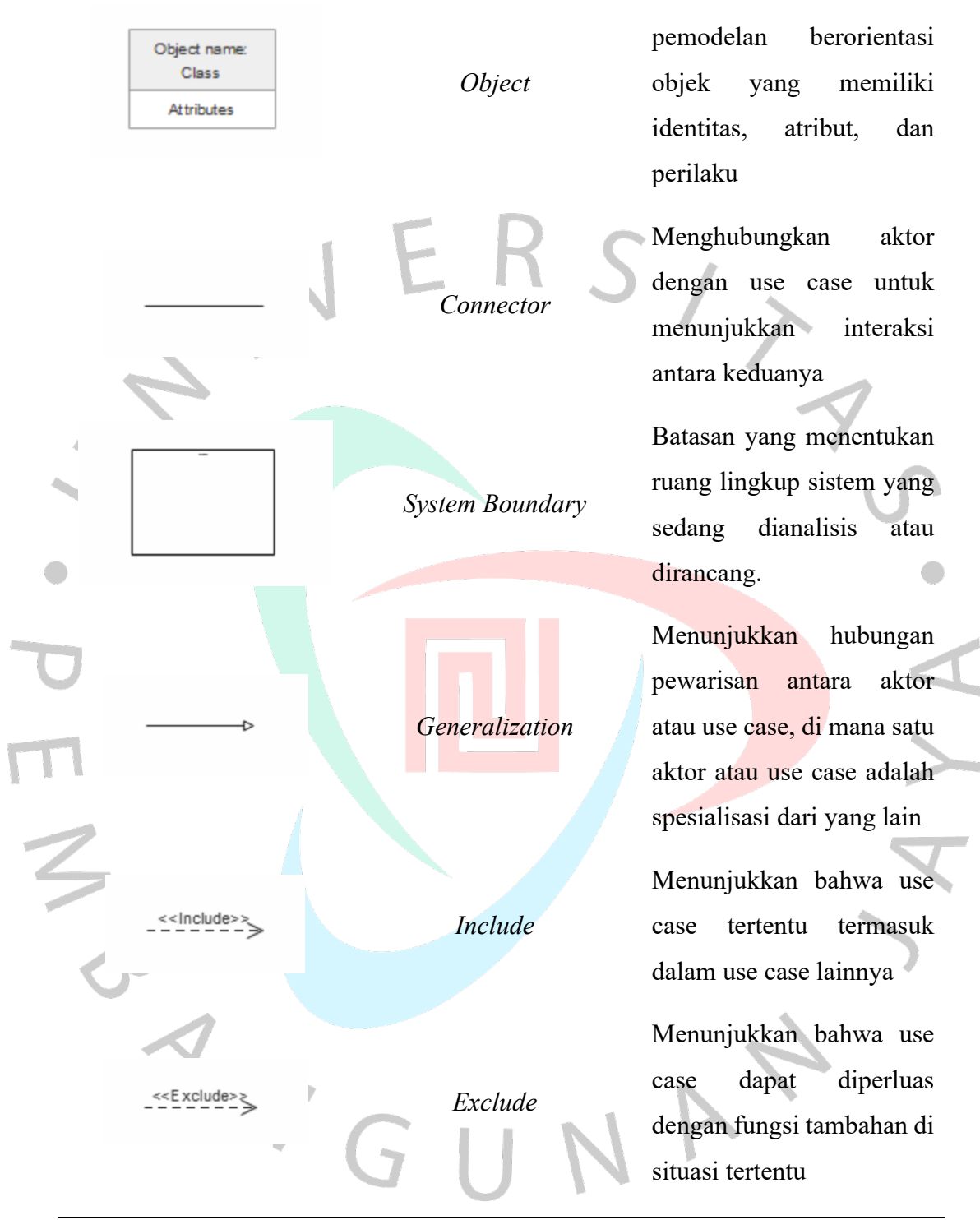
peristiwa. Diagram ini digunakan untuk memodelkan cara kerja dan interaksi antara objek atau elemen dalam sistem guna menjalankan fungsionalitasnya. *Behavior* diagram menggambarkan alur kerja, perubahan status, serta hasil dari penggunaan sistem seiring waktu. Beberapa diagram yang digunakan untuk menjelaskan fungsionalitas sistem antara lain *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

b) *Use Case Diagram*

Menurut Ahmad (2020), diagram *Use Case* merepresentasikan rangkaian interaksi yang saling terhubung antara sistem dan aktor. Diagram ini digunakan untuk menggambarkan jenis interaksi yang terjadi antara pengguna (user) suatu program dengan sistem itu sendiri. *Use Case* disajikan melalui skenario penggunaan yang menunjukkan bagaimana sistem dimanfaatkan. Selain itu, *Use Case* berfungsi untuk mendeskripsikan perilaku atau fungsi sistem yang akan dikembangkan, sekaligus menggambarkan hubungan interaksi antara pengguna (aktor) dengan sistem yang telah dirancang.

Tabel 2. 2 Tabel Simbol Use Case Diagram

Simbol	Notasi	Keterangan
	<i>Actor</i>	Mewakili pengguna eksternal atau sistem lain yang berinteraksi dengan sistem
	<i>Use Case</i>	Mewakili fungsi atau layanan yang diberikan oleh sistem kepada aktor



Entitas konkret dalam pemodelan berorientasi objek yang memiliki identitas, atribut, dan perilaku

Menghubungkan aktor dengan use case untuk menunjukkan interaksi antara keduanya

Batasan yang menentukan ruang lingkup sistem yang sedang dianalisis atau dirancang.

Menunjukkan hubungan pewarisan antara aktor atau use case, di mana satu aktor atau use case adalah spesialisasi dari yang lain

Menunjukkan bahwa use case tertentu termasuk dalam use case lainnya

Menunjukkan bahwa use case dapat diperluas dengan fungsi tambahan di situasi tertentu

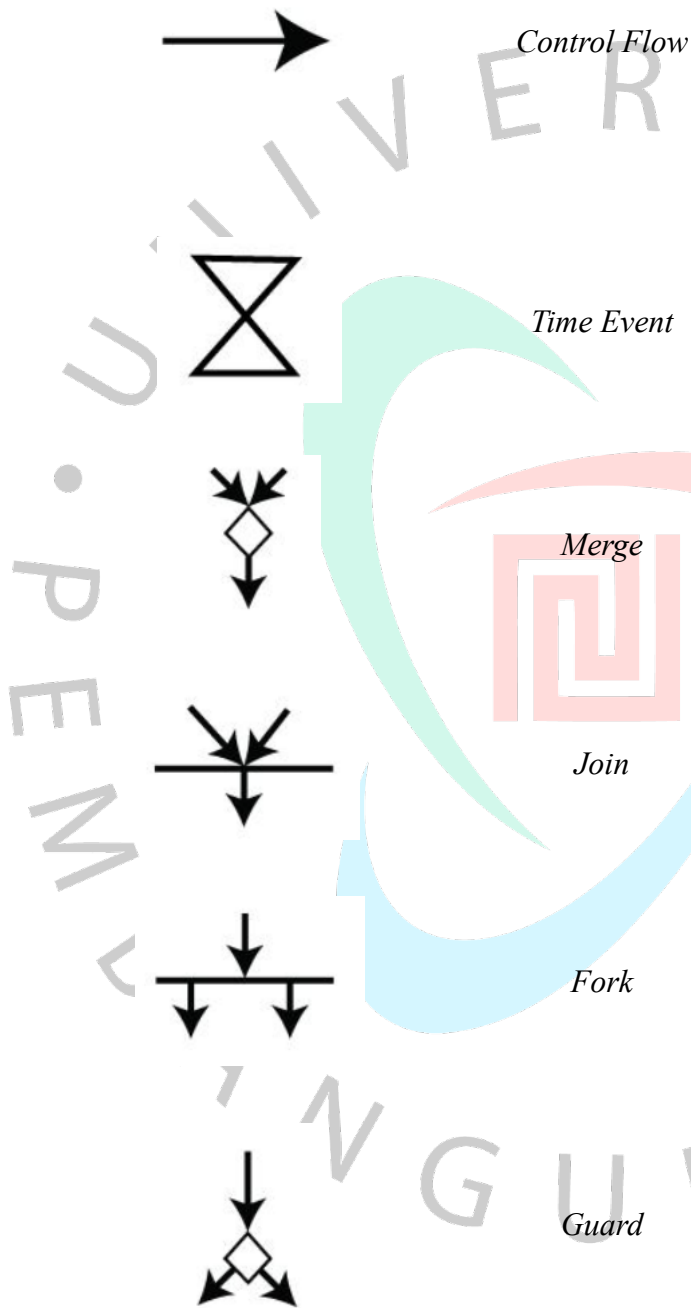
c) *Activity Diagram*

Activity Diagram dalam bahasa Inggris, adalah sebuah diagram yang berfungsi untuk memodelkan berbagai proses dalam suatu sistem.

Diagram ini menggambarkan urutan proses yang terjadi dalam sistem secara vertikal. Sebagai bagian dari diagram UML, *activity diagram* sering digunakan dalam pengembangan *Use Case*.

Tabel 2. 3 Tabel Simbol Activity Diagram

Simbol	Notasi	Keterangan
	<i>Activity State</i>	Menggambarkan tindakan yang dilakukan atau sedang berlangsung dalam sistem
	<i>Swimlane</i>	Membagi diagram aktivitas menjadi kolom atau baris untuk memisahkan tanggung jawab objek yang menjalankan aktivitas tertentu
	<i>Decision node</i>	Titik dalam diagram yang menunjukkan adanya pilihan, mengindikasikan bahwa terdapat beberapa kemungkinan atau jalur alternatif yang bisa diambil
	<i>Final State</i>	Menandakan bahwa proses telah selesai
	<i>Initial State</i>	Menunjukkan status awal, tindakan pertama, atau



titik mulai aktivitas dalam diagram aktivitas

Menunjukkan arah atau jalur yang diikuti oleh alur aktivitas, menghubungkan berbagai aktivitas, keputusan, atau acara dalam proses

Mewakili momen atau acara tertentu yang dipicu berdasarkan waktu

Menggabungkan alur yang telah dipecah menjadi beberapa bagian kembali ke dalam satu alur

Menyatukan kembali aktivitas yang sebelumnya berjalan paralel

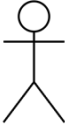
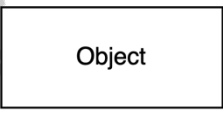


Membagi perilaku menjadi beberapa aktivitas atau aksi yang berjalan secara paralel

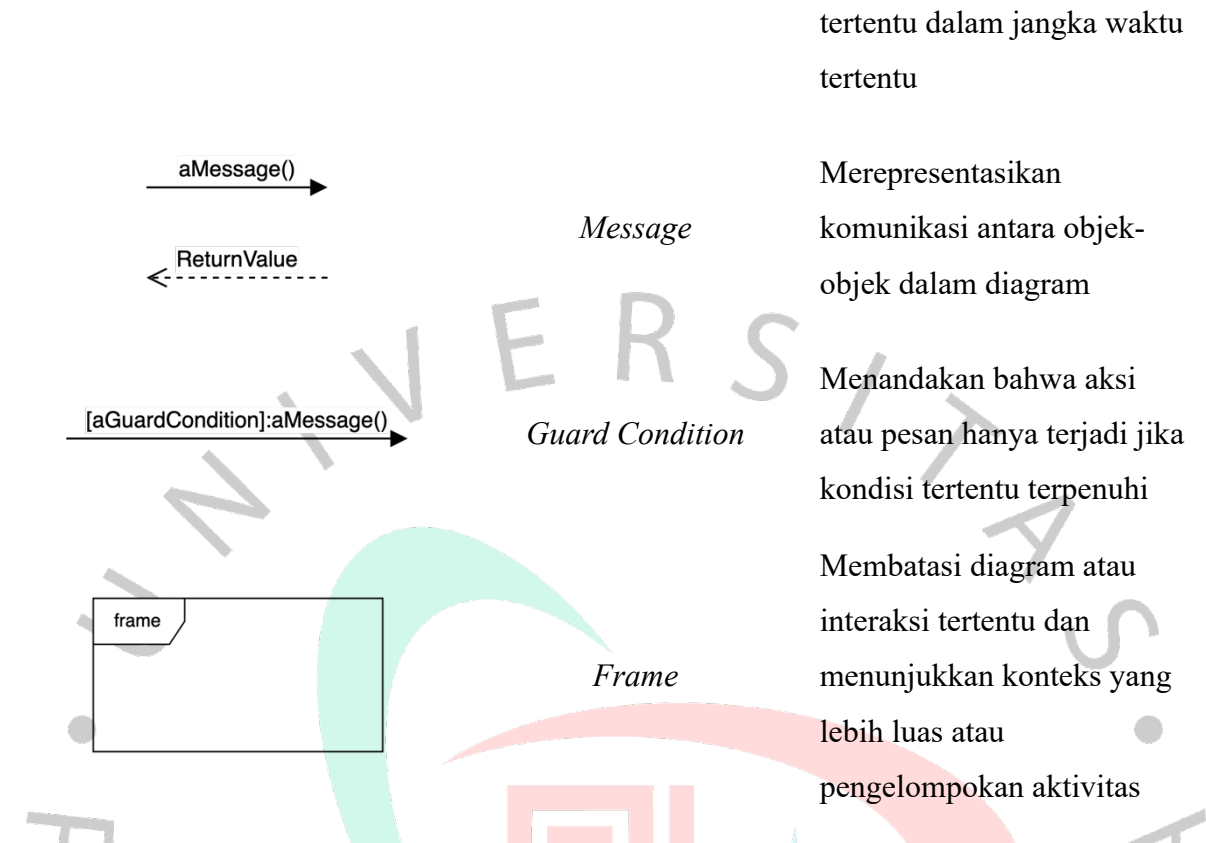
Titik keputusan yang mengarahkan alur ke jalur berbeda berdasarkan kondisi tertentu

d) *Sequence Diagram*

Sequence Diagram adalah diagram yang digunakan untuk menggambarkan interaksi antara berbagai objek. Diagram ini secara khusus menjelaskan perilaku dalam satu skenario tertentu. Diagram tersebut memperlihatkan sejumlah objek beserta pesan-pesan yang dikirimkan di antara objek-objek tersebut dalam sebuah *use case*. Selain itu, diagram ini menampilkan interaksi antar objek dalam sistem yang diorganisasikan berdasarkan urutan atau rangkaian waktu. Interaksi tersebut mencakup berbagai elemen, seperti pengguna, tampilan (*display*), dan elemen lain yang terlibat dalam pengiriman pesan.

Tabel 2. 4 Tabel Simbol Sequence Diagram

Simbol	Notasi	Keterangan
	<i>Actor</i>	Menggambarkan pengguna yang berada di luar sistem dan berinteraksi dengan sistem
	<i>Object</i>	Objek berfungsi untuk mencatat perilaku suatu objek dalam sistem
	<i>Lifeline</i>	Kotak yang berisi objek, digunakan untuk menggambarkan aktivitas dari objek tersebut
	<i>Execution</i>	Menunjukkan bahwa objek atau aktor sedang menjalankan aktivitas



tertentu dalam jangka waktu tertentu

Merepresentasikan komunikasi antara objek-objek dalam diagram

Menandakan bahwa aksi atau pesan hanya terjadi jika kondisi tertentu terpenuhi

Membatasi diagram atau interaksi tertentu dan menunjukkan konteks yang lebih luas atau pengelompokan aktivitas

C. Elisitasi

Elisitasi kebutuhan adalah proses untuk mengidentifikasi dan memahami secara menyeluruh kebutuhan sistem yang akan dikembangkan. Tujuan utama dari proses ini adalah memperoleh pemahaman yang jelas dan akurat mengenai kebutuhan bisnis maupun teknis sebelum sistem mulai dibangun. Proses ini dilakukan untuk menggali kebutuhan pengguna secara lebih rinci, sehingga sistem yang dihasilkan dapat benar-benar sesuai dengan harapan dan kebutuhan mereka. Tahapan elisitasi kebutuhan menjadi dasar dalam pengembangan sistem, dengan kemungkinan adanya perubahan atau penyesuaian pada setiap tahap sesuai dengan kebutuhan yang berkembang.

1) Elisitasi Tahap I

Pada tahap awal ini, seluruh rancangan sistem baru yang diajukan oleh pengguna dikumpulkan. Untuk mendapatkan pemahaman yang lebih mendalam mengenai kebutuhan serta fitur yang diinginkan, dilakukan wawancara dengan pihak terkait.

2) Elisitasi Tahap II

Tahap ini melibatkan proses pengelompokan kebutuhan yang telah dikumpulkan pada Tahap I dengan menerapkan metode MDI. Metode ini digunakan untuk memilah kebutuhan berdasarkan tingkat kepentingannya, yaitu kebutuhan yang harus dipenuhi, kebutuhan yang bersifat tambahan atau opsional, serta kebutuhan yang tidak memiliki keterkaitan langsung dengan sistem. Berikut adalah penjelasan lebih lanjut mengenai metode MDI:

- *M (Mandatory)*

Kebutuhan ini bersifat wajib dan harus ada dalam sistem baru.

- *D (Desirable)*

Kebutuhan ini tidak harus ada, tetapi jika dimasukkan akan menyempurnakan sistem.

- *I (Inessential)*

Kebutuhan ini tidak relevan atau tidak terkait langsung dengan sistem yang akan dikembangkan.

3) Elisitasi Tahap III

Pada tahap ini, kebutuhan yang termasuk kategori "I" di Tahap II dieliminasi. Kebutuhan yang tersisa kemudian diklasifikasikan lagi menggunakan metode TOE (*Technical, Operational, Economic*). TOE digunakan untuk menilai kelayakan kebutuhan berdasarkan beberapa kriteria:

- *Technical (T)*

Menilai bagaimana kebutuhan tersebut akan diterapkan secara teknis dalam sistem.

- *Operational (O)*

Menilai bagaimana kebutuhan tersebut akan berfungsi dan digunakan dalam sistem.

- *Economic (E)*

Menilai biaya yang diperlukan untuk mengimplementasikan kebutuhan tersebut.

Setiap kategori TOE kemudian dikelompokkan berdasarkan tingkat kesulitan pengerjaannya:

- *High (H)*

Kebutuhan sulit untuk dikerjakan karena teknisnya kompleks, penggunaannya rumit, atau biayanya tinggi, sehingga mungkin dieliminasi.

- *Middle (M)*

Kebutuhan yang dapat dikerjakan dengan tingkat kesulitan menengah.

- *Low (L)*

Kebutuhan yang mudah dikerjakan dengan biaya rendah dan teknis yang sederhana.

4) Elisitasi Tahap Final

Hasil akhir dari proses elisitasi ini berupa rancangan final yang menjadi acuan utama dalam pengembangan sistem yang akan dibangun. Pada tahap ini, seluruh kebutuhan yang telah dianalisis, disaring, dan dikategorikan ditetapkan sebagai kerangka kerja atau panduan utama bagi tim pengembang dalam merealisasikan sistem.

D. Database

Basis data, atau *database*, merupakan kumpulan informasi yang tersusun secara sistematis dan disimpan dalam komputer, sehingga dapat diakses melalui program untuk memperoleh informasi yang dibutuhkan. Pengelolaan data dalam database dilakukan dengan bantuan perangkat lunak khusus yang dikenal sebagai sistem manajemen basis data (DBMS). Secara konsep, database bertujuan untuk menyederhanakan interaksi antara pengguna dan sistem melalui abstraksi data, memungkinkan berbagai perspektif terhadap data, baik dari sisi pengguna, programmer, maupun administrator.

DBMS (Sistem Manajemen Basis Data) adalah perangkat lunak yang berfungsi sebagai penghubung antara pengguna dan database, memungkinkan pengelolaan data menjadi lebih terstruktur dan efisien. Dalam sistem database, terdapat dua jenis bahasa utama yang sering digunakan, yaitu *Database Definition Language* (DDL) dan *Database Manipulation Language* (DML). Beragam jenis DBMS tersedia untuk membantu pengguna dalam mengelola file di perangkat mereka. Sistem ini mendukung penyimpanan, pengelolaan, serta pengaturan data komputer secara lebih optimal. Selain itu, DBMS juga mempermudah akses, pengeditan, serta pemeliharaan data, sambil tetap memastikan keamanan dan integritas informasi yang tersimpan. Sistem ini menyediakan antarmuka bagi pengguna atau aplikasi untuk menjalankan query, menyimpan, dan mengambil data dengan mudah. Selain itu, DBMS juga mengelola transaksi dan mencegah konflik akses data dalam lingkungan yang digunakan oleh banyak pengguna.

- Beberapa contoh DBMS yang populer adalah Oracle, MySQL, Microsoft SQL Server, dan PostgreSQL, yang digunakan dalam berbagai aplikasi, baik di dunia bisnis maupun aplikasi web dan perangkat lunak.

Istilah database berasal dari ilmu komputer dan awalnya merujuk pada catatan yang disimpan dalam format digital. Namun, konsep ini telah ada jauh sebelum era komputer dalam bentuk buku besar atau dokumen bisnis lainnya. Database mengorganisasikan data berdasarkan skema yang menggambarkan hubungan antarobjek, biasanya menggunakan model relasional yang menyajikan data dalam bentuk tabel yang saling terkait. Model lain seperti model hierarkis dan jaringan juga digunakan, meskipun lebih kompleks dalam representasi hubungan antar data.

- **Struktur Database**

Struktur database merujuk pada cara data disusun dan diorganisir dalam database. Berikut adalah beberapa elemen utama dalam struktur database:

1. **Tabel (Table):** Tabel adalah komponen dasar dalam database yang menyimpan data dalam bentuk baris dan kolom. Setiap tabel memiliki nama unik, dan kolom di dalamnya menyimpan data tertentu.

2. **Kolom (Column):** Kolom mendefinisikan tipe data yang akan disimpan dalam tabel, seperti integer, teks, tanggal, dll. Setiap kolom memiliki nama dan tipe data tertentu.
3. **Baris (Row):** Baris adalah unit data yang disimpan dalam tabel. Setiap baris berisi satu set nilai yang sesuai dengan kolom dalam tabel.
4. **Kunci Utama (Primary Key):** Kunci utama adalah kolom atau kumpulan kolom yang secara unik mengidentifikasi setiap baris dalam tabel. Nilai kunci utama tidak boleh duplikat atau kosong.
5. **Kunci Asing (Foreign Key):** Kunci asing adalah kolom yang menghubungkan satu tabel dengan tabel lain. Ini merujuk pada kolom kunci utama di tabel lain, membentuk hubungan antar tabel.
6. **Indeks (Index):** Indeks digunakan untuk meningkatkan kecepatan pencarian data dalam tabel. Ini adalah struktur data tambahan yang menyimpan nilai kolom tertentu beserta referensi ke baris terkait.
7. **Relasi (Relation):** Relasi adalah hubungan antar tabel dalam database. Misalnya, tabel A bisa memiliki kunci asing yang menghubungkannya dengan tabel B.
8. **View:** View adalah tabel virtual yang dihasilkan dari hasil query. View tidak menyimpan data secara fisik, tetapi menyajikan data yang diambil dari satu atau lebih tabel.
9. **Skema (Schema):** Skema adalah struktur yang menggambarkan cara database disusun, termasuk tabel, kolom, relasi, dan aturan integritas data. Struktur ini membantu pengelolaan dan pemrosesan data secara efisien dalam sistem manajemen database (DBMS).

E. User Interface Design

Desain antarmuka pengguna (*User Interface*) merupakan proses merancang tampilan dan interaksi antara pengguna dengan sistem, baik dalam bentuk aplikasi desktop, web, maupun mobile. Tujuan utama dari desain UI adalah memastikan interaksi pengguna dengan sistem berlangsung secara efisien, efektif, dan nyaman. Dalam merancang UI, aspek visual dan fungsional sangat diperhatikan agar pengalaman pengguna (*User Experience*) menjadi optimal. Proses desain UI

sendiri terdiri dari beberapa tahapan. Pertama, dilakukan penelitian dan analisis pengguna (*user research*) untuk memahami kebutuhan, preferensi, dan pola perilaku pengguna yang akan menggunakan sistem. Setelah itu, langkah selanjutnya membuat wireframe, yaitu kerangka kasar yang menampilkan susunan elemen-elemen dasar dari antarmuka. Tahap berikutnya adalah *prototyping*, di mana prototipe fungsional dibuat untuk menguji interaksi pengguna dengan sistem.

F. Black Box Testing

Black box testing merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi fungsi eksternal sistem tanpa mengakses atau mengetahui struktur internal maupun kode program di dalamnya. Dalam metode ini, penguji hanya memperhatikan input yang dimasukkan dan output yang dihasilkan tanpa harus memahami proses internal yang terjadi dalam sistem. Pendekatan ini umum digunakan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan.

Black box testing bertujuan untuk memastikan bahwa setiap fungsi dalam perangkat lunak beroperasi dengan benar dan sesuai dengan kebutuhan pengguna sebagaimana dijelaskan dalam spesifikasi atau dokumentasi. Dalam metode ini, penguji tidak perlu memahami cara perangkat lunak dikembangkan atau diimplementasikan secara teknis, melainkan hanya mengamati bagaimana sistem berperilaku saat digunakan oleh pengguna.

2.3 Tinjauan Pustaka

1. RANCANG BANGUN APLIKASI *POINT OF SALE* (POS) STUDI KASUS MINI MARKET TRUSTMART

Pada jurnal penelitian berjudul "**RANCANG BANGUN APLIKASI *POINT OF SALE* (POS) STUDI KASUS MINI MARKET TRUSTMART**" yang disusun oleh Herlini Oktarina, Jaka Fitra menjelaskan bahwa Perkembangan teknologi informasi pada era globalisasi telah menjadi salah satu pendorong utama dalam transformasi berbagai sektor, termasuk industri ritel. Dalam operasional

minimarket, pencatatan transaksi secara manual sering kali menimbulkan ketidakefisienan serta meningkatkan kemungkinan terjadinya kesalahan data. Misalnya, pencatatan penjualan yang dilakukan secara manual tidak hanya memerlukan waktu lebih lama, tetapi juga berisiko menghasilkan informasi yang kurang akurat, yang dapat berdampak pada pengelolaan stok serta penyusunan laporan keuangan. Untuk mengatasi permasalahan ini, diperlukan solusi yang lebih efisien, salah satunya adalah penerapan aplikasi *Point of Sale* (POS) yang terotomatisasi. Aplikasi POS tidak hanya mempercepat proses transaksi, tetapi juga menyediakan metode yang lebih efektif dalam mengelola berbagai aspek bisnis, seperti pengelompokan barang, pencatatan data supplier, serta pembuatan laporan keuangan. Dengan adanya sistem yang saling terhubung, minimarket dapat meningkatkan akurasi, efisiensi operasional, serta daya saing di pasar yang semakin kompetitif.

Penelitian ini bertujuan untuk merancang dan membangun aplikasi POS berbasis web dengan studi kasus di minimarket Trustmart. Aplikasi ini dikembangkan untuk mengelola berbagai transaksi ritel, termasuk proses pembelian dan penjualan, serta menyediakan laporan yang dibutuhkan oleh manajemen. Dalam proses pengembangannya, digunakan metode Research and Development (RnD), yang meliputi analisis kebutuhan, perancangan sistem, pengembangan perangkat lunak, serta pengujian aplikasi. Dengan diterapkannya aplikasi POS, diharapkan tercipta sistem yang terintegrasi antara transaksi penjualan dan manajemen stok barang. Integrasi ini sangat penting untuk memastikan bahwa data yang dihasilkan akurat dan selalu diperbarui, sehingga dapat mendukung pengambilan keputusan yang lebih efektif bagi pemilik dan karyawan.

2. Rancang Bangun Aplikasi Point of Sales Distro Management System dengan Menggunakan Framework React Native

Dalam jurnal penelitian berjudul “*Rancang Bangun Aplikasi Point of Sales Distro Management System dengan Menggunakan Framework React Native*” yang ditulis oleh Putu Dika Arta Wigunaa, I Putu Agus Swastikaa, dan I Putu Satwikaa, dijelaskan bahwa perkembangan industri fashion remaja mengalami peningkatan

pesat, ditandai dengan munculnya berbagai produk inovatif dan menarik. Namun, banyak pelaku usaha distro masih mengandalkan metode manual dalam menjalankan bisnis mereka, yang menyebabkan berbagai kendala, seperti kesulitan dalam pencatatan transaksi penjualan, manajemen produk, serta pengecekan stok barang. Pengelolaan secara manual tidak hanya memakan waktu, tetapi juga meningkatkan risiko kesalahan dalam pencatatan data. Dengan meningkatnya permintaan konsumen, muncul kebutuhan mendesak akan solusi berbasis teknologi yang dapat meningkatkan efisiensi operasional, salah satunya melalui aplikasi Point of Sales (POS). Dalam industri fashion yang terus berkembang, distro menjadi salah satu bentuk usaha yang semakin diminati, khususnya oleh kalangan anak muda. Distro merupakan toko yang menjual produk fashion yang mengikuti tren terkini, sering kali bekerja sama dengan brand clothing atau band indie. Namun, banyak distro masih menghadapi tantangan dalam pengelolaan stok, pencatatan transaksi yang akurat, serta pembayaran kepada pihak clothing.

Penggunaan metode manual, seperti pencatatan di buku atau aplikasi sederhana seperti Microsoft Excel, sudah tidak lagi memadai untuk mengelola operasional yang semakin kompleks. Oleh karena itu, diperlukan sistem yang terintegrasi antara distro dan clothing untuk mempermudah manajemen bisnis. Salah satu solusi yang dapat diterapkan adalah sistem *Distro Management System* (DIMANS) berbasis web, yang dirancang untuk mendukung pencatatan transaksi dan pengelolaan data secara real-time. Dengan adanya sistem ini, proses penjualan dan pengelolaan produk dapat dilakukan lebih efisien, sehingga membantu distro dalam meningkatkan kinerja dan daya saingnya.

3. RANCANG BANGUN SISTEM INFORMASI POINT OF SALE DENGAN FRAMEWORK CODEIGNITER PADA CV POWERSHOP

Dalam jurnal penelitian berjudul “Rancang Bangun Sistem Informasi Point of Sale dengan Framework CodeIgniter pada CV Powershop” yang ditulis oleh Maydianto dan Muhammat Rasid Ridho, dijelaskan bahwa CV Powershop merupakan startup kecil yang bergerak di bidang printer dan komputer, yang menghadapi tantangan besar dalam pengelolaan inventaris. Saat ini, penggunaan metode manual dalam pencatatan sering menyebabkan ketidaksesuaian antara

jumlah stok aktual dengan data yang tercatat, sehingga menyulitkan pemantauan inventaris yang akurat. Permasalahan ini dapat berpengaruh terhadap kelancaran penjualan, tingkat kepuasan pelanggan, serta efisiensi operasional perusahaan secara keseluruhan. Untuk mengatasi tantangan ini, penulis mengusulkan pengembangan sistem informasi Point of Sale (POS) berbasis web. Sistem ini bertujuan untuk mengotomatiskan pengelolaan inventaris, menyederhanakan operasi, dan menyediakan data real-time mengenai ketersediaan stok. Dengan penerapan framework CodeIgniter, solusi yang diusulkan ini diharapkan dapat meningkatkan akurasi catatan inventaris dan memfasilitasi pengambilan keputusan yang lebih baik. Secara keseluruhan, tujuan utamanya adalah beralih dari metode manual ke pendekatan sistematis, sehingga meningkatkan efektivitas dan efisiensi pengelolaan inventaris di CV Powershop. Pemanfaatan teknologi komputer yang semakin berkembang telah mengubah berbagai aspek aktivitas manusia, terutama dalam lingkungan bisnis di mana sistem Point of Sale (POS) menjadi sangat penting. Sistem POS kritis untuk merekam transaksi secara akurat dan mengelola data penjualan, serta bergantung pada perangkat keras seperti PC dan printer agar berfungsi dengan baik. Di Indonesia, peran Usaha Mikro, Kecil, dan Menengah (UMKM) sangat signifikan, dengan mencakup sekitar 99,99% populasi usaha dan memberikan kontribusi besar terhadap perekonomian nasional. Meskipun penting, banyak UMKM masih bergantung pada proses manual dalam manajemen data, yang dapat menyebabkan ketidakefisienan dan kesalahan. Ketergantungan pada metode tradisional ini menghalangi bisnis dari memaksimalkan potensi mereka dan merespon dengan cepat terhadap tuntutan pasar. Oleh karena itu, pengenalan sistem informasi POS yang terkomputerisasi dianggap sebagai langkah penting untuk mengatasi ketidakefisienan ini.

4. RANCANG BANGUN APLIKASI POINT OF SALE (POS) BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER

Pada jurnal penelitian berjudul "RANCANG BANGUN APLIKASI POINT OF SALE (POS) BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER" yang disusun oleh Herdino Oktrianda menjelaskan bahwa Perkembangan pesat teknologi informasi telah berdampak

signifikan pada berbagai sektor, termasuk dunia usaha, terutama dalam bidang pertokoan. Banyak perusahaan mulai beralih dari sistem manual yang rentan terhadap kesalahan menuju sistem berbasis komputer yang lebih efisien dan akurat. Dalam konteks ini, Toko Aliamart yang berlokasi di Desa Cibunarjaya menjadi contoh nyata dari usaha yang masih menggunakan metode manual untuk mengelola operasionalnya. Toko ini menghadapi berbagai masalah dalam pencatatan transaksi, pengelolaan data barang, dan perhitungan stok. Saat ini, proses pencatatan transaksi di Toko Aliamart dilakukan secara manual menggunakan kertas, yang meningkatkan risiko kesalahan dalam penulisan dan data hilang atau rusak. Selain itu, pemilik toko harus mengandalkan kalkulator untuk menghitung transaksi, sehingga memperlambat proses pelayanan. Ketidakaktifan dalam memanfaatkan sistem informasi modern juga menyebabkan kesulitan dalam mendapatkan informasi yang akurat tentang stok barang yang tersedia. Oleh karena itu, penting untuk mengembangkan sebuah sistem *Point of Sales* (POS) yang berbasis web guna mengatasi permasalahan ini dan meningkatkan efisiensi operasional toko. Dalam pendahuluan, penting untuk memahami peran sistem informasi dalam memfasilitasi efisiensi operasional bisnis. Sistem POS dikembangkan dengan tujuan untuk meningkatkan kecepatan dan akurasi dalam melakukan transaksi jual beli. Sistem ini tidak hanya memungkinkan proses penjualan, tetapi juga terintegrasi dengan fungsi akuntansi dan manajemen inventaris, sehingga memudahkan pemilik toko dalam mengelola berbagai aspek usaha mereka. Adapun tujuan dari penelitian ini adalah untuk merancang aplikasi POS berbasis web yang dapat membantu pemilik Toko Aliamart dalam berbagai proses kunci, seperti pendataan barang, pembuatan nota transaksi, dan penghitungan hasil penjualan. Dengan adanya sistem yang terkomputerisasi ini, diharapkan pemilik toko dapat melakukan semua kegiatan operasionalnya dengan lebih cepat dan efisien.

5. RANCANG BANGUN APLIKASI POINT OF SALE BERBASIS ANDROID PADA MEBEL SELAPARANG

Pada jurnal penelitian berjudul "**RANCANG BANGUN APLIKASI POINT OF SALE BERBASIS ANDROID PADA MEBEL SELAPARANG**" yang disusun oleh Muhammad Yahya Jaber, Misbahuddin, dan L. A Syamsul Irfan Akbar menjelaskan bahwa Point of Sale (POS) adalah sistem yang sangat krusial

dalam mendukung kegiatan jual beli, terutama dalam konteks perdagangan modern. Sistem ini menawarkan berbagai manfaat, seperti efisiensi dalam proses transaksi dan keamanan data, yang menjadi sangat penting bagi para pedagang. Dalam industri mebel Selaparang, misalnya, penggunaan aplikasi POS dapat membantu mengelola data penjualan dan inventaris dengan lebih baik. Keberadaan sistem manual, yang masih digunakan oleh sebagian pengusaha, sering kali menyebabkan inefisiensi, risiko kehilangan data, dan kerugian akibat kerusakan fisik seperti nota kertas yang hilang atau terbakar. Aplikasi POS yang dikembangkan menggunakan platform Android hadir sebagai solusi inovatif. Dengan Android yang merupakan sistem operasi terbuka dan mudah diakses, aplikasi ini tidak hanya dapat digunakan oleh banyak orang, tetapi juga menawarkan fleksibilitas dalam pemasaran dan manajemen data. Selain itu, aplikasi ini memanfaatkan cloud computing untuk penyimpanan data, yang memungkinkan akses yang lebih aman dan terorganisir. Dengan fitur-fitur seperti metode CRUD (*Create, Read, Update, Delete*), aplikasi ini memudahkan pedagang dalam mengelola informasi mengenai barang dan transaksi, menjadikannya alat yang sangat diperlukan dalam meningkatkan efektivitas bisnis.

Sistem *Point of Sales* (POS) memiliki peranan yang sangat penting dalam dunia perdagangan, berfungsi sebagai alat yang membantu pedagang dalam mengelola berbagai aspek operasional bisnis mereka. Sebelum adanya sistem POS, sebagian besar usaha mengandalkan metode manual, seperti pembukuan tradisional dan penggunaan nota kertas. Metode ini tidak hanya inefisien, tetapi juga rawan terhadap kesalahan dan kehilangan data, yang dapat berdampak serius pada keberlangsungan usaha. Dalam konteks ini, penggunaan sistem POS menjadi solusi yang efektif, tidak hanya untuk penjual besar tetapi juga untuk pedagang kecil dan menengah. Dengan sistem POS, para pedagang dapat memantau setiap transaksi penjualan secara lebih efektif, mengetahui produk yang paling laku, dan mengelola stok barang dengan lebih baik. Data yang dihasilkan dari sistem ini memberikan informasi real-time yang diperlukan untuk pengambilan keputusan bisnis yang lebih cepat dan tepat. Meskipun saat ini masih banyak usaha yang menggunakan sistem desktop, kehadiran sistem berbasis Android menawarkan kepraktisan dan fleksibilitas yang tidak dapat ditawarkan oleh sistem tradisional.