



BAB 2 LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian oleh (Steven Sanjaya Putra, 2023) dengan judul Rancang Bangun Aplikasi *Point of Sale* Pada Kedai Kopi Elemen Kopi Berbasis Web. Sistem pencatatan transaksi yang dilakukan oleh Elemen Kopi masih menggunakan metode konvensional, Hal tersebut menyebabkan banyak kesalahan yang terjadi pada setiap proses transaksi yang dilakukan. Sistem yang dibuat adalah aplikasi point of sale yang dikembangkan menggunakan metode Rapid Application Development. dengan tujuan untuk menyediakan aplikasi kepada pemilik Elemen Kopi dalam melakukan pencatatan akuntansi baik transaksi jual beli maupun pengeluaran dan pemasukan barang. Hasil diuji menggunakan metode black-box dengan teknik equivalence partitioning menunjukan semua fitur pada sistem dapat berjalan sesuai dengan skenario yang diujikan.

Penelitian oleh (Gavin Haryanto Hutagalung, 2023) dengan judul Pengembangan *Back-End* Aplikasi Pendaftaran *English Proficiency Test* Widyatama Berbasis WEB. bertujuan untuk memudahkan pihak Lembaga Bahasa dalam melakukan pendataan, pengelolaan, dan pengolahan. serta memudahkan peserta dalam mendaftar EPT. metode yang digunakan adalah *waterfall* yang terdiri dari analisis, desain, implementasi, pengujian dan pemeliharaan. bahasa pemrograman menggunakan PHP dan database menggunakan MySQL. Hasil yang diperoleh adalah dalam pengolahan dan pengelolaan data menjadi efisien dan praktis.

Penelitian oleh (Fitri Rizky Wahyuni Siregar, 2023) dengan judul Sistem informasi Inventory Iparepart Motor Pada Bengkel AI Motor. memiliki masalah yaitu proses yang sedang berlangsung masih manual menggunakan microsoft excel,

membuat proses pencarian barang membutuhkan waktu yang lama. Tujuan dari dilakukannya penelitian ini yaitu membantu efektivitas pihak bengkel dalam pencatatan stok. Hasil penelitian ini berupa Aplikasi sistem informasi inventory sparepart yang dapat meningkatkan akurasi dalam kesesuaian antara data barang dengan jumlah fisik yang ada.

Penelitian oleh (Sofia Sarah, 2023) dengan judul E – Manajemen Butik Dengan Menggunakan Metode *Rapid Application Development*(RAD). dalam prosesnya perusahaan tersebut masih menggunakan cara manual dalam pencatatan stok barang, yaitu pengecekan secara satu-satu dan kemudian dicatat di buku. metode penelitian yang digunakan adalah *Rapid Application Development* (RAD). menggunakan metode iteratif dalam mengembangkan sistem dimana model bekerja sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan pengguna. Hasil penelitian berupa Aplikasi e-manajemen yang berhasil dibangun dan berjalan normal tanpa kendala menggunakan website sebagai media aplikasi.

Penelitian oleh (Muhammad Yasir Saan, 2023) dengan judul Rancang Bangun Aplikasi *Point of Sale* Penjualan Kopi Dengan Menggunakan Framework Codeigniter Berbasis WEB. memiliki beberapa kendala seperti menyimpan data transaksi masih menggunakan buku dan pencatatan stok barang kurang lengkap, dari permasalahan diatas penulis coba membuat aplikasi point of sale. Metode yang digunakan adalah *Waterfall* dengan tahapan analisis, desain, pengujian dan pemeliharaan. Hasil yang diperoleh berupa aplikasi *point of sale* yang dapat mempermudah perusahaan dalam pengelolaan barang dan kasir secara mudah dan efisien, serta memiliki pengelolaan laporan stock maupun transaksi yang rapi dan informatif.



Tabel 2.1 Perbandingan Penelitian Terdahulu

No	Judul	Teknologi	Metode	DBMS	Hasil
1	Rancang Bangun Aplikasi Point of Sale Pada Kedai Kopi Elemen Web(2023)	Web, Framework Laravel, PHP, Xampp web server	Rapid Application Development(RAD)	MySQL	aplikasi point ofsale pada kedai kopi Elemen Kopi yang dibuat berbasis web
2	PENGEMBANGAN BACK-END APLIKASI PENDAFTARAN ENGLISH PROFICIENCY TEST WIDYATAMA BERBASIS WEB(2023)	Web, Framework Codeigniter, Javascript, PHP	Waterfall	MySQL	adanya sistem pendaftaran dan pendataan peserta EPT berbasis web yang dapat mendukung bagi para peserta mengakses informasi. Hasil yang didapat juga untuk pihak Lembaga Bahasa adalah dalam pengolahan dan pengelolaan data menjadi efisien dan praktis.
3	Sistem informasi Inventory Sparepart Motor Pada Bengkel AI Motor(2023)	Web, Framework Codeigniter,	System Development Life Cycle (SDLC)	MySQL	Aplikasi sistem informasi inventory sparepart ini dapat menjadi media untuk meningkatkan akurasi dalam



			PHP			kesesuaian antara data barang dengan jumlah fisik yang ada
4	E – MANAJEMEN BUTIK DENGAN MENGGUNAKAN METODE APPLICATIO DEVELOPM	RAPID T(RAD)(2023)	Web, Framework Codeigniter, PHP	Rapid Application Development (RAD)		Aplikasi e manajemen berhasil dibangun dan berjalan normal tanpa kendala menggunakan website sebagai media aplikasi
5	RANCANG APLIKASI POINT OF SALE PENJUALAN MINGGUAN MENGGUNAKAN FRAMEWORK CODEIGNITER BERBASIS WEB(2023)	BANGUN POINT OF SALE MINGGUAN BERBASIS	Web, Framework Codeigniter, PHP, Xampp web server	Menggunakan Metode Waterfall	MySQL	Pada aplikasi point of sale yang telah.dibangun berhasil merancang pengeluaran perusahaan sehingga dapat mengetahui pengeluaran apa saja yang sudah terjadi

Dari Tabel 2.1 diatas diperoleh perbandingan penelitian :

Perbedaan penelitian ini dengan penelitian-penelitian diatas adalah penelitian ini memiliki fitur dashboard sehingga berguna untuk menampilkan informasi utama yang dibutuhkan oleh admin

Penelitian ini memiliki fitur yang dapat mengintegrasikan Pembelian Unit dan Penjualan Unit melalui Fitur Stok barang.

2.2 Kajian Pustaka

2.2.1 Moseli88



Gambar 2.1 Kantor Moseli88 Jombang

Pada gambar 2.1 diatas merupakan gambar dari Kantor Moseli88. Moseli88 adalah distributor sepeda listrik yang beralamat di Jl. Prof Buya Hamka No.26, Jombang, Jawa Timur. Distributor Moseli88 merupakan distributor Resmi dari produk Sepeda Listrik Pacific, Exotic & Agathos dan bertanggung jawab langsung kepada PT. Pasific Indah Pratama. Sebagai distributor langsung, kantor ini tidak hanya



menjual unit sepeda listrik saja, tetapi juga menjual suku cadang dari semua varian unit yang ada dan melayani jasa perbaikan apabila unit mengalami kendala setelah beberapa waktu pemakaian. Kantor ini tidak hanya melayani penjualan untuk perorangan saja, namun juga melayani penjualan pada toko-toko Retail dalam pulau jawa, maupun luar pulau jawa.

2.2.2 Rapid Application Development (RAD)

Rapid Application Development (RAD) merupakan metodologi yang muncul untuk menjawab masalah lambatnya penyelesaian pengembangan sistem informasi pada SDLC konvensional. Rapid application development (RAD) bertujuan untuk mengembangkan aplikasi secara cepat, melalui pengulangan dan *feedback* secara berulang.(Sarosa, 2017). *Rapid Application Development* (RAD) adalah model proses pengembangan *Software* yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. (Shalahuddin, 2018)

Fungsi RAD :

1. Mengembangkan software dengan cepat : Dengan RAD, prototype dibuat dengan cepat dan pengembangannya dilakukan secara berulang-ulang. Hal ini membuat proses pengembangan software semakin cepat dan efisien.
2. Memiliki fleksibilitas yang tinggi : RAD dirancang untuk mengakomodasi ketentuan pembuatan software yang berubah-ubah, sehingga metode ini ideal untuk proyek yang memerlukan fleksibilitas selama proses pengembangannya.
3. Membuat software yang sesuai kebutuhan pengguna : RAD dalam prosesnya melibatkan end users sehingga dapat memastikan bahwa software yang dikembangkan memenuhi kebutuhan dan harapan mereka.
4. Membuat software yang hemat biaya : RAD seringkali menggunakan kembali komponen software yang telah



dipakai sebelumnya, sehingga dapat mengurangi biaya pengembangan.

Tahapan RAD :

1. Menentukan persyaratan : developer dan tim terkait akan menggali informasi tentang apa yang dibutuhkan end user. Informasi dari pengguna software cukup penting supaya dapat memahami kebutuhan, preferensi, dan harapan mereka terhadap produknya. Tim pengembangan kemudian akan menerjemahkan informasi tersebut dan menjadikannya persyaratan yang berfungsi sebagai panduan pengembangan.
2. Membuat prototype : Prototype dibuat untuk diuji oleh pengguna supaya mereka dapat memberikan feedback tentang kegunaan, desain, dan fungsionalitasnya. Berdasarkan feedback yang diterima, tim developer kemudian akan membuat perubahan dan perbaikan yang diperlukan.
3. Konstruksi cepat : Pada fase ini, akan dikembangkan produk dalam serangkaian tahap, yaitu perancangan, coding, integrasi fitur, dan testing. Di tahap ini, pengguna juga dapat memberikan feedback untuk memastikan bahwa software-nya memenuhi kebutuhan dan harapan mereka.
4. Implementasi : Pada fase ini, tim developer akan menyelesaikan pengembangan software, melakukan testing akhir, dan memberikan training kepada pengguna untuk memakai software-nya. Kemudian, maintenance akan dilakukan supaya tidak ada bug dalam sistem dan dapat memastikan bahwa software tersebut fungsional dan efektif.

2.2.3 Framework

Framework adalah kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dikombinasikan dengan tujuan



mempermudah dan mempercepat proses pengembangan aplikasi web. (Raharjo, 2015). *Framework* merupakan koleksi atau kumpulan potongan program yang disusun atau diorganisasikan, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus menulis dari awal semua kodenya. (Hakim, 2010).

Fungsi Framework :

1. Membuat kode program menjadi lebih terstruktur : Framework biasanya memiliki pola arsitektur dalam menuliskan kode. Sehingga, kode yang dituliskan lebih mudah dan struktur.
2. Meningkatkan keamanan : framework dapat meningkatkan keamanan website. Seperti contohnya framework Laravel yang sudah mengadopsi berbagai sistem keamanan seperti autentikasi, enkripsi, dan hashing.
3. Mempercepat pembuatan website : framework dapat mempercepat pembuatan website. Hal itu karena pengembang dapat menggunakan komponen-komponen yang sudah disediakan dan tidak perlu menulis kode dari awal, sehingga dapat mempercepat pembuatan sebuah website.
4. Pemeliharaan dan perawatan website lebih mudah : framework dapat mempermudah dalam memperbaiki dan merawat website. Perbaikan bug, maintenance menambah fitur dan meningkatkan keamanan website akan jadi lebih mudah karena kebanyakan framework sudah menggunakan pola arsitektur yang beragam.

2.2.4 CodeIgniter

CodeIgniter adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para programmer web untuk membuat atau mengembangkan aplikasi berbasis web. (Sallaby & Kenedi, 2020). CodeIgniter merupakan *framework* web untuk



bahasa pemrograman PHP yang diciptakan oleh penemu dan pendiri EllisLab bernama Rick Ellis pada tahun 2006. EllisLab didirikan pada tahun 2002 merupakan Tim kerja yang bergerak dalam bidang pembuatan software dan tool untuk para web developer. (Budi Raharjo, 2015)

Fitur-Fitur CodeIgniter :

1. Model-View-Controller (MVC) : Model-View-Controller (MVC) adalah praktik standar industri yang memisahkan data, logika, dan presentasi dalam aplikasi web.
2. Library Bawaan : Fitur ini memungkinkan pengguna menemukan berbagai solusi seputar pemrograman, termasuk buku, tutorial, dan panduan penggunaan terkait CodeIgniter. Pengembang CodeIgniter memiliki akses ke dokumentasi lengkap yang menjelaskan cara kerja sistem aplikasi serta informasi lainnya seputar pengembangan PHP. Dalam library bawaan ini, tersedia tutorial atau petunjuk yang mencakup masalah umum yang pernah dihadapi oleh pengembang lain dan solusi yang ditemukan, sehingga bisa menjadi referensi bagi pengguna lain.
3. Sangat Ringan : Dalam era di mana kecepatan pemuatan aplikasi menjadi prioritas, CodeIgniter memberikan waktu pemuatan kurang dari satu detik setelah instalasi. Dengan respons cepat kurang dari 50 milisekon, CodeIgniter menjadi lebih cepat dibandingkan dengan beberapa framework lainnya.
4. Source Code yang Kompak : Ukuran source code yang kecil menjadi keunggulan, menyederhanakan implementasi dan pembaruan sistem aplikasi. CodeIgniter memiliki source code yang kurang lebih hanya 2 MB, memudahkan pengguna untuk memahami CodeIgniter dan cara kerjanya.

2.2.5 Perancangan Basis Data

1. *Database*



Database atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi. (Abdulloh, 2018). *Database* merupakan kumpulan data yang saling berhubungan secara sistematis dan dirancang untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.(Indrajani, 2015). Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah menyimpan data yang sudah diolah dan membuat informasi saat dibutuhkan. (Sukamto & Shalahuddin, 2015)

2. *Entity Relationship Diagram (ERD)*

Dalam pembuatan basis data, Entity Relationship Diagram (ERD) adalah diagram berupa grafis yang menunjukkan hubungan antara entitas (Pulungan, Febrianti, Lestari, Gurning, & Fitriana, 2023). Entitas adalah model data yang berhubungan dengan entitas lainnya, entitas tersebut akan dihubungkan dengan tanda garis pada sistem untuk menjelaskan hubungan yang ada di sistem database manajemen sistem. ERD digunakan untuk memodelkan relasional database yang didasarkan pada teori himpunan dalam matematika (Azhar Fadilah Zuhri, Ahmad, lin Parlina, Rafiqqa Dewi, & Solikhun Solikhun, 2020).



Simbol	Nama	Deskripsi
	Entitas	Suatu objek atau sesuatu yang dapat dibedakan atau didefinisikan dengan cara yang unik.
	Atribut	Karakteristik hubungan atau entitas yang memberikan penjelasan detail hubungan atau entitas.
	Relationship / Relasi	Menunjukkan hubungan yang terjadi antara satu atau lebih entitas.
	Link	Berfungsi sebagai penghubung entitas ke atribut dan entitas ke relasi.

Gambar 2.2 Simbol ERD

3. MySQL

MySQL adalah sistem yang berguna untuk melakukan proses pengaturan koleksi-koleksi struktur data (*database*) baik yang meliputi proses pembuatan atau proses pengelolaan *database*. (Rusli, 2019). MySQL merupakan satu dari kumpulan *software database server* yang umum digunakan. MySQL bersifat *Open Source* dan menggunakan SQL. (Subianto, 2019). MySQL merupakan *software RDBMS* (atau server database) yang dapat mengelola database dengan instan dan dapat menampung data dalam jumlah besar, dapat diakses oleh banyak pengguna (*multi user*), dan dapat melakukan suatu proses secara bersamaan (*multi-threaded*). (Aninditya, 2020)



2.2.6 Sistem Informasi

Sistem informasi merupakan gabungan dari berbagai komponen teknologi informasi yang saling bekerjasama dan menghasilkan suatu informasi guna untuk memperoleh satu jalur komunikasi dalam suatu organisasi atau kelompok. (Johny Seah,2020)

Macam – macam Sistem Informasi :

1. E-Commerce: Adalah sistem yang digunakan untuk melayani transaksi penjualan secara elektronik (umumnya bersifat online).
2. Sistem Informasi Eksekutif : Sistem yang digunakan untuk memberikan informasi yang dibutuhkan oleh fungsi eksekutif dari organisasi atau perusahaan.
3. Sistem Pakar : Sistem yang berisi pengetahuan dari suatu bidang keahlian seolah-olah sistem adalah seorang pakar.
4. Sistem Informasi Strategik : Sistem informasi yang dirancang untuk mencapai tujuan tertentu, seperti memperoleh keunggulan kompetitif, mengurangi kelemahan kompetitif, atau memenuhi tujuan strategik perusahaan atau organisasi.
5. Intelegensi Bisnis : Kumpulan dari aplikasi-aplikasi berfokus pada pengumpulan dan analisis data, yang merupakan informasi yang digunakan untuk membuat keputusan strategik untuk perusahaan atau organisasi.
6. Customer Relationship Management : Sistem yang bertujuan untuk meningkatkan proses yang berpusat pada layanan pelanggan dengan tujuan memberikan layanan yang lebih baik, termasuk hal-hal yang disenangi oleh pelanggan.
7. Supply Chain Management : Sistem yang mengelola hubungan antara proses bisnis penting dengan vendor, pelanggan, dan mitra bisnis lainnya.
8. Enterprise Resource Planning : Sistem yang menggabungkan dan mengotomasikan berbagai proses bisnis dalam organisasi untuk meningkatkan efisiensi dan



ketangkasan dalam menangani berbagai macam situasi yang berbeda yang ada di perusahaan atau organisasi. ERP adalah kombinasi dari kemampuan CRM dan SCM.

2.2.7 Desain *User Interface*

User Interface merupakan suatu bentuk dari komunikasi. *User Interface* yang baik dapat meningkatkan kemudahan penggunaan, mengurangi tingkat kejadian error, dan memberikan pengalaman pengguna dan stakeholder yang baik (Rokhmawati, Evantio, & Saputra, 2019). Pengalaman pengguna yang menyenangkan dan mudah dipahami sangat dipengaruhi oleh *User Interface* (UI) (Danang Tejo Kumoro & Valian Yoga Pudya Ardhana, 2023).

Metode Desain *User Interface* :

1. Metode User Centered Design
UCD didefinisikan oleh User Experience Professionals Association (UXPA) sebagai pendekatan desain dengan proses yang didasarkan pada informasi tentang orang-orang atau pihak-pihak yang menggunakan produk atau sistem tersebut (Danang Tejo Kumoro & Valian Yoga Pudya Ardhana, 2023).
2. Balsamiq
Balsamiq merupakan aplikasi yang digunakan untuk menggambar wireframe untuk tampilan antarmuka perangkat lunak. Balsamiq menyediakan banyak komponen-komponen antarmuka yang sederhana sehingga designer tidak perlu terlalu fokus dalam detail yang menyita banyak waktu dan bisa fokus dengan rancangan fungsional antarmuka aplikasi. Balsamiq adalah aplikasi berbayar.



2.2.8 Unified Modeling Language (UML)

UML adalah seperangkat diagram, struktur, dan teknik untuk memodelkan dan merancang program dan aplikasi berorientasi objek (kroenke, 2018). UML sendiri merupakan pemodelan Bahasa yang berisikan notasi, dimana simbol simbol yang akan digunakan pada model serta aturan aturan yang akan menuntun bagaimana akan menggunakannya (Susianto D, 2019). UML digunakan sebagai notasi untuk bermacam kegiatan, seperti memodelkan kasus bisnis, analisa bentuk sistem, dan arsitektur desain awal (Rumpe, 2017).

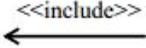
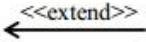
Use Case Diagram

Use Case Diagram adalah diagram yang digunakan untuk menggambarkan interaksi pada sebuah sistem informasi yang menggambarkan satu atau lebih aktor. *use case diagram* berguna untuk mengetahui siapa saja yang dapat menggunakan fitur fitur yang ada pada sistem dan fitur fitur yang ada pada sitem tersebut (Suhimarita dan Susianto, 2019).

Pada Gambar 2.2 merupakan beberapa Komponen *Use Case* antara lain :

- *Actor*, suatu entitas yang akan menggunakan atau digunakan oleh sistem.
- *Connection*, suatu penghubung yang menghubungkan antara actor ke *use case*.
- *Relationship*, suatu hubungan antara aktor itu sendiri dengan *use case*.



Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan use case
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

Gambar 2.3 Komponen Use Case Diagram

Activity Diagram.

Activity Diagram adalah diagram yang digunakan untuk Penggambaran alur kerja perangkat lunak dari dari suatu proses bisnis. *activity diagram* adalah diagram yang menyerupai obrolan bagan alur horizontal yang menunjukkan tindakan dan peristiwa saat terjadi (Tilley & Rosenblatt, 2016). *Activity diagram* menggambarkan aktivitas pada sistem, bukan apa yang dilakukan aktor, kapan aktivitas dapat dilakukan pada sistem (Suhimarita & Susianto, 2019). Pada Gambar 2.3 merupakan beberapa Simbol pada *Activity Diagram* Antara lain :

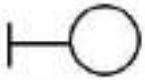
Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2.4 Simbol Activity Diagram

Sequence Diagram

Sequence Diagram berguna untuk menggambarkan interaksi antar objek selama jangka waktu tertentu. Karena pola interaksi bervariasi dari satu *use case* ke yang lain, setiap *sequence diagram* hanya menunjukkan interaksi yang berkaitan dengan *use case* yang spesifik (Valacich & Gerogr, 2016). *Sequence Diagram* mewakili interaksi terperinci antara aktor dan sistem atau antara objek yang berkolaborasi dalam blok waktu tertentu (Unhelkar, 2018).

Pada Gambar 2.4 merupakan beberapa simbol *Sequence Diagram* antara lain :

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

Gambar 2.5 Simbol *Sequence Diagram*

Class Diagram

Class diagram mendefinisikan class-class yang akan dibuat untuk membangun sistem. Kelas memiliki atribut atau properti dan metode atau operasi. Atribut adalah variabel-variabel yang dimiliki oleh suatu class, dan metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas (None Khana Wijaya, None Rishi Suprianto, & None Endi Istiawan, 2022).

Pada Gambar 2.5 merupakan beberapa Simbol pada *Class Diagram* Antara lain :

Simbol	Nama	Deskripsi
	Kelas	Kelas yang ada di sistem.
	Antarmuka / Interface	Interface dalam pemrograman OOP
	Asosiasi / association	Hubungan antar kelas yang bersifat umum
	Asosiasi berarah / directed association	Hubungan antar kelas yang menunjukkan bahwa suatu kelas digunakan oleh kelas lain.
	Generalisasi / Inheritance	Menunjukkan bahwa suatu kelas adalah spesialisasi dari suatu kelas lainnya. Ini juga bermakna bahwa suatu kelas adalah child dari sebuah parent class.
	Ketergantungan / dependency	Relasi bahwa suatu instance dari kelas tersebut bergantung pada suatu instance dari kelas lain untuk bisa ada.
	Agregasi / aggregation	Relasi antarmuka dengan makna semua-bagian, menunjukkan bahwa suatu kelas adalah kumpulan dari kelas lain.

Gambar 2.6 Simbol Class Diagram