

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini penulis menggunakan beberapa tinjauan pustaka untuk mendukung penelitian yang sedang dilakukan. Berikut adalah tinjauan pustaka yang digunakan penulis yang dapat dilihat di tabel 2.1.

Tabel 2.1 Daftar Literatur

No	Peneliti	Judul	Tahun
1	(Yusa Rahman et al., 2020)	Perancangan Aplikasi Servis Headphone Berbasis WEB Pada Centra Phone Banjarmasin.	2020
2	(Wahyuni, Hady, Estiyanti et al., 2021)	Analisis Dan Perancangan Sistem <i>Point of Sales</i> Menggunakan Metode <i>Agile Development</i> Pada Toko Eka Putra Sukawati.	2021
3	(Utomo et al., 2023)	<i>Point Of Sales</i> Menggunakan Metode <i>Agile Development</i> Pada Bengkel Mandala Motor.	2023
4	(Yulianti&Rahmawati, 2022)	Pengembangan Sistem <i>Forecasting</i> Penjualan Pada Aplikasi <i>Point of Sales</i> Menggunakan Metode <i>Trend Least Square</i> .	2022
5	(Junifer Pangaribuan et al., 2022)	Pengembangan Sistem Informasi <i>Point of Sales</i> untuk UMKM studi kasus: ARJUNA FARM.	2022

Tabel 2.2 Daftar Literatur (Lanjutan)

No	Peneliti	Judul	Tahun
6	(Umar et al., 2019)	Perancangan Database <i>Point of Sales</i> Apotek Dengan Menerapkan Model Data Relasional.	2019
7	(Grawidi Yuarita & Marisa, n.d. 2017)	Perancangan Aplikasi <i>Point of Sales</i> (POS) Berbasis WEB Menggunakan Metode Siklus Hidup.	2017
8	(Dwi Praba1 & Santoso, n.d. 2023)	Pengembangan Aplikasi <i>Point of Sales</i> Menggunakan Metode Agile Dengan Pola Scrum.	2023
9	(Amin & Pasha, 2023)	Penerapan Metode <i>Design Thinking</i> dan <i>Agile</i> dalam Rancang Bangun Aplikasi Penjualanku.	2023
10	(Bin Tahir et al., 2019)	Aplikasi <i>Point OF Sales</i> Menggunakan <i>Framework Laravel</i> .	2019

2.1.1 Literatur 1

Oleh Yusa Rahman et al, Zayid Musiafa, Kholik Setiawan dari Jurusan Sistem Informasi, Universitas Islam Kalimantan MAB Banjarmasin, dengan judul Perancangan Aplikasi Servis *Headphone* Berbasis WEB Pada *Centra Phone* Banjarmasin. Hasil dari penelitian ini bertujuan agar memberikan kemudahan baik administrator, teknisi, pemilik toko maupun pelanggan untuk melakukan proses serah terima barang, maupun proses pencarian data pelanggan secara cepat dan mudah. Aplikasi servis *handphone* dikelola oleh administrator, kemudian dilanjutkan oleh teknisi untuk mengelola data servis dan pemilik toko melihat data

laporan. Aplikasi servis *handphone* ini dibuat menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai pengelola basis datanya. Penggunaan aplikasi sangat memudahkan untuk mendata semua data servis dan menyimpan nota servis apabila sewaktu-waktu dibutuhkan.

2.1.2 Literatur 2

Oleh Ayu Sri Wahyuni Jelantik, Ni Made Estiyanti dari Jurusan Sistem Informasi Akutansi dan Putu Trisna hady permana dari Jurusan Teknik Informatika STMIK Primakara Bali, dengan judul Analisis Dan Perancangan Sistem *Point of Sales* Menggunakan Metode *Agile Development* Pada Toko Eka Putra Sukawati. Dalam penelitian ini telah berhasil dilakukan, sehingga dapat menghasilkan laporan penjualan, laporan data pelanggan dan laporan data barang secara tepat waktu/*real time* serta transaksi *point of sales* sesuai dengan harapan dan meningkatkan kinerja perusahaan agar efektif dan efisien. Adapun saran yang disampaikan dalam rancangan sistem ini adalah dengan dengan melakukan penambahan pada bagian penjualan berupa sistem pemesanan secara online tanpa perlu datang ke toko.

2.1.3 Literatur 3

Oleh Andriansyah Utomo, Agung Triayudi, Ira Diana Sholihati dari Jurusan Infomatika, Universitas Nasional Jakarta Selatan Indonesia dengan judul *Point Of Sales* Menggunakan Metode *Agile Development* Pada Bengkel Mandala Motor. Setelah melakukan penelitian ini, didapatkan hasil berupa sistem *Point of Sales* yang dapat mengelola penjualan dan pembelian ke pemasok, menampilkan laporan persediaan barang, dan menampilkan laporan pendapatan dan pengeluaran penjualan sebagai laba rugi. Aplikasi ini juga telah dievaluasi dengan menggunakan metode *Software Usability Scale* (SUS), dimana rata-rata dari 200 responden yaitu 76,6% termasuk dalam kategori tingkat penerimaan aplikasi yang dapat diterima, kategori tingkat kelas berada pada posisi C, dan kata sifat kategori peringkat berada di posisi dapat diterima sehingga dapat disimpulkan bahwa aplikasi layak digunakan oleh pengguna dan telah terpenuhi. Diantara saran untuk pengembangan ke depan adalah penambahan fitur notifikasi pengingat janji temu *customer service* berlangganan untuk memudahkan proses *booking* untuk melakukan *service*

kendaraan. Sistem ini semoga bisa digunakan juga diuji kinerjanya, hingga efektivitas sistem bisa diketahui.

2.1.4 Literatur 4

Oleh Ita Yulianti, dari Jurusan Sistem Informasi Akutansi, Universitas Bima Sarana Informatika dan Ami Rahmawati Dari Jurusan Sistem informasi dari Universitas Nusa Mandiri dengan judul Pengembangan Sistem *Forecasting* Penjualan Pada Aplikasi *Point of Sales* Menggunakan Metode *Trend Least Square*. Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan bahwa pengembangan metode *trend least square* sangat tepat digunakan dalam pengembangan sistem forecasting karena memiliki kemampuan peramalan yang sangat baik. Hal ini dibuktikan dengan nilai error (MAPE) yang dihasilkan hanya sebesar 0,0067% saja. Selain itu, secara umum pengembangan sistem *forecasting* pada aplikasi POS ini juga dapat membantu menunjang kegiatan pelayanan terhadap pelanggan dan memantau perkembangan income serta perencanaan persediaan menu yang akan dijual

2.1.5 Literatur 5

Oleh Jefri Junifer Pangaribuan, Okky Putra Barus, Yudhistira A.Pratama Faisal Nadjar, Ade Maulana Dari Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan dengan judul Pengembangan Sistem Informasi *Point of Sales* untuk UMKM studi kasus: ARJUNA FARM. Berdasarkan hasil proses yang didapat dengan menggunakan implementasi untuk peternakan Arjuna Farm dalam *Point of Sales* (PoS) sangat berguna dan membantu dalam mencatat orderan barang dan penjualan barang dengan detail yang lengkap. Dalam pengembangan sistem informasi yang dilakukan dapat berguna untuk semua pelaku UMKM yang sedang menjalankan bisnis untuk keperluan dalam meningkatkan kinerja penjualan dan memberi efisiensi dalam pekerjaan. Dengan sistem yang dibangun juga membantu dalam menyimpan data agar aman di database dan mengurangi kesalahan atau resiko yang akan muncul kedepannya.

2.1.6 Literatur 6

Oleh Rusydi Umar, Abdul Hadi, Pangah Widiandana, Fahmi Anwar Muhammad Jundullah, Ahmad Ikrom Dari Jurusan Program Studi Teknik Informatika, dan Program Studi Magister Teknik Informatika Universitas Ahmad Dahlan dengan judul Perancangan Database *Point of Sales* Apotek Dengan Menerapkan Model Data Relasional. Penelitian ini menghasilkan entitas yang diperlukan sebanyak 8 entitas, berisikan keterangan dan kegiatan entitas sebagai desain konseptual kemudian penelitian ini memiliki desain logikal yang menghasilkan relasi model dan logikal untuk mempresentasikan entitas, relasi dan attribute yang telah diidentifikasi sebelumnya.

2.1.7 Literatur 7

Oleh Titania Grawidi Yuarita, Fitri Marisa, dari Jurusan Teknik Informatika, Universitas Widyagama Malang dengan judul Perancangan Aplikasi *Point of Sales* (POS) Berbasis WEB Menggunakan Metode Siklus Hidup. Berdasarkan pembahasan penelitian ini maka diperoleh kesimpulan sebagai Aplikasi *Point of Sale* yang dibangun ini adalah dari hasil survey yang dilakukan pada perusahaan yang sedang berkembang dan dalam melakukan transaksi masih menggunakan cara yang manual. Sehingga sering kali terjadi kesalahan dan kehilangan faktur pembelian dan penjualan. Aplikasi yang dibuat ini digunakan untuk meminimalisir terjadinya kesalahan dan kehilangan faktur pembelian dan penjualan, meyimpan data dengan lengkap, dan memudahkan pengelola untuk melakukan transaksi pengecekan stok roti, harga produksi, data penjualan, data pembelian dan pengecekan rugi dan laba perusahaan.

2.1.8 Literatur 8

Oleh Ardian Dwi Praba, Tri Santoso dari Jurusan Sistem Informasi, Universitas Nusa Mandiri Jaktarta Timur dengan judul Pengembangan Aplikasi *Point of Sales* Menggunakan Metode *Agile* Dengan Pola *Scrum* Tahap awal perancangan sebuah system informasi memiliki peran yang sangat penting untuk dalam menentukan keberhasilan sebuah aplikasi. Proses bisnis yang jelas akan

membuat pekerjaan menjadi lebih efektif dan cepat. Disamping itu, metode pengembangan sistem scrum juga membantu pengembang aplikasi dalam menyelesaikan perubahan kebutuhan sistem yang ada. Dengan sumber daya yang terbatas dan proses bisnis yang banyak dan sering terjadi perubahan alur maupun tampilan pada saat browser *development*, metode *scrum* mampu untuk mempermudah dalam pembagian pekerjaan atau menganalisa proses pembuatan aplikasi sampai selesai.

Kedepan metode *Agile* akan semakin sering digunakan khususnya untuk pengembangan aplikasi yang sifatnya kompleks dan dinamis. Walaupun proses pengembangan sering terjadi penambahan dan pengurangan, dokumentasi aplikasi tetap harus dilakukan agar pengembangan aplikasi bisa berjalan dengan baik.

2.1.9 Literatur 9

Oleh Zaid Amin, Nabila Pasha dari Jurusan Sistem Informasi, Universitas Bima Darma, Palembang dengan judul Penerapan Metode *Design Thinking* dan *Agile* dalam Rancang Bangun Aplikasi Penjualanku. Penelitian ini menghasilkan aplikasi yang sudah dianalisis masalah dan kebutuhannya, berdasarkan studi kasus *anindya collection* dalam proses pembukuan data penjualan. Aplikasi ini dapat menyimpan data penjualan dan produk secara terintegrasi dan terorganisir, mengolah dan menyajikannya serta dapat merekap data penjualan tersebut menjadi data penjualan. Aplikasi yang dibangun berbasis website agar pengguna tidak harus melakukan penginstalan aplikasi. Aplikasi ini juga dijalankan online yang dapat menjadi solusi atas keterbatasan tempat dan waktu akses data dan informasi penjualan yang selama ini berbentuk fisik.

2.1.10 Literatur 10

Oleh Tamus Bin Tahir, Muh. Rais, Moch. Apriyadi HS dari Jurusan Fakultas Teknik Dan Informatika, Universitas Patria Artha sama Program Studi Teknik Elektro, Fakultas Teknik, Universitas Khairun dengan judul Aplikasi *Point OF Sales* Menggunakan *Framework Laravel* dalam penelitian ini Aplikasi point of sales sangat membantu pemilik toko Galaxy Cell dalam manajemen data toko. Penggunaan *framework laravel* sangat membantu programmer dalam membangun

sebuah aplikasi. Hal ini tentunya karena karena *syntax laravel* yang bersih dan fungsional serta *library* yang banyak dan mudah digunakan sehingga dapat mempercepat pembangunan modul Hasil pengujian yang dilakukan dengan menggunakan metode *Black-Box testing* mengimpulkan bahwa fungsi yang dibuat pada modul dapat berfungsi dengan baik.

Berdasarkan Sepuluh *literatur* yang dijelaskan diatas bahwa dapat dilihat yang menjadi pembeda antara penelitian terdahulu dengan penelitian yang sekarang sedang berjalan adalah mengelola pelayanan servis *headphone* secara akurat dan melakukan pencatatan transaksi penjualan sparepart *headphone* menggunakan *point of sales*, memberikan informasi status servis kepada pelanggan secara efektif dan cepat sehingga meminimalisir kesalahan petugas pada toko Comshop.

2.2 Point Of Sales

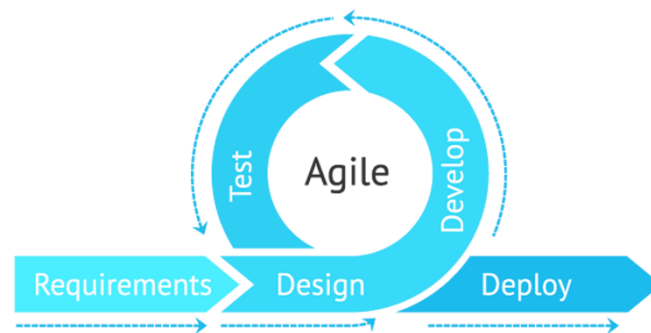
Sistem POS adalah sistem yang diperlukan untuk memperbarui dari proses manual (Fernandes Andry et al., 2019). Berdasarkan penelitian yang dilakukan oleh (Naufal, Nawangnugraeni and Suseno, 2022), *Point of Sales* atau sering dikatakan sistem POS yaitu, kegiatan jual beli dengan menggunakan sistem informasi. Bagi suatu organisasi Manfaat dari sistem *Point of Sales* adalah untuk mengolah transaksi-transaksi, mengurangi biaya dan menghasilkan pendapatan sebagai salah satu produk atau pelayanan mereka (Galang Ramadhan et al., 2020).

manfaat dari *Point of Sales* adalah mudah melakukan proses *controlling* semua transaksi karena laporan disediakan dengan cepat, mempermudah perusahaan melihat hasil penjualan dan pengambilan keputusan jika di butuhkan. Adapun fitur utama dari PoS adalah, Fitur pencatatan transaksi pembelian dan penjualan, fitur pencatatan stok barang, fitur pembuatan laporan. Sedangkan fitur tambahan bisa

berupa pembuatan struk transaksi untuk *customer*, penambahan data *customer* yang dapat di manfaat kan sebagai kebutuhan *marketing*.

2.3 Metode Agile Development

Agile Development yang merupakan salah satu metode pengembangan yang dimana digunakan sebagai bahan untuk berinteraksi secara intensif kepada klien dalam melakukan proses perancangan sistem (Wahyuni, Hady, Estiyanti, 2021). Metode *agile* merupakan metodologi pengembangan perangkat lunak yang berlandaskan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembangan terhadap perubahan yang ada.



Gambar 2.1 Proses Pengembangan Agile Development

Pengembangan Agile didasarkan pada "*Manifesto for Agile Software Development*" yang dirilis pada tahun 2001 oleh sekelompok praktisi pengembangan perangkat lunak. *Manifesto* ini menggaris bawahi empat nilai inti (Hikmah et al., 2021):

1. Individu dan Interaksi Lebih dari Proses dan Alat: Menekankan pentingnya kolaborasi dan komunikasi yang efektif antara anggota tim,
2. Perangkat Lunak yang Bekerja Lebih dari Dokumentasi yang Lengkap: Lebih mengutamakan pengembangan perangkat lunak yang dapat digunakan dan memberikan nilai nyata daripada hanya fokus pada dokumentasi yang terperinci,
3. Kerjasama dengan Pelanggan Lebih dari Negosiasi Kontrak: Mengedepankan kerjasama yang erat dengan pelanggan atau pemangku kepentingan untuk

memastikan perangkat lunak yang dikembangkan sesuai dengan kebutuhan dan ekspektasi mereka,

4. Merangkai Perubahan Lebih dari Mengikuti Rencana: Mengakui bahwa kebutuhan dan prioritas dapat berubah seiring waktu, sehingga perangkat lunak harus dapat beradaptasi dengan cepat terhadap perubahan tersebut.

2.4 Analisis Kebutuhan (*Requirement Analysis*)

Pada tahap ini, pengembang menggunakan hasil analisis kebutuhan user untuk merumuskan sistem yang akan dibangun. Analisa ini dapat dilakukan dengan melakukan wawancara atau dengan melihat kegiatan bisnis dan sistem yang sudah ada. Penulis menggunakan teknik analisis kebutuhan seperti *Business Process Model and Notation* (BPMN) dalam penelitian ini.




2.5 BPMN (*Business Process Modeling Notation*)

Business Process Modeling Notation (BPMN) adalah notasi grafis yang menggambarkan logika dari langkah-langkah dalam proses bisnis (Fitriawati, 2004). Notasi ini telah didesain secara khusus untuk mengkoordinasikan urutan proses dan pesan yang mengalir antara peserta dalam kegiatan yang berbeda. Untuk menyediakan notasi yang dapat dengan mudah dipahami oleh semua pengguna bisnis. Termasuk adalah analisis bisnis yang menciptakan draf awal dari proses sampai pengembang teknis yang bertanggung jawab untuk mengimplementasikan teknologi yang digunakan untuk menjalankan proses- proses tersebut.


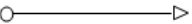
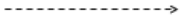
Terdiri dari macam-macam *Business Process Modeling Notation*:

1. *Flow objects*
2. *Connecting objects*
3. *Swimlanes*
4. *Artifacts*



Tabel 2.3 Simbol-simbol *Flow objects*

No	Simbol	Nama	Keterangan
1		<i>Event</i>	Menunjukkan kejadian dari suatu kegiatan di mulai dari <i>Start</i> , <i>Intermediate</i> , <i>End</i> .
2		<i>Activity</i>	Menunjukkan Aktivitas yang dilakukan oleh suatu proses bisnis
3		<i>Gateway</i>	Fungsi untuk menunjukkan alur percabangan atau pilihan seperti <i>Forking</i> , <i>Merging</i> , <i>Joining</i>



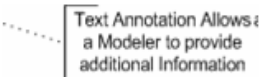
Tabel 2.4 Simbol-simbol *Connecting objects*

No	Simbol	Nama	Keterangan
1		<i>Sequence flow</i>	Menunjukkan Urutan aktivitas yang dilakukan dalam suatu proses
2		<i>Message flow</i>	Menunjukkan alur pesan yang terjadi antara pelaku proses
3		<i>Association</i>	Menunjukkan input atau output yang diperlukan oleh suatu aktivitas

Tabel 2.5 Simbol-simbol *Swimlanes*

No	Simbol	Nama	Keterangan
1		<i>Pool</i>	Merepresntasikan pelaku suatu aktivitas
2		<i>Lane</i>	Pengelompokan pelaku yang lebih detail

Tabel 2.6 Simbol-simbol *Artifacts*

No	Simbol	Nama	Keterangan
1		<i>Data object</i>	Menunjukkan data apa yang dibutuhkan oleh suatu aktivitas, atau data apa yang dihasilkan oleh suatu aktivitas.
2		<i>Group</i>	Tujuannya untuk lebih memudahkan dalam menganalisis proses dan pendokumentasian.
3		<i>Annotation</i>	Merupakan komentar atau deskripsi singkat untuk memperjelas diagram

2.6 Website

Situs *web*, juga dikenal sebagai situs *web*, adalah teknologi internet yang dapat digunakan sebagai sarana utama untuk memberikan informasi kepada khalayak melalui penyimpanan data dan informasi melalui *Uniform Resource Locator* (URL) (Bilqis Nabilla dan Filda, 2020.) Informasi dan Data terdiri dari suara, gambar, teks, dan video interaktif.

Website statis dan dinamis berbeda berdasarkan jenis kontennya. *Website* statis memiliki halaman yang tetap dan tidak berubah kecuali diubah oleh pengembang atau pemilik aplikasi. Informasi di dalamnya juga bersifat satu arah dan dibuat oleh pengembang atau pemilik aplikasi. *Website* dinamis, di sisi lain memiliki konten yang terus diupdate. Update biasanya terjadi pada halama. Sebuah aplikasi *web* juga terdiri dari berbagai bagian, seperti

2.6.1 Hyper Text Markup Language (HTML)

Dengan menggunakan HTML, yang merupakan bahasa penandaan yang terdiri dari banyak tag, konten sebuah website ditulis dan diatur secara strukturnya.

Selain itu, *World Wide Web Consortium* (W3C) mendefinisikan dan mengawasi HTML sebagai standar internet.

2.6.2 Cascading Style Sheet (CSS)

Jika HTML adalah kerangka Karena tujuan CSS adalah untuk meningkatkan tampilan situs web, ia dapat digambarkan sebagai kulit atau pakaian dari sebuah website. Selain itu, CSS dapat digabungkan dengan *Framework* untuk meningkatkan kinerjanya.

2.6.3 Javascript

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web lebih dinamis dan interaktif. Memiliki fitur seperti berorientasi objek, *client-side*, *high-level programming*, dan *loosely typed* (Dicoding Intern, 2020). *JavaScript* diciptakan oleh Brendan Eich pada tahun 1994 dan mulai dikenal. Pada awalnya, bahasa ini disebut *Mocha*, tetapi kemudian berubah dengan nama seperti *Mona* dan *LiveScript* sebelum akhirnya dikenal dengan nama *JavaScript*.

2.6.4 PHP

PHP (*HyperText PreProcessor*) merupakan bahasa pemrograman yang di proses di server, Fungsi utama PHP dalam membangun website adalah untuk melakukan pengelolaan data dalam database (Tumini & Fitria, 2021). PHP (*HyperText PreProcessor*) merupakan suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat di mengerti oleh komputer yang bersifat *server side* yang dapat di tambahkan ke dalam HTML

2.6.5 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal, *MySQL* menggunakan bahasa SQL untuk mengakses database nya (Sofwan, 2003). Lisensi *Mysql* adalah *FOSS License Exception* dan ada juga yang versi komersial nya. Tag *Mysql* adalah “*The World's most popular open source database*”. *MySQL* tersedia untuk beberapa *platform*, di antara nya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap *Mysql*, anda dapat

menggunakan *software* tertentu, di antara nya adalah *php myadmin* dan *mysql yog*. Pada kesempatan kali ini, kita akan menggunakan *php myadmin*, yang terdapat dalam *bundle xampp*, yang dapat di peroleh di www.apachefriends.org.

2.7 UML (*Unified Modeling Language*)

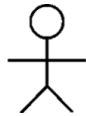


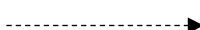
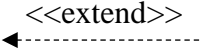
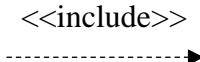
Unified Modeling Language (UML) adalah "bahasa" yang telah menjadi standar industri untuk visualisasi, desain, dan dokumentasi sistem perangkat lunak, UML menyediakan standar untuk merancang model sistem (Dharwiyanti & Wahono, 2003).

Dengan menggunakan *UML*, kita dapat membangun model untuk semua jenis aplikasi perangkat lunak, dan aplikasi ini dapat berjalan di perangkat keras apa pun, sistem operasi apa pun, jaringan apa pun, dan ditulis dalam bahasa pemrograman apa pun. Karena *UML* juga menggunakan kelas dan operasi dalam konsep dasarnya, *UML* paling cocok untuk menulis perangkat lunak dalam bahasa berorientasi objek seperti *C++*, *Java*, *C#*, atau *VB.NET*. Namun, *UML* masih dapat digunakan untuk memodelkan aplikasi prosedural di *VB* atau *C*. Beberapa diagram-diagram yang ada pada *UML* yaitu:

2.7.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Penekanannya adalah pada "apa" yang dilakukan sistem, bukan "bagaimana" (Dharwiyanti & Wahono, 2003). Sebuah *use case* merepresentasikan interaksi antara aktor dan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya, masuk ke sistem, membuat daftar belanja, dan sebagainya. Aktor adalah unit manusia atau mesin yang berinteraksi dengan sistem, untuk melakukan tugas-tugas tertentu. Berikut simbol yang digunakan dalam *use case* diagram dapat dilihat di tabel 2.9:

Tabel 2.7 Simbol-simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang, sistem, atau alat dalam berkomunikasi dengan <i>use case</i>
2		<i>Use Case</i>	Abstraksi dan interaksi antara sistem dan aktor
3		<i>Association</i>	Abstraksi hubungan antara subjek dan <i>use case</i>
4		<i>Generalisasi</i>	Menunjukkan kemampuan aktor untuk berpartisipasi dalam <i>usecase</i>
5		<i>Extend</i>	Menunjukkan bahwa <i>use case</i> adalah tambahan fungsional ke <i>use case</i> lain jika kondisinya benar
6		<i>Include</i>	Menunjukkan bahwa satu <i>use case</i> sepenuhnya bergantung pada <i>use case</i> lain

2.7.2 Class Diagram

Class Diagram merupakan spesifikasi yang, ketika dipakai, membuat objek dan merupakan inti dari desain dan pengembangan berorientasi objek. Class menggambarkan keadaan (atribut/properti) dari suatu sistem dan penyediaan layanan untuk memanipulasi keadaan (metode/fungsi). Diagram kelas menggambarkan struktur dan deskripsi kelas, paket, dan objek serta hubungannya satu sama lain, seperti: *Containment*, *inheritance*, *association*, dan lain-lain.

Class memiliki tiga area pokok:



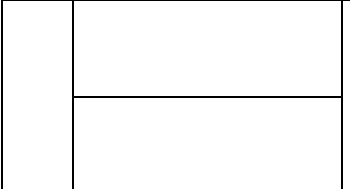
1. Nama (dan *stereotype*),
2. Atribut,
3. Metoda.

Atribut dan metoda dapat memiliki salah satu sifat berikut:


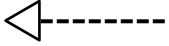


- a. *Private*, tidak dapat dipanggil dari luar kelas yang *relevan*,

- b. *Protected*, hanya class yang relevan dan anak-anak diwarisi yang dapat diakses,
- c. *Public*, bisa dipanggil siapa saja.

Tabel 2.8 Simbol-simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Association</i>	Yang menghubungkan satu objek dengan objek lainnya
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3		<i>Class</i>	Kumpulan objek dengan atribut dan operasi yang sama

Tabel 2.9 Simbol-simbol *Class Diagram* (Lanjutan)

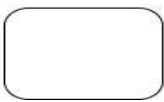
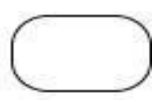



No	Simbol	Nama	Keterangan
4		<i>Collaboration</i>	Deskripsi urutan kegiatan yang dilakukan oleh sistem, yang memberikan hasil yang terukur bagi aktor
5		<i>Realization</i>	Operasi yang sebenarnya dilakukan oleh objek
6		<i>Dependency</i>	Hubungan dimana perubahan pada elemen independen akan mempengaruhi elemen yang bergantung pada elemen yang tidak independen.
7		<i>Generalization</i>	Hubungan di mana objek anak (keturunan) berbagi perilaku dan struktur data objek yang berada di atas objek induk (leluhur)

2.7.3 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang dirancang, bagaimana setiap aliran dimulai, keputusan apa yang dapat dibuat, dan bagaimana mereka berakhir (Dharwiyanti & Wahono, 2003). *Activity diagram* juga dapat menggambarkan proses bersamaan yang dapat terjadi dalam beberapa eksekusi.

Activity diagram merupakan diagram status kustom di mana sebagian besar status adalah tindakan dan sebagian besar transisi dipicu oleh akhir status sebelumnya (pemrosesan internal). Oleh karena itu, diagram aktivitas tidak secara akurat menggambarkan perilaku internal sistem (dan interaksi antar subsistem), melainkan menggambarkan proses dan jalur aktivitas dari tingkat yang lebih tinggi secara keseluruhan.

Tabel 2.10 Simbol-simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		<i>Activity</i>	Menunjukkan bagaimana setiap kelas antarmuka berinteraksi satu sama lain
2		<i>Action</i>	Status sistem yang mencerminkan eksekusi tindakan
3		<i>Initial Node</i>	Bagaimana objek dibuat atau diinisialisasi
4		<i>Activity Final Node</i>	Bagaimana benda dibuat dan dihancurkan
5		<i>Extend</i>	Satu aliran pada waktu tertentu berubah menjadi banyak aliran

2.8 Pengujian *Black Box*

Black box testing adalah metode pengujian terhadap sistem berdasarkan pada kebutuhan fungsional dan kebutuhan non fungsional karena tidak mengacu pada inti sistem seperti hasil dari kode program dan database yang telah dirancang (Nur Ichsanudin dan Uminingsih et al., 2022). Pengujian ini diharapkan dengan tujuan sistem dapat berfungsi sesuai kegunaannya dan dapat berjalan dengan baik.

Pengujian kotak hitam adalah pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini memiliki beberapa teknik yang dapat digunakan. *Black box* testing ini juga memiliki kelebihan misalnya, penguji tidak harus mengerti bahasa pemrograman untuk membangun perangkat lunak. Selain itu, penguji juga tidak harus memeriksa keseluruhan kode.

Namun, pengujian ini juga memiliki kelemahan. Karena penguji tidak perlu mengetahui serta memeriksa kode, ada peluang terdapat kesalahan yang tidak terdeteksi pada bagian kode.