

BAB II

LANDASAN TEORI

2.1 Konsep Sistem Informasi

Dalam organisasi, sistem informasi memberikan dukungan bagi keseluruhan organisasi. Sistem informasi membantu berbagai tugas operasional dalam administratif maupun keputusan strategis, penggunaan sistem informasi dapat membantu suatu organisasi dalam mencapai tujuan dan meningkatkan kinerja organisasi secara keseluruhan (Simanungkalit, 2019). Dalam sektor pertanian, sistem informasi dapat membantu dalam pengelolaan data pertanian dan sumber daya manusia. Sistem informasi juga dapat digunakan untuk meningkatkan efisiensi, produktivitas, dan keberlanjutan dalam sektor pertanian, serta membantu petani dan pihak terkait dalam mengambil keputusan yang lebih baik berdasarkan informasi yang akurat dan terkini (Sari, 2017).

2.1.1 Pengertian Sistem

Pengertian sistem dalam konteks sistem informasi merujuk pada sistem yang terdiri dari komponen-komponen yang saling terkait dan bekerja sama untuk mengumpulkan, menyimpan, mengolah, dan menyebarkan informasi guna mendukung pengambilan keputusan, pengendalian, dan operasi suatu organisasi (Wiranataa, 2021).

Dengan demikian, sistem digunakan untuk memahami hubungan antara komponen-komponen sistem dan bagaimana komponen-komponen tersebut saling berinteraksi untuk mencapai tujuan sistem. Dalam pengembangan sistem informasi, pendekatan sistem dapat membantu meningkatkan efektivitas, efisiensi, kualitas, fleksibilitas, dan pengambilan keputusan dalam penggunaan sistem informasi.

2.1.2 Pengertian Informasi

Informasi adalah data yang telah diolah dan diinterpretasikan sehingga memiliki makna dan relevansi bagi pengguna (Ahmad & Munawir, 2018:7). Informasi dapat digunakan untuk membantu pengambilan keputusan, memberikan wawasan baru, atau meningkatkan pemahaman tentang suatu topik atau situasi. Dalam konteks sistem informasi, informasi adalah output dari sistem yang dihasilkan dari pengolahan data oleh perangkat lunak dan perangkat keras dalam sistem informasi. Informasi yang dihasilkan oleh sistem informasi dapat berupa laporan, grafik, tabel, atau bentuk lain yang dapat membantu pengguna dalam mengambil keputusan atau memahami situasi yang sedang dihadapi (Ihsan, 2022). Informasi yang dihasilkan oleh sistem informasi harus akurat, relevan, dan dapat diandalkan agar dapat memberikan manfaat bagi pengguna dan organisasi.

Oleh karena itu, penting untuk memperhatikan kualitas data dan proses pengolahan data dalam pengembangan sistem informasi agar informasi yang dihasilkan dapat memenuhi kebutuhan pengguna dan organisasi.

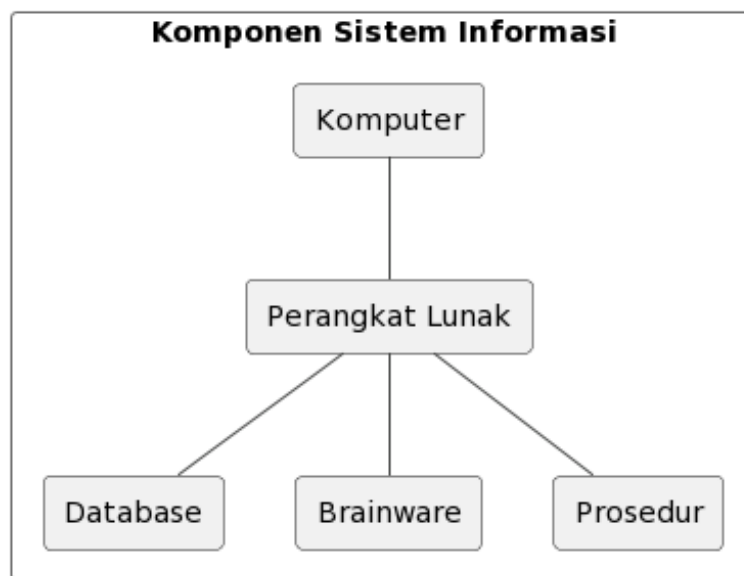
2.1.3 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem yang terdiri dari orang-orang, peralatan, dan prosedur-prosedur yang digunakan untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi yang dibutuhkan untuk mendukung operasi suatu organisasi dimana sistem informasi dapat membantu dalam mengelola data dan informasi secara efisien dan efektif, sehingga dapat mempermudah berjalannya suatu proses (Anggraeni & Irviani, 2017). Dalam konteks teknologi informasi, sistem informasi dapat digunakan untuk mengolah data menjadi informasi yang berkualitas sehingga berguna untuk pengambilan keputusan dimana kombinasi antara teknologi informasi dan aktivitas orang yang menggunakan teknologi tersebut untuk mendukung operasi dan manajemen dalam sebuah organisasi (Febyanita, 2019).

Dengan demikian, sistem informasi memiliki peran krusial dalam mengolah data menjadi informasi yang berkualitas sehingga berguna untuk pengambilan keputusan. Dalam era teknologi informasi saat ini, penerapan sistem informasi di sektor pertanian dapat membantu para petani meningkatkan produktivitas dan mengoptimalkan penggunaan sumber daya yang ada.

2.1.4 Komponen Sistem Informasi

Dalam pengembangan sistem informasi, penting untuk memperhatikan dan mengintegrasikan komponen-komponen sistem sehingga informasi dapat berjalan dengan efektif dan efisien, Komponen-komponen tersebut saling berinteraksi dan bekerja sama untuk menciptakan sistem informasi yang berfungsi dengan baik (Bratha, 2022). Komponen-komponen sistem informasi dapat mencakup.



Gambar 2. 1 : Komponen Sistem Informasi

1. Komputer
Komputer merupakan salah satu komponen penting dalam sistem informasi. Komputer digunakan untuk mengumpulkan, menyimpan, mengolah, dan menyebarkan informasi dalam sistem informasi.
2. Perangkat Lunak

Perangkat lunak atau software merupakan komponen yang berperan dalam mengelola dan memproses data dalam sistem informasi. Perangkat lunak dapat berupa aplikasi atau program yang dirancang khusus untuk memenuhi kebutuhan sistem informasi.

3. Database

Database adalah komponen yang digunakan untuk menyimpan data dalam sistem informasi. Database menyediakan struktur yang terorganisir untuk menyimpan dan mengelola data yang diperlukan oleh sistem informasi.

4. Brainware

Brainware atau manusia adalah komponen yang tidak kalah penting dalam sistem informasi. Manusia berperan sebagai pengguna, pengelola, dan pengambil keputusan dalam sistem informasi. Keterampilan, pengetahuan, dan pengalaman manusia sangat mempengaruhi efektivitas dan efisiensi penggunaan sistem informasi.

5. Prosedur

Prosedur adalah langkah-langkah atau aturan yang harus diikuti dalam penggunaan sistem informasi. Prosedur membantu mengatur dan mengontrol penggunaan sistem informasi agar berjalan dengan baik dan sesuai dengan kebutuhan organisasi.

2.2 Kelompok Tani

Menurut Peraturan Menteri Pertanian Nomor 82/Permentan/OT.140/8/2013, tentang Pedoman Pembinaan Kelompok Tani dan Gabungan Kelompok Tani, pengertian kelompok tani adalah kumpulan petani/peternak/pekebun yang dibentuk atas dasar kesamaan kepentingan, kesamaan kondisi lingkungan sosial, ekonomi, dan sumberdaya, kesamaan komoditas, dan keakraban untuk meningkatkan dan mengembangkan usaha anggota. Kelompok tani merupakan entitas informal yang memiliki peran penting dalam suatu wilayah terdiri dari para petani yang bersatu dengan tujuan bersama untuk meningkatkan produksi pertanian guna mendapatkan pendapatan yang lebih baik serta kesejahteraan yang lebih tinggi (Keraf & Feka, 2023). Dalam konteks ini, pembinaan kelompok tani di desa Sidapdap Simanosor

menjadi suatu aspek krusial dalam membangkitkan serta mengembangkan potensi dan kemandirian kelompok tani. Kelompok tani muncul dari keterikatan dan keselarasan antar anggotanya, serta dipandu oleh kepentingan bersama dalam pemanfaatan sumber daya pertanian. Kolaborasi ini menjadi landasan untuk meningkatkan produktivitas usaha tani dan kesejahteraan masyarakat dan wilayah tersebut.

2.2.1 Kegiatan Kelompok Tani

Kegiatan kelompok tani mencakup serangkaian aktivitas yang dilakukan oleh anggota kelompok dengan tujuan meningkatkan produksi pertanian dan kesejahteraan mereka. Kegiatan-kegiatan ini melibatkan proses yang saling terkait dan melibatkan partisipasi aktif dari semua anggota kelompok. Beberapa kegiatan yang umum dilakukan oleh kelompok tani meliputi.

1. Pengolahan

Setiap Anggota berpartisipasi dalam kegiatan pertanian mulai dari persiapan lahan, penanaman tanaman, pemeliharaan tanaman, pemanenan, pengolahan hasil pertanian, dan pemasaran produk pertanian.

2. Kolaborasi

Dalam pelaksanaan kegiatannya, kelompok tani dapat memanfaatkan keahlian dan pengalaman dari setiap anggota, sehingga meningkatkan efisiensi dan produktivitas pertanian di wilayah tersebut.

2.2.2 Fungsi Kelompok Tani

Kelompok tani memiliki fungsi yang luas dalam konteks pertanian. Fungsi-fungsi ini mencakup berbagai aspek yang penting untuk perkembangan pertanian.

1. Sarana Pembelajaran

Kelompok tani berperan sebagai wadah untuk pertukaran pengetahuan dan pengalaman antara anggota. Mereka dapat saling belajar dari praktik-praktik terbaik dalam bertani serta berbagi informasi tentang teknik baru yang efektif.

2. Pengawasan

Kelompok tani sebagai mekanisme koordinasi dalam pelaksanaan kegiatan pertanian, termasuk pelestarian hutan, penggunaan pupuk dan pestisida, serta pengaturan pemanenan.

3. Kerja Sama

Kelompok tani juga berfungsi sebagai sarana komunikasi dengan pihak luar, seperti lembaga pemerintah dan pasar, untuk mengakses sumber daya dan informasi yang mendukung kegiatan pertanian.

2.2.3 Tujuan Kelompok Tani

Tujuan adanya kelompok tani meluas ke berbagai aspek, baik bagi anggota kelompok maupun masyarakat di sekitarnya.

1. Konservasi Sumber Daya Alam

Kelompok tani tidak hanya terbatas pada pengelolaan pertanian konvensional, tetapi juga melibatkan pemanfaatan sumber daya alam, termasuk hutan, dengan pendekatan yang berkelanjutan. Dalam hal ini, kelompok tani memiliki peran penting dalam mengelola hutan dan mengalihgunakan lahan hutan menjadi lahan agroforestry.

2. Tantangan Pertanian

Tujuan kelompok tani juga mencakup mengatasi tantangan yang dihadapi bersama-sama, seperti perubahan iklim, fluktuasi harga komoditas pertanian, atau masalah-masalah teknis. Dengan berkolaborasi, kelompok tani dapat mencari solusi yang lebih baik.

3. Ekonomi

Bagi masyarakat lokal, kelompok tani berperan dalam pengembangan ekonomi di daerah tersebut dengan meningkatkan produksi pertanian dan memberikan kontribusi pada peningkatan ekonomi.

2.3 Pemrograman Web

Pemrograman web merupakan keterampilan kunci dalam pengembangan aplikasi dan situs web interaktif. Ini melibatkan penggunaan bahasa pemrograman tertentu untuk merancang dan mengembangkan berbagai elemen dalam suatu situs web, seperti antarmuka pengguna, basis data, dan fungsionalitas yang diperlukan. Pemrograman web memungkinkan pembuatan situs web yang dinamis dan interaktif, di mana pengguna dapat berinteraksi dengan konten dan data yang ditampilkan.

2.3.1 WEBSITE

Website adalah kumpulan halaman web yang saling terhubung dan dapat diakses melalui internet yang dapat digunakan untuk berbagai tujuan, seperti memberikan informasi, melakukan bisnis, berbagi ide, dan mengekspresikan kreativitas (Didik, 2017). Website dapat diakses melalui browser web dan dapat dilihat pada berbagai perangkat, seperti komputer desktop, laptop, tablet, dan smartphone. Website dapat dikembangkan menggunakan berbagai bahasa pemrograman dan teknologi, seperti HTML, CSS, JavaScript, PHP, dan MySQL. Desain dan konten website dapat bervariasi tergantung pada tujuan dan audiens target. Kualitas sebuah website dapat dinilai berdasarkan berbagai faktor, seperti kegunaan, kualitas informasi, kualitas interaksi, dan desain. Terdapat berbagai alat dan metode yang dapat digunakan untuk merancang, mengembangkan, dan mengevaluasi website, seperti UML, Webqual, dan Analytical Hierarchy Process (AHP).

2.3.2 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman berbasis server-side yang populer digunakan dalam pengembangan web karena dapat melakukan parsing script php menjadi script web sehingga dari sisi client (Ardhana, 2012). PHP banyak dipakai untuk memprogram situs web dinamis, walaupun tidak

tertutup kemungkinan digunakan untuk pemakaian lain. PHP merupakan bahasa pemrograman open source yang dapat dijalankan pada server-side.

Bahasa pemrograman PHP dapat digunakan untuk membangun aplikasi web berbasis GUI pada sisi client. Selain itu, dalam pengembangan aplikasi web, PHP dapat digunakan bersama dengan database MySQL, bahasa ini sangat kuat dalam menghubungkan basis data dengan situs web, memungkinkan aplikasi untuk menyimpan, mengambil, dan mengelola data dengan efisien. Dengan sintaks yang intuitif dan dukungan komprehensif, PHP memudahkan pengembang dalam membuat aplikasi web yang dinamis dan responsif.

2.3.3 MYSQL

MYSQL adalah salah satu sistem manajemen basis data paling populer yang digunakan dalam pengembangan web. Basis data adalah komponen kunci dalam penyimpanan data secara terstruktur dan aman. Penggunaan MYSQL dalam kombinasi dengan bahasa pemrograman seperti PHP memungkinkan situs web untuk menyimpan informasi penting seperti profil pengguna, informasi produk, dan data transaksi. Kemampuan untuk mengelola data ini adalah elemen vital dalam pengembangan aplikasi web modern.

MySQL menggunakan bahasa Structured Query Language (SQL) untuk melakukan operasi pada data, seperti mengambil data, memperbarui data, menghapus data, dan lain-lain. Fungsi utama MySQL adalah:

1. Menyimpan dan mengelola data dalam tabel.
2. Mengambil data dari tabel menggunakan query SQL.
3. Memperbarui, memasukkan, dan menghapus data dalam tabel.
4. Membuat dan mengubah struktur tabel.
5. Mengelola akses pengguna dan izin ke basis data.
6. Melakukan backup dan restore data dalam basis data.

2.3.4 WordPress

WordPress merupakan platform manajemen konten populer untuk pembuatan situs web. WordPress menyediakan solusi yang mudah digunakan untuk mengelola konten seperti tema dan plugin yang tersedia memungkinkan dalam menyesuaikan tampilan dan fungsionalitas situs sesuai kebutuhan. WordPress juga menawarkan antarmuka intuitif untuk mengedit dan mempublikasikan konten dengan mudah, menjadikannya pilihan yang tepat bagi pembuatan situs web menarik dan mudah diatur.

2.3.5 Library dan Plugin

Dalam ekosistem pengembangan web, library dan plugin memiliki peranan penting dalam memperluas fungsionalitas serta efisiensi proses pembuatan situs. Keduanya adalah sumber daya tambahan yang memudahkan pengembang dalam menghadirkan fitur-fitur yang kompleks tanpa harus membuatnya dari awal.

1. Library

Library adalah kumpulan kode yang sudah siap digunakan oleh pengembang dalam pembuatan situs web. Penggunaan library dapat menghemat waktu dan pengembangan, karena dapat memanfaatkan kode yang sudah ada untuk menghadirkan elemen interaktif dan menarik pada situs.

2. Plugin

Plugin adalah komponen tambahan yang memperluas fungsionalitas platform atau sistem manajemen konten. Dalam WordPress, plugin merupakan modul yang dapat ditambahkan ke situs web untuk memberikan fitur tambahan yang tidak ada dalam konfigurasi dasar.

2.4 Alat Bantu Perancangan Sistem

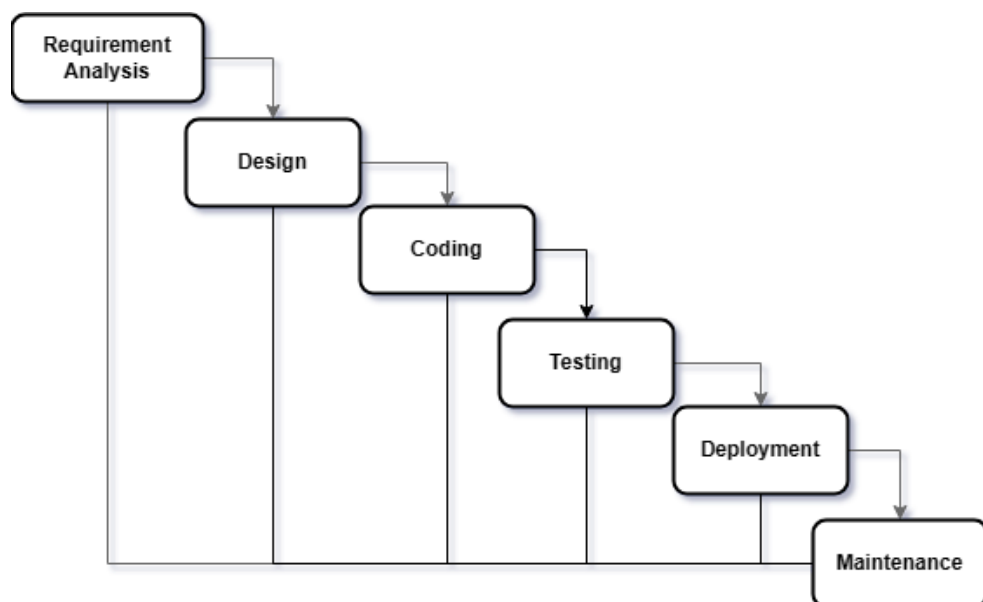
Selain pemahaman konsep sistem informasi, pengembangan website untuk meningkatkan aksesibilitas dan efisiensi sistem informasi pada kelompok tani di Desa Sidapdap Simanosor juga memerlukan berbagai alat bantu yang mendukung proses perancangan dan pengembangan. Alat-alat bantu ini membantu dalam

menggambarkan struktur sistem, mengorganisir informasi, dan memastikan keselarasan antara fungsi-fungsi yang diperlukan.

Dalam perancangan sistem informasi dan program, langkah awal adalah menyusun model pemecahan masalah secara logis, yang sesuai dengan tantangan yang dihadapi. Pada tahap ini, terdapat beberapa alat bantu yang dimanfaatkan untuk mendorong pemecahan masalah dalam pembuatan sistem.

2.4.1 Model *Waterfall*

Model *waterfall* adalah model proses pengembangan perangkat lunak yang mengikuti tahapan secara berurutan, dimulai dari analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan dimana model ini sering disebut sebagai model "linier-bersekuensial" karena setiap tahapan harus selesai sebelum tahapan berikutnya dimulai (Gull & Butt, 2012). Model ini merupakan salah satu pendekatan dalam pengembangan perangkat lunak yang mengikuti serangkaian tahap yang linear dan berurutan, mirip dengan aliran air (*waterfall*) yang mengalir turun dari atas ke bawah. Penjelasan tahapan dapat dilihat seperti pada Gambar 1.2.



Gambar 2. 2 : Model Waterfall

1. Analisis Kebutuhan (*Requirement Analysis*)

Tahap pertama adalah mengumpulkan dan menganalisis kebutuhan dari sistem yang akan dikembangkan. Ini melibatkan interaksi dengan pemangku kepentingan untuk memahami tujuan, persyaratan, dan kebutuhan bisnis yang harus diakomodasi dalam sistem.

2. Perancangan (*Design*)

Setelah persyaratan dikumpulkan, tim mengembangkan desain sistem yang terperinci. Ini mencakup perancangan arsitektur, desain antarmuka pengguna, desain database, dan rancangan komponen perangkat lunak lainnya.

3. Pengkodean (*Coding*):

Tahap ini melibatkan penulisan kode perangkat lunak berdasarkan desain yang telah dibuat sebelumnya. Para pengembang bekerja untuk mengimplementasikan berbagai fitur dan fungsi yang diperlukan oleh sistem.

4. Pengujian (*Testing*):

Setelah kode ditulis, tahap pengujian dimulai. Berbagai jenis pengujian dilakukan untuk memastikan bahwa perangkat lunak berjalan dengan benar sesuai dengan spesifikasi. Pengujian mungkin melibatkan pengujian unit, integrasi, fungsional, kinerja, dan lain-lain.

5. Implementasi (*Deployment*):

Setelah perangkat lunak dinyatakan lulus pengujian, maka perangkat lunak tersebut siap untuk diterapkan di lingkungan produksi. Ini melibatkan instalasi perangkat lunak di sistem target dan memastikan bahwa sistem bekerja dengan benar di lingkungan produksi.

6. Pemeliharaan (*Maintenance*):

Setelah perangkat lunak diimplementasikan, perawatan dan pemeliharaan dilakukan untuk memastikan perangkat lunak tetap berjalan dengan baik. Ini melibatkan perbaikan kesalahan, peningkatan fitur, dan perubahan sesuai kebutuhan bisnis yang berkembang.

2.4.2 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa pemodelan terpadu yang digunakan dalam perancangan sistem informasi dan perangkat lunak dimana memungkinkan pengembang perangkat lunak untuk memodelkan perangkat lunak secara visual dengan menggunakan beberapa jenis diagram (Yasin, 2021). UML juga dapat membantu dalam menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak, karena menyediakan metode standar yang dapat dimengerti oleh berbagai pihak terkait dalam proses pengembangan.

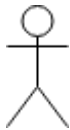
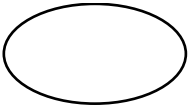

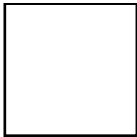
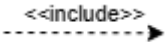
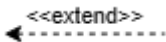

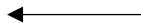
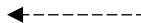
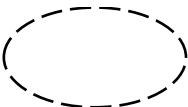
UML digunakan untuk menggambarkan berbagai aspek sistem perangkat lunak, termasuk struktur, interaksi, perilaku, dan relasi antar komponen dalam sistem. Dalam UML, ada beberapa jenis diagram yang mewakili aspek-aspek yang berbeda dari sistem perangkat lunak. Berikut adalah beberapa jenis diagram UML yang umum digunakan:


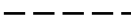
1. *Use Case Diagram*
2. *Class Diagram*
3. *Sequence Diagram*
4. *Activity Diagram*
5. *State Machine Diagram*
6. *Component Diagram*
7. *Deployment Diagram*

2.4.3 *Use Case Diagram*

Use Case Diagram digunakan untuk menggambarkan interaksi antara pengguna atau aktor dengan sistem dalam bentuk skenario kasus penggunaan, dimana diagram ini membantu dalam mengidentifikasi kebutuhan fungsional sistem dari sudut pandang pengguna (Adhi, 2018). Aktor dalam diagram ini bisa berupa pengguna, sistem eksternal, atau entitas lain yang berinteraksi dengan sistem. Kasus penggunaan *use case* menggambarkan perilaku fungsional sistem dari perspektif pengguna.

Dalam *Use Case Diagram*, terdapat beberapa simbol yang digunakan untuk menggambarkan elemen-elemen yang berperan dalam interaksi antara pengguna dengan sistem. Berikut ini adalah beberapa simbol yang umum digunakan dalam *Use Case Diagram* beserta keterangannya.

Simbol	Nama	Keterangan
	<i>Aktor</i>	Entitas di luar sistem yang berinteraksi dengan sistem. Aktor bisa menjadi pengguna manusia, perangkat keras, atau sistem eksternal
	<i>User Case</i>	Aksi atau skenario yang dapat dilakukan oleh aktor pada sistem
	<i>Association</i>	Hubungan antara aktor dan kasus penggunaan yang menggambarkan interaksi antara mereka
	<i>System</i>	Sistem yang sedang dikembangkan
	<i>Include</i>	Menunjukkan bahwa suatu use case mencakup fungsionalitas dari use case lain
	<i>Extend</i>	Menunjukkan bahwa suatu use case memiliki skenario alternatif yang dapat diperpanjang tergantung kondisi tertentu
	<i>Dependency</i>	Menunjukkan hubungan antara elemen-elemen yang satu mempengaruhi yang lain, misalnya penggunaan suatu class oleh use case
	<i>Generalization</i>	Menunjukkan hubungan hierarki di antara aktor atau use case, di mana yang satu adalah bentuk khusus dari yang lain
	<i>Realization</i>	Menunjukkan implementasi atau realisasi use case oleh suatu class atau komponen
	<i>Collaboration</i>	Menunjukkan hubungan antara beberapa use case yang bekerja bersama untuk mencapai tujuan tertentu


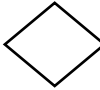
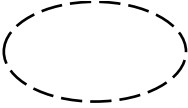
	<i>Note</i>	Berisi keterangan atau penjelasan tambahan untuk elemen-elemen dalam diagram
	<i>Anchor</i>	Mengaitkan simbol ke luar dari diagram, sering digunakan untuk menghubungkan diagram use case yang berbeda

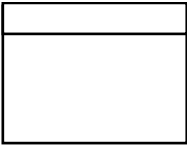
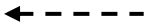

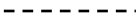
Tabel 2. 1 : Simbol Pada Use Case Diagram

2.4.4 Class Diagram

Class Diagram adalah salah satu jenis diagram dalam UML (*Unified Modeling Language*) yang digunakan untuk memodelkan struktur statis dari suatu sistem berorientasi objek (Rusdiana, 2016). Diagram ini membantu dalam memvisualisasikan entitas dalam sistem dan hubungan di antara mereka. Kelas-kelas dalam diagram ini merepresentasikan objek-objek yang ada dalam sistem, dan hubungan antar kelas menggambarkan asosiasi, komposisi, pewarisan, dan lainnya.

Berikut ini *adalah* beberapa simbol yang umum digunakan dalam *Use Case Diagram* beserta keterangannya.

Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

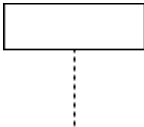
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya



Tabel 2. 2 : Simbol Pada *Class Diagram*

2.4.5 *Sequence Diagram*

Sequence Diagram adalah diagram yang mencoba menggambarkan secara detail interaksi antar objek dalam sebuah sistem, pesan atau perintah yang dikirimkan, dan waktu *implementasinya* biasanya diagram ini digunakan untuk menggambarkan alur penjelasan objek dalam bentuk diagram grafis (Hong & Park, 2014). Diagram ini biasanya digunakan dalam pengembangan perangkat lunak untuk memodelkan interaksi antar objek dalam sistem

Berikut ini adalah beberapa simbol yang umum digunakan dalam *Sequence Diagram* beserta keterangannya

Simbol	Nama	Keterangan
	<i>LifeLine</i>	Entitas di luar sistem yang berinteraksi dengan sistem. Aktor bisa menjadi pengguna manusia, perangkat keras, atau sistem eksternal



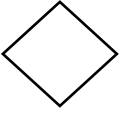
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi

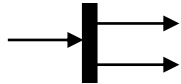

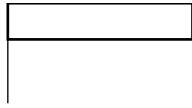
Tabel 2. 3 : Simbol Pada *Sequence Diagram*

2.4.6 *Activity Diagram*

Activity Diagram adalah salah satu jenis diagram yang digunakan untuk merepresentasikan tindakan atau aktivitas yang terjadi dalam proses atau sistem sistem *berorientasi* objek dengan menggunakan bahasa pemodelan *Unified Modeling Language* (UML). Diagram ini membantu dalam memvisualisasikan langkah-langkah yang harus diambil dalam satu proses. Notasi dalam diagram ini mencakup aktivitas, tindakan, keputusan, dan fork/join yang membantu dalam menggambarkan aliran logika dan paralel dalam proses

. Berikut ini *adalah* beberapa simbol yang umum digunakan dalam *Sequence Diagram* beserta keterangannya.

Simbol	Nama	Keterangan
	<i>Initial</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	<i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.




	<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	<i>Final</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.


Tabel 2. 4 : Simbol Pada *Activity Diagram*

2.4.7 *State Machine Diagram*

State Machine Diagram adalah diagram yang digunakan untuk merepresentasikan perilaku sistem yang terdiri dari kumpulan state, transisi antar state, dan aksi yang terkait *dengan* transisi yang menunjukkan perubahan keadaan yang terjadi pada objek dalam sistem sepanjang waktu. *State Machine Diagram* membantu memvisualisasikan bagaimana objek berperilaku dalam berbagai keadaan yang berbeda, membantu memahami alur perubahan keadaan dan respons dari objek terhadap input eksternal.

Berikut beberapa penjelasan tentang simbol yang umum digunakan dalam diagram *State Machine*.

Simbol	Nama	Keterangan
	<i>State</i>	digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisi saat itu.
	<i>Start</i>	Digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram flowchart.
	<i>End</i>	Digunakan untuk menjelaskan/ menggambarkan akhir (<i>end</i>) dari kejadian dalam suatu diagram state chart.
[Guard]	<i>Guard</i>	Merupakan syarat transisi yang bersangkutan.

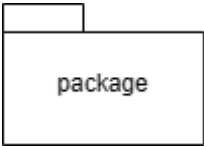

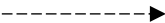


	<i>Point</i>	Digunakan untuk menggambarkan/ menjelaskan apakah akan masuk (<i>entry point</i>) ke dalam status atau keluar (<i>exit point</i>).
<i>event</i>	<i>Event</i>	Digunakan untuk menjelaskan kondisi yang menyebabkan sesuatu pada status.

Tabel 2. 5 : Simbol Pada *State Machine Diagram*

2.4.8 *Component Diagram*

Component Diagram adalah diagram yang digunakan untuk merepresentasikan komponen-komponen dari sebuah sistem dan hubungan antar komponen tersebut.. *Component Diagram* membantu memahami bagaimana komponen-komponen saling berinteraksi dalam sistem, membantu mengidentifikasi dependensi dan pemisahan tanggung jawab.

Berikut ini penjelasan tentang beberapa simbol yang umum digunakan dalam *Component Diagram*.

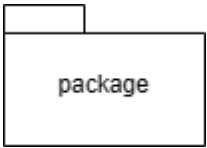
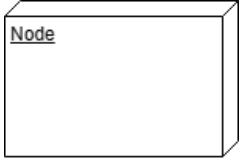
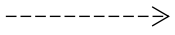
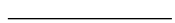
Simbol	Nama	Keterangan
	<i>Package</i>	Sebuah bungkus dari satu atau lebih komponen.
	<i>Component</i>	Merupakan komponen dari sistem
	<i>Dependency</i>	Kebergantungan antar komponen mengarah pada komponen yang dipakai.
	<i>Interface</i>	Sebagai antar muka komponen agar tidak mengakses langsung komponen
	<i>Link</i>	Menggambarkan relasi antar komponen

Tabel 2. 6 : Simbol Pada *Component Diagram*

2.4.9 Deployment Diagram

Deployment Diagram adalah diagram yang digunakan untuk merepresentasikan distribusi komponen-komponen dari sebuah sistem ke dalam lingkungan yang lebih besar, diagram ini menggambarkan bagaimana komponen sistem diterapkan pada perangkat keras dan perangkat lunak fisik. *Deployment Diagram* membantu merencanakan distribusi dan penyebaran komponen dalam lingkungan produksi, membantu memvisualisasikan bagaimana komponen berjalan di lingkungan nyata.

Berikut ini adalah penjelasan tentang beberapa simbol yang umum digunakan dalam *Deployment Diagram*.

Simbol	Nama	Keterangan
	<i>Package</i>	Sebuah bungkus dari satu atau lebih komponen.
	Node	biasa mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen
	<i>Dependency</i>	Ketergantungan atau dependency atau kebergantungan antar node, arah panah mengarah pada node yang dipakai
	<i>Link</i>	Menggambarkan relasi antar node

Tabel 2. 7 : Simbol Pada *Deployment Diagram*