

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem Informasi Manajemen

a. Sistem

Sistem terdiri dari komponen fungsional yang memiliki tugas atau fungsi tertentu yang memiliki keterkaitan dengannya dan kemudian bekerja sama untuk mencapai tujuan dari suatu proses atau pekerjaan tertentu (Gede Endra Bratha, 2022).

Menurut (Ariansyah & Wijaya, 2021) Sistem adalah suatu benda atau kesatuan yang merupakan kumpulan berbagai bagian dan komponen serta proses dan metode sarana untuk mencapai suatu tujuan yang saling berhubungan secara terorganisir sesuai dengan fungsinya untuk mencapai tujuan yang sama.

Berdasarkan uraian di atas, sistem adalah suatu kumpulan komponen, proses atau metode yang secara teratur berhubungan satu sama lain dan menjalin hubungan. Untuk mencapai tujuan tertentu, tidak mungkin memisahkan bagian-bagian ini satu sama lain karena masing-masing memiliki tugas dan fungsi yang unik.

b. Informasi

Informasi merupakan kumpulan data yang terolah menjadi sesuatu yang memiliki arti dan kegunaan yang lebih luas, yaitu kumpulan data yang telah diolah menjadi suatu format yang berguna

bagi penggunaanya (Ariansyah & Wijaya, 2021). Menurut (Gede Endra Bratha, 2022) Informasi adalah hasil dari pengolahan sebuah model, farmasi, organisasi atau suatu perubahan data yang memiliki sebuah nilai tertentu, dan dapat menambah pengetahuan bagi yang menerimanya.

Berdasarkan definisi diatas, dapat disimpulkan bahwa informasi merupakan suatu kumpulan data yang memiliki nilai tertentu dan dapat menambah pengetahuan bagi penerimanya.

c. Sistem Informasi Manajemen

Sistem informasi manajemen ialah sistem informasi yang memiliki hasil dengan berbagai proses untuk mencapai tujuan pengelolaan tertentu. Sistem informasi manajemen merupakan sarana penyediaan informasi tepat waktu kepada manajemen terkait dengan lingkungan eksternal organisasi dan aktivitas bisnis internal organisasi. Sistem ini dirancang untuk memfasilitasi proses manajemen, meningkatkan proses perencanaan dan pengawasan, serta mendukung proses pengambilan keputusan (Gede Endra Bratha, 2022).

Peneliti menarik kesimpulan bahwa sistem informasi manajemen adalah sebuah sistem informasi yang terdapat proses *input*, *output* yang memiliki tujuan dan manfaat untuk mendukung kegiatan manajemen yang dilakukan oleh suatu organisasi.

2. Praktek Kerja Lapangan

a. Praktek Kerja Lapangan

Program lapangan adalah upaya universitas untuk menciptakan profil kemampuan lulusan dan mahasiswa program studi yang siap berkarir dan sukses. Kompetensi tersebut antara lain mencakup kompetensi pengembangan diri, pengetahuan dan keterampilan, keterampilan kreasi dan inovasi, serta kompetensi hidup bermasyarakat berbasis pembelajaran aktif mahasiswa (Anam et al., 2020).

Praktek Kerja Lapangan (PKL) adalah kegiatan dalam pendidikan yang melakukan pelatihan dilaksanakan secara khusus dan sistematis di bawah pengawasan yang tepat pada suatu bidang tertentu, dengan tujuan untuk memenuhi kebutuhan tenaga ahli di bidang tersebut. Mahasiswa yang mendaftar pada kegiatan PKL ini diharapkan dapat memanfaatkan ilmu yang diperolehnya di dunia industri dan mendapatkan pengalaman baru serta keterampilan tingkat lanjut di bidang pekerjaannya. (Abdussalaam & Ramdani, 2023).

Berdasarkan definisi diatas, PKL ialah upaya dalam bidang pendidikan berbentuk pelatihan yang bertujuan untuk melatih kemampuan mahasiswa dan menerapkan ilmu yang telah di gelutinya, dengan demikian mahasiswa mendapatkan pengalaman baru serta dapat menambah pengetahuan lebih dalam pada dunia industri.

3. *Website*

a. *Website*

Internet telah sangat membantu pertumbuhan dan perkembangan sektor perdagangan dan jasa Indonesia sejak pertama kali diluncurkan pada tahun 1969 dan berkembang pesat pada tahun 1990-an. Kendala ruang (geografi) dan waktu dengan mudah dielakkan menggunakan internet (Siaha Widodo, 2019). Menurut (Dewi et al., 2021), *website* adalah kumpulan halaman dalam *domain* yang menampung berbagai jenis konten yang dapat diakses oleh pengguna internet melalui mesin pencari. Informasi yang dapat disertakan di situs *web* biasanya terdiri dari gambar, animasi, audio, video dan konten tekstual untuk berbagai tujuan (Andi & Yakub, 2019).

Situs *web* adalah aplikasi yang menggunakan protokol *Hyper Text Transfer Protocol* (HTTP) yang berisi materi multimedia dalam bentuk teks, grafik, suara, animasi, atau video, menurut (Ferdiansyah, 2018). Dokumen-dokumen ini harus diakses melalui *browser*. Menurut (Asmara, 2019), situs *web* mengklaim bahwa informasi tersedia di setiap halaman di dalam domain tersebut. Sebuah situs *web* terdiri dari halaman-halaman yang dihubungkan bersama untuk membentuk jaringan bangunan yang saling berhubungan, berisi informasi tekstual, gambar atau video, suara, dan apa pun (Bekti, 2005).

Situs *web* juga dapat disebut sebagai situs *web* atau portal, menurut (Haan, 2020). Ini adalah sekumpulan halaman *web* yang

terhubung satu sama lain. Beranda situs *web* adalah halaman utamanya; atau dikenal sebagai, setiap halaman disebut sebagai halaman *web* satu per satu. Situs *web* ini dapat diakses oleh semua orang yang menggunakan internet di dunia.

Berdasarkan definisi diatas, *website* merupakan kumpulan halaman berisi konten yang berisi berbagai multimedia seperti gambar, teks, grafik, suara, animasi, atau video dan dapat diakses oleh pengguna melalui internet dengan menggunakan mesin pencarian,

b. HTML (*HyperText Markup Language*)

Menurut (Pangestu & Putra, 2022), HTML (*HyperText Markup Language*) yaitu file teks sederhana yang dapat dibuat dengan editor teks apa pun. Dokumen-dokumen ini sering juga disebut sebagai halaman *web*. Dokumen HTML (*HyperText Markup Language*) adalah dokumen yang dilihat di browser *web*. HTML (*HyperText Markup Language*) adalah sistem untuk menyematkan dokumen dengan tabel yang menentukan bagaimana teks dokumen itu seharusnya terlihat dan bagaimana dokumen itu dihubungkan bersama (Supriady & Safitri, 2022). HTML ialah bahasa *markup* yang berguna untuk mengembangkan aplikasi dan situs *web* (Hussein, 2022). Maka dapat disimpulkan bahwa HTML (*HyperText Markup Language*) adalah dokumen atau, halaman *web* yang dilihat di *browser web* dan dapat dibuat dengan editor teks apa pun.

Berdasarkan definisi diatas, HTML (*HyperText Markup Language*) adalah bahasa komputer yang digunakan untuk membuat struktur dasar dan tampilan dari sebuah halaman *website* dan dapat dibuat menggunakan teks editor seperti atom, notepad++, sublime text, dan teks editor lainnya.

c. **CSS (*Cascading Style Sheet*)**

CSS (*Cascading Style Sheet*), termasuk bahasa pemrograman yang digunakan untuk mengatur tampilan *web* agar tertata dengan baik serta menarik. CSS dapat mengontrol font, warna tabel atau tata letak menu bahwa jendela situs saling eksklusif. Informasi terkait dapat berubah secara otomatis. CSS digunakan untuk mengatur gaya elemen halaman, mulai dari mengatur teks hingga mengatur tata letak (batas layar). CSS memiliki fungsi untuk mendapatkan *style* yang konsisten untuk elemen tertentu (Permana, 2013).

Menurut (Permana et al., 2022), CSS (*Cascading Style Sheets*) adalah aturan yang menangani berbagai komponen situs *web* agar lebih terstruktur dan konsisten. CSS dapat digunakan untuk memisahkan konten dari tampilan visualnya di situs *web*. CSS dibuat dan dikembangkan pada tahun 1996 oleh organisasi W3C (*World Wide Web Consortium*). Dulu HTML tidak dilengkapi *tag* untuk memformat halaman, malah menulis markup di *website* yang dibuat sudah cukup. HTML dan CSS merupakan pasangan yang tidak dapat dipisahkan, karena HTML adalah dasar dari sebuah *website* dan CSS adalah untuk

pemformatan (semua aspek tampilan sebuah *website*) dan kedua bahasa pemrograman ini harus berjalan beriringan.

Menurut (Ardiyansa et al., 2022) *Cascading Style Sheets*, juga dikenal sebagai CSS, adalah kumpulan dokumen *web* yang digunakan untuk mendesain tampilan situs *web*. CSS bukanlah bahasa pemrograman, tetapi aturan untuk mengatur komponen halaman *web* agar terstruktur, koheren, dan situs *web* yang mudah disesuaikan.

Berdasarkan adanya definisi diatas, CSS (*Cascading Style Sheets*) dapat digunakan untuk menentukan tampilan atau format pada sebuah *website* supaya terstruktur, selain itu kita dapat mengubah warna atau bentuk font pada sebuah *table* sesuai dengan format yang ada pada CSS.

d. *Javascript*

JavaScript adalah bahasa pemrograman sisi klien, menurut (Sari et al., 2022), dan digunakan untuk membuat situs *web*. Bahasa untuk pemrograman sisi klien *JavaScript* adalah bentuk bahasa pemrograman yang diproses oleh klien. Untuk mengintegrasikan manipulasi script HTML dan CSS pada sisi *client/browser*, salah satu bahasa *scripting website* yang paling banyak digunakan adalah *JavaScript* (Firdaus, 2019). Penjelasan di atas mengarah pada kesimpulan bahwa *JavaScript* adalah bahasa pemrograman yang dijalankan oleh klien seperti Google Chrome dan yang tujuannya meliputi pengembangan situs *web* dan aplikasi seluler, halaman *web*

interaktif, *server web* dan aplikasi server.

Berdasarkan definisi diatas, *JavaScript* adalah Bahasa pemrograman yang biasa digunakan untuk pengembangan *website* supaya lebih interaktif dan dinamis yang dapat berintegrasi dengan CSS dan HTML.

e. MySQL

Database open source yang unik adalah MySQL. Basis data sangat penting untuk menyimpan banyak informasi dalam bentuk informasi sambil membangun aplikasi atau situs *web* yang rumit dan dinamis. Server basis data juga diperlukan oleh situs *web* untuk menyimpan banyak data. Nama pengguna, kata sandi, dan informasi pengguna lainnya adalah beberapa contohnya. Selain itu, MySQL dapat mengelola semua jenis *database*, membuatnya dapat dikelola. Saat ini banyak aplikasi berbasis *web* yang dibuat dengan menggunakan MySQL, sebuah DBMS (*Database Management System*) yang menggunakan instruksi SQL (*Structured Query Language*). MySQL tersedia dalam dua lisensi berbeda, yang pertama adalah perangkat lunak bebas dan dapat digunakan oleh siapa saja, dan yang kedua adalah *shareware*, di mana penggunaan program berpemilik ini dibatasi. (Hamidi et al., 2022) mengemukakan hal serupa, MySQL menyatakan bahwa semua pengguna memiliki izin untuk melakukan hanya aktivitas yang diizinkan untuk mereka lakukan..

MySQL adalah perangkat lunak *server* data yang menggunakan SQL, atau bahasa *query* terstruktur, menurut (Novendri, 2019). Pengolahan basis data dilakukan dengan menggunakan bahasa terstruktur SQL. Suatu *Database Management System* (DBMS) yang bersifat *open source* disebut MySQL (Prahasti et al., 2022). Menurut definisi yang di atas, MySQL adalah DBMS (*Database Management System*) *open source* yang digunakan untuk pemrograman.

MySQL merupakan sebuah perangkat lunak sistem manajemen database dan bersifat *open source*, pada MySQL menggunakan perintah dasar atau bahasa pemrograman yang terstruktur yang biasanya disebut dengan SQL (*Structured Query Language*).

f. PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) memiliki sifat dinamis yang berguna untuk mengembangkan *website*, menurut (Novendri, 2019) PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman sisi *server* yang tertanam dalam struktur HTML dan memungkinkan konversi basis kode program menjadi kode mesin yang dapat dipahami komputer (Prahasti et al., 2022).

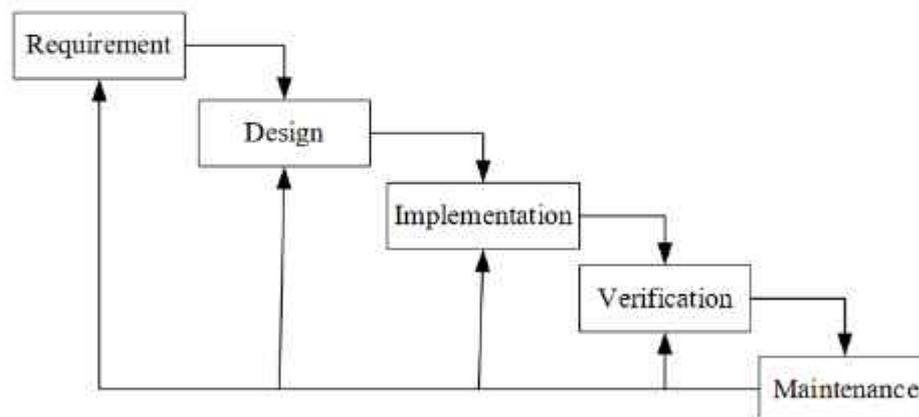
Berdasarkan definisi diatas, PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang bersifat *open source* yang digunakan untuk membangun sebuah *website*, yang memiliki tugas mengubah kode sumber program menjadi kode mesin dan kemudian dimasukkan kedalam kerangka HTML yang dikenali oleh komputer.

4. Perancangan Sistem

a. Waterfall

Metode *Waterfall* merupakan rancangan yang terdiri dari alur rancangan kebutuhan pengguna, perangkat yang akan dikembangkan serta melakukan uji coba dan validasi hingga implementasi sistem (Saputri & Eriana, 2021). Menurut (Hidayanti et al., 2022) pendekatan pengembangan *software* menggunakan metode *waterfall* dengan sistematis dimulai pada tingkat kemajuan sistem informasi sampai analisis, desain, kode, pengujian, dan pemeliharaan.

Tahapan metode waterfall adalah sebagai berikut :



Gambar 1. 1 Metode *Waterfall*

(Pressman, 2012)

Menurut (Pressman, 2012) dalam bukunya yang berjudul *Rekayasa Perangkat Lunak*, penjelasan beberapa tahapan metode waterfall adalah sebagai berikut :

1. *Requirement analysis* (analisa kebutuhan), adalah pendefinisian kebutuhan perangkat lunak berdasarkan permasalahan dan persyaratan yang diperlukan untuk desain aplikasi berdasarkan

kinerja dan fungsionalitas antarmuka.

2. *Design* (desain), melakukan kesiapan untuk membuat desain antarmuka *software* yang akan digunakan untuk sistem yang dibangun.
3. *Coding* (implementasi), melakukan implementasi koding dan desain dengan bahasa pemrograman PHP.
4. *Testing* (pengujian), setelah sistem dibuat dan selesai melakukan koding, aplikasi diuji untuk melihat apakah desain sistem dapat siap digunakan pengguna.
5. *Maintenance* (perawatan), setelah aplikasi ini digunakan dan diimplementasikan perlu adanya maintenance secara berkala dalam mengelola aplikasi *website* sehingga aplikasi akan tetap berjalan sesuai dengan fungsinya, selain itu juga dilakukan evaluasi atau perbaikan jika terdapat bagian yang mengalami kendala atau error pada sistem.

Berdasarkan definisi diatas, metode *Waterfall* merupakan pendekatan pengembangan perangkat lunak yang mengungkapkan serangkaian proses mulai dari menentukan kebutuhan pengguna dan perangkat yang akan dikembangkan, melakukan pengujian verifikasi, dan mengimplementasikannya ke dalam sistem.

b. UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa yang umum berguna untuk memvisualisasikan, merancang, dan mendokumentasikan

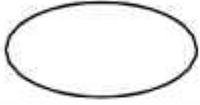
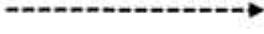
software. Standar untuk membuat model sistem disediakan oleh UML (Sugiarti, 2013:34). Syarif & Nugraha (Syarif & Nugraha, 2020) mengungkapkan hal yang sama.

Berdasarkan definisi diatas, UML adalah bahasa visual yang menggunakan diagram dan prosa terkait untuk mewakili dan mengkomunikasikan informasi terkait sistem. Diagram adalah blok bangunan utama UML, beberapa spesifik (seperti beberapa jenis diagram waktu), sementara yang lain lebih umum. Bahasa pemodelan yang digunakan untuk perangkat lunak atau sistem berorientasi objek disebut UML (*Unified Modeling Language*).

1) *Usecase Diagram*

Usecase diagram adalah model yang mendefinisikan perilaku dari sistem yang telah dibangun. Menurut (Tohari, 2014), *usecase* adalah rangkaian atau deskripsi dari sebuah grup yang terhubung satu sama lain dan membentuk sistem yang teratur di bawah kendali seorang aktor. *Usecase diagram* yang dapat dipahami secara lebih ringkas digunakan untuk menentukan fungsionalitas mana yang akan ada pada sistem yang akan dibangun dan siapa yang memiliki kewenangan untuk menggunakannya. Menurut (Hasanah, 2020) seperti terlihat pada tabel 2.1, terdapat beberapa simbol dalam *usecase* yang digunakan untuk mendeskripsikan diagram. Penting untuk diketahui bahwa simbol-simbol ini tidak mewakili arsitektur sistem, persyaratan non-fungsional, atau sasaran kinerja.

Tabel 2. 1 Simbol pada *Usecase*

Simbol	Keterangan
	Aktor: peran <i>user</i> , system atau alat berkomunikasi menggunakan <i>usecase</i>
	<i>Usecase</i> : abstraksi dan interaksi antara sistem dan actor
	<i>Association</i> : akbstraksi dari penghubung antara aktor <i>usecase</i>
	<i>Generalisasi</i> : spesialisasi aktor guna berpartisipasi dengan <i>usecase</i>
<<Include>>	Menunjukkan <i>usecase</i> secara keseluruhan yang fungsionalitas yang berasal dari <i>usecase</i> lainnya
<<Extend>>	Menunjukkan <i>usecase</i> tambahan fungsional dari <i>usecase</i> lainnya apabila suatu kondisi telah terpenuhi

2) *Class Diagram*

Landasan dari pengembangan dan desain berorientasi objek adalah diagram kelas, yang merupakan spesifikasi yang ketika diinstansiasi, menghasilkan objek (Tohari, 2014). Sementara fungsi kelas yang berbeda diwakili oleh operasi atau metodenya, atributnya adalah variabel kelas. Diagram kelas menunjukkan komposisi dan

hubungan antara kelas, paket, dan objek, seperti penahanan, pewarisan, asosiasi, dan sebagainya.

Menurut (Siregar et al., 2018), diagram kelas adalah hubungan antar kelas dalam model desain sistem dan deskripsi setiap kelas. Itu juga menampilkan hukum dan kewajiban entitas yang mengontrol aktivitas sistem.

Class Diagram kelas digunakan untuk menggambarkan jenis objek sistem dan hubungannya dengan objek lain, dapat menampilkan simbol-simbol yang terdapat dalam diagram kelas pada table berikut ini (Suhani et al., 2023).

Tabel 2. 2 Simbol *Class Diagram*

Simbol	Keterangan
 Class	Kumpulan objek dengan atribut berbeda yang melakukan operasi yang sama.
 <i>Association</i>	Beberapa kelas yang memiliki hubungan dan pengertian yang melibatkan keberagaman.
 <i>directed association</i>	Hubungan antar kelas yang artinya kelas tersebut digunakan oleh kelas lain.
 <i>Aggregation</i>	Menunjukkan bahwa keseluruhan hubungan disebut sebagai relasi.
 <i>Composition</i>	Hubungan antara konfigurasi dan kelas tempat bergantungnya.

Simbol	Keterangan
 <i>Dependency</i>	Menampilkan operasi suatu kelas yang menggunakan kelas lain.

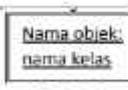
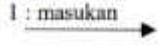
3) *Sequence Diagram*

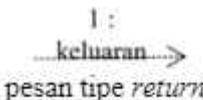
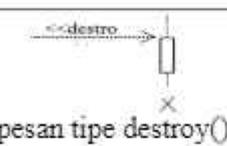
Sequence Diagram, yang menggambarkan pesan yang dikirim dan diterima oleh objek serta masa pakainya, digunakan untuk memvisualisasikan aktivitas atau perilaku objek *dalam use case*. Akibatnya, untuk melihat diagram urutan, kita harus mengetahui objek yang ada dalam *use case* serta metode kelas yang berlaku untuk objek tersebut.

Menurut (Aisyah, 2019), mewakili pesan yang ditampilkan dari waktu ke waktu untuk menunjukkan interaksi antar komponen sistem. termasuk pengguna, antarmuka, dan hal-hal lain di dalam dan sekitar sistem disebut dengan *sequence diagram*. Berdasarkan definisi yang diberikan di atas, dapat disimpulkan bahwa *sequence diagram* adalah gambaran yang menggambarkan bagaimana objek berperilaku dan berfungsi untuk menampilkan rangkaian komunikasi yang telah terjadi sebagai hasil interaksi antara mereka.

Berikut table *sequence diagram* menurut (Irfan et al., 2023).

Tabel 2. 3 Simbol *Sequence Diagram*

Simbol	Keterangan
 Aktor	Proses, orang, atau sistem berbeda yang terhubung ke sistem data yang dibuat berada di luar sistem informasi yang dibuat itu sendiri, jadi meskipun simbol aktor adalah gambar seseorang, aktor tersebut belum tentu orang.
 <i>Lifeline</i>	Mewakili siklus hidup objek.
 Objek	Mendeklarasikan object yang berinteraksi dengan pesan.
 waktu aktif	Menyatakan bahwa objek tersebut aktif dan interaktif, dan segala sesuatu yang berkaitan dengan waktu aktifnya merupakan langkah yang dibangun ke dalam objek.
 pesan tipe <i>create</i>	Mendeklarasikan suatu objek akan membuat objek lain, dan panah menunjuk ke objek yang dibuat. Arah panah menunjuk ke objek yang mempunyai operasi/metode. Karena ini memanggil operasi/metode, operasi/metode yang dipanggil harus ada dalam diagram kelas tergantung pada kelas objek yang berinteraksi dengan.
 pesan tipe <i>send</i>	Objek yang mengirimkan masukan/data/informasi ke objek lain, dengan tanda

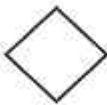
Simbol	Keterangan
	Menunjukkan bahwa setelah melakukan operasi atau metode tertentu, objek kembali ke objek yang ditentukan, dengan panah menunjuk ke objek yang menerima pengembalian.
	Menyatakan suatu benda untuk mengakhiri hidup benda lain. Panah menunjuk ke objek akhir.

4) Activity Diagram

Activity Diagram atau diagram aktivitas “menggambarkan alur kerja atau aktivitas dari suatu sistem atau proses bisnis”, menurut (Yuni Sugiarti, 2013). Penting untuk dicatat bahwa *activity diagram* hanya menunjukkan aktivitas yang dapat dilakukan oleh sistem, bukan tindakan yang mungkin diambil oleh aktor. Selain itu, proses paralel yang dapat berlangsung selama beberapa eksekusi dapat ditampilkan dalam *activity diagram*. Hampir semua status dalam *activity diagram* adalah tindakan, dan transisi dimulai saat status sebelumnya berakhir.

Tabel 2. 4 Simbol *Activity Diagram*

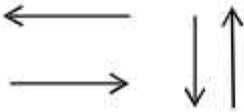
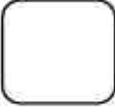
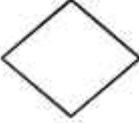
Simbol	Keterangan
	Aktor: merupakan peran orang, ketika berkomunikasi dengan <i>usecase</i>

Simbol	Keterangan
awal	
 aktivitas	<i>Usecase</i> : abstraksi dan interaksi antarsistem dan actor
 Percabangan	<i>Association</i> : akbstraksi dari penghubungantara aktor dengan dengan <i>usecase</i>
 Penggabungan	<i>Generalisasi</i> : menunjukkan spesialisasi aktor untuk bisa berpartisipasi dengan <i>usecase</i> .
 Akhir	Status akhir sistem merupakan kondisi akhir dari sebuah sistem

5) *Flowchart*

Flowchart adalah kumpulan notasi visual simbolik yang menggambarkan alur kerja, aliran data, dan urutan proses sistem. *Flowchart* juga dapat dilihat sebagai alat analisis untuk menggambarkan secara rasional, ringkas, dan jelas berbagai elemen dari suatu sistem informasi. Pemanfaatan *flowchart* bertujuan untuk membantu *programmer* memahami langkah-langkah dalam menyelesaikan masalah dengan cara jelas, ringkas (Syamsiah, 2019).

Tabel 2. 5 Simbol *Flowchart*

Simbol	Keterangan
 <i>Flow</i>	<p>Simbol untuk menghubungkan satu simbol dengan simbol yang lainnya. Simbol ini dikenal sebagai konektor <i>line</i></p>
 <i>On-page reference</i>	<p>Digunakan sebagai penanda keluar masuk untuk menunjukkan keluar masuk atau sambungan proses dalam satu lembar kerja yang sama.</p>
 <i>Off-Page reference</i>	<p>Digunakan sebagai penunjuk untuk keluar masuk dan sambungan proses dalam lembar kerja yang berbeda.</p>
 <i>Terminator</i>	<p>Digunakan untuk mengawali dan mengakhiri program.</p>
 <i>Process</i>	<p>menunjukkan bahwa suatu proses dilakukan oleh komputer.</p>
 <i>Decision</i>	<p>Menunjukkan kondisi yang mencerminkan dan mengarahkan kepada dua pilihan jawaban Ya atau Tidak.</p>
	<p>Menggambarkan proses input output dilakukan secara langsung dan tidak</p>

Simbol	Keterangan
<i>Input/output</i>	bergantung pada perantara
 <i>Manual operation</i>	menunjukkan bahwa komputer tidak melakukan proses
 <i>Document</i>	menunjukkan bahwa masukan berasal dari dokumen fisik atau keluarannya harus dicetak

6) *Black Box Testing*

Pengujian *Black Box* menurut (Setiyani, 2019) adalah pengujian kualitas perangkat lunak yang mengutamakan fungsionalitas perangkat lunak. Menemukan fungsi yang tidak tepat, antarmuka yang kurang menarik, kesalahan struktur data, kesalahan kinerja, kesalahan inisialisasi, dan terminasi adalah tujuan utama pengujian kotak hitam. Pengujian ini dilakukan setelah program dibuat untuk mengetahui apakah program dapat beroperasi dengan baik dan sesuai dengan spesifikasi, serta untuk mengetahui apakah fungsi, *input* dan *output*nya memenuhi persyaratan tersebut.

A. Kajian Empiris

Sebelum adanya penelitian ini, terdapat beberapa penelitian terdahulu yang membahas terkait sistem informasi manajemen praktik kerja lapangan. Berikut adalah beberapa penelitian terdahulu yang sesuai dengan topik penelitian ini:

1. Penelitian yang dilakukan oleh (Jofan Rifano et al., 2022), menggunakan PHP sebagai Bahasa pemrogramannya dan MySQL sebagai *database*. Aplikasi ini dirancang untuk memudahkan proses bisnis PKL yang dilakukan oleh pemegang, pihak kampus dan pihak yang terkait dengan instansi.
2. Penelitian yang dilakukan oleh (Satria et al., 2022) pada penelitian ini, menggunakan Bahasa pemrograman PHP, *database* MySQL, *Framework Codeigniter*. Sistem ini dirancang untuk memberikan pengambilan keputusan penempatan PKL mahasiswa.
3. Penelitian yang dilakukan oleh (Anam et al., 2020) aplikasi ini dibangun dengan bahasa pemrograman PHP dengan *framework CodeIgniter* dan *database PostgreSQL*. Sistem ini dirancang untuk memudahkan tugas pegawai Tata Usaha untuk mempermudah melakukan pengelolaan data kegiatan PKL dengan mudah dan lebih terstruktur, membantu pembimbing lapangan dalam melihat report dengan jangka waktu tertentu.
4. Penelitian yang dilakukan oleh Asri & Christiawan (2023) dalam penelitian ini, Aplikasi Pendaftaran Peserta Magang dibangun menggunakan metode

Agile, Bahasa pemrograman PHP. Sistem ini dirancang untuk mempermudah proses magang, mulai dari pendaftaran dan penerimaan peserta magang, manajemen data peserta, proses pencarian pembimbing, penjadwalan serta penilaian hingga proses pembuatan laporan PKL.

B. Kerangka Berfikir

