



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Perangkat Lunak

2.1.1.1 Pengertian Perangkat Lunak

Perangkat lunak atau disebut juga dengan *software* merupakan salah satu komponen dalam komputer. Seperti namanya yaitu perangkat lunak, sifatnya berbeda dengan *hardware* atau perangkat keras, jika perangkat keras adalah komponen yang nyata yang dapat dilihat dan disentuh oleh manusia, maka *software* atau perangkat lunak tidak dapat disentuh dan dilihat secara fisik. *Software* memang tidak tampak secara fisik dan tidak berwujud benda tapi kita bisa mengoperasikannya, melalui *software* atau perangkat lunak inilah suatu komputer dapat menjalankan suatu perintah.

Menurut Rianto (2021:5), mendefinisikan bahwa "*Software* (Perangkat Lunak) merupakan suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer."

Menurut Sukamto dan Shalahuddin (2016:2), "Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)."

Berdasarkan beberapa definisi perangkat lunak diatas, dapat disimpulkan bahwa perangkat lunak atau *software* adalah instruksi-instruksi atau data yang diformat secara digital, yang bisa dibaca dan ditulis oleh komputer sesuai kehendak pemakai.



2.1.1.2 Jenis-Jenis dan Contoh Perangkat Lunak

Jenis-jenis dan contoh perangkat lunak menurut Alda (2021:4), adalah sebagai berikut :

1. Sistem Operasi, merupakan perangkat lunak yang mengoperasikan komputer serta menyediakan antarmuka dengan perangkat lunak lain atau dengan pengguna. Contoh sistem operasi: *MS DOS*, *MS Windows* (dengan berbagai generasi), *Macintosh*, *OS/2*, *UNIX* (dengan berbagai versi), *LINUX* (dengan berbagai distribusi), *NetWare*, dan lain-lain.
2. Program Utilitas, merupakan program khusus yang berfungsi sebagai perangkat pemeliharaan komputer, seperti anti virus, partisi hardisk, manajemen hardisk, dan lain-lain. Contoh produk program utilitas: *Norton Utilities*, *PartitionMagic*, *McAfee*, dan lain-lain.
3. Program Aplikasi, merupakan program yang dikembangkan untuk memenuhi kebutuhan yang spesifik. Contoh: aplikasi akuntansi, aplikasi perbankan, aplikasi manufaktur, dan lain-lain.
4. Program Paket, merupakan program yang dikembangkan untuk kebutuhan umum, seperti:
 - a. Pengolah kata / *editor* naskah: *Wordstar*, *MS Word*, *Word Perfect*, *AmiPro*, dan lain-lain.
 - b. Pengolah angka / lembar kerja: *Lotus123*, *MS Excell*, *QuattroPro*, dan lain-lain.
 - c. Presentasi: *MS PowerPoint*, dan lain-lain.
 - d. Desain Grafis: *CorelDraw*, *PhotoShop*, dan lain-lain.
5. Bahasa Pemrograman, merupakan perangkat lunak untuk pembuatan atau pengembangan perangkat lunak lain.



2.1.2 Komputer

2.1.2.1 Pengertian Komputer

Komputer semula dipergunakan sebagai alat yang dipakai untuk mengolah data serta untuk menggambarkan orang yang perkerjaannya melakukan perhitungan aritmatika dengan atau tanpa alat bantu, tetapi arti kata ini kemudian dipindahkan kepada mesin itu sendiri.

Pengertian komputer menurut Rianto (2021:3), "Komputer merupakan suatu perangkat elektronika yang memiliki kemampuan untuk menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis berdasarkan perangkat aturan tertentu."

Menurut Prawiro (dalam Harmayani, dkk, 2021:1), "Komputer adalah perangkat elektronik yang dapat digunakan untuk mengolah data sesuai dengan prosedur yang telah dirumuskan sebelumnya sehingga menghasilkan informasi bermanfaat bagi penggunanya."

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa komputer adalah alat elektronik yang dapat mengolah data digital, dengan mengikuti serangkaian program yang digunakan untuk membantu pelaksanaan pekerjaan.

2.1.2.2 Elemen-Elemen dan Sistem Komputer

Elemen-elemen dan sistem komputer adalah *software*, *hardware*, dan *brain ware* :

1. *Hardware* (Perangkat Keras) adalah peralatan di sistem komputer yang secara fisik terlihat dan dapat dijamah, seperti *monitor*, *keyboard*, *mouse*, dan lain-lain.
2. *Software* (Perangkat Lunak) adalah program yang berisi perintah-perintah untuk melakukan pengolahan data, *Software* terdiri dari :
 - a. Sistem operasi, *DOS*, *IjnuX*, *Windows*, *Mac*, dan lain-lain.
 - b. Bahasa pemrograman, *Visual Basic*, *C++*, *Pascal*, *Java*, *Visual C*, dan lain-lain.



- c. Aplikasi, *MS Office*, *Antivirus*, *Winamp*, *Mozilla*, dan lain-lain.
3. *Brain ware* adalah manusia yang terlihat dalam mengoperasikan serta mengatur sistem komputer.

Sumber : Sinaga (2020:9-10).

2.1.3 Internet

2.1.3.1 Pengertian Internet

Internet merupakan suatu perpustakaan besar yang didalamnya terdapat jutaan bahkan miliaran informasi yang berupa teks dalam bentuk media elektronik selain itu internet dikenal sebagai dunia maya, karena hampir seluruh aspek kehidupan di dunia nyata ada di *internet*. *Internet* juga banyak memberikan manfaat di berbagai bidang seperti bidang pendidikan, bidang kesehatan, bidang informasi, bidang bisnis, bidang sosial dan hiburan, dan lain sebagainya.

Pengertian *internet* menurut Ladiqi dan Wekke (2018:16-17), "*Internet* merupakan singkatan dari *International Networking* (jaringan internasional), secara teknis merupakan dua komputer atau lebih saling berhubungan membentuk jaringan komputer hingga meliputi jutaan komputer di dunia yang saling berinteraksi dan bertukar informasi."

Menurut Sobri, dkk (2017:4), "*Internet* merupakan sebuah dunia maya jaringan komputer yang berbentuk dari miliaran komputer di dunia."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *Internet* adalah jaringan global yang menghubungkan seluruh jaringan di dunia agar saling berinteraksi dan melakukan pertukaran informasi.

2.1.3.2 Fungsi dan Tujuan Internet

Internet memiliki beberapa fungsi dan tujuan sebagai berikut:

1. Memudahkan arus pertukaran informasi.
2. Menghubungkan manusia yang terpisah oleh jarak.

Sumber : Kumparan (2021).



2.1.4 Pengertian Data

Data dapat digambarkan sebagai fakta dan angka yang belum diolah, data yang belum diolah atau data mentah tidak dapat membantu dalam pengambilan keputusan, namun data adalah bahan mentah yang terorganisir, terstruktur dan diinterpretasikan untuk menciptakan sistem informasi yang berguna.

Menurut Pamungkas (2017:1), "Data merupakan nilai yang merepresentasikan deskripsi dari suatu objek atau kejadian."

Menurut Pane, dkk, (2020:7), "Data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan (observasi), dan dapat diolah menjadi bentuk yang lebih kompleks, seperti: informasi, *database*, atau solusi untuk masalah tertentu."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa data adalah kumpulan atas angka, karakter, dan fakta yang akan diklasifikasi, dimodifikasi, dan diolah oleh program agar dapat menjadi informasi yang dapat berguna bagi penggunanya.

2.1.5 Pengertian Basis Data (*Database*)

Istilah basis data atau *database* berawal dari ilmu komputer. Basis data mengacu pada koleksi dari data-data yang saling berhubungan dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (*Database Management System/DBMS*).

Menurut Pamungkas (2017:2), "Basis data merupakan kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu."

Menurut Sukamto dan Salahuddin (2016:43) menyatakan bahwa "Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan."



Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang saling berhubungan, diorganisasi secara sistematis, menggunakan program komputer agar dapat dimanipulasi dan diperoleh dengan cepat dan mudah.

2.1.6 Metode Pengembangan Sistem

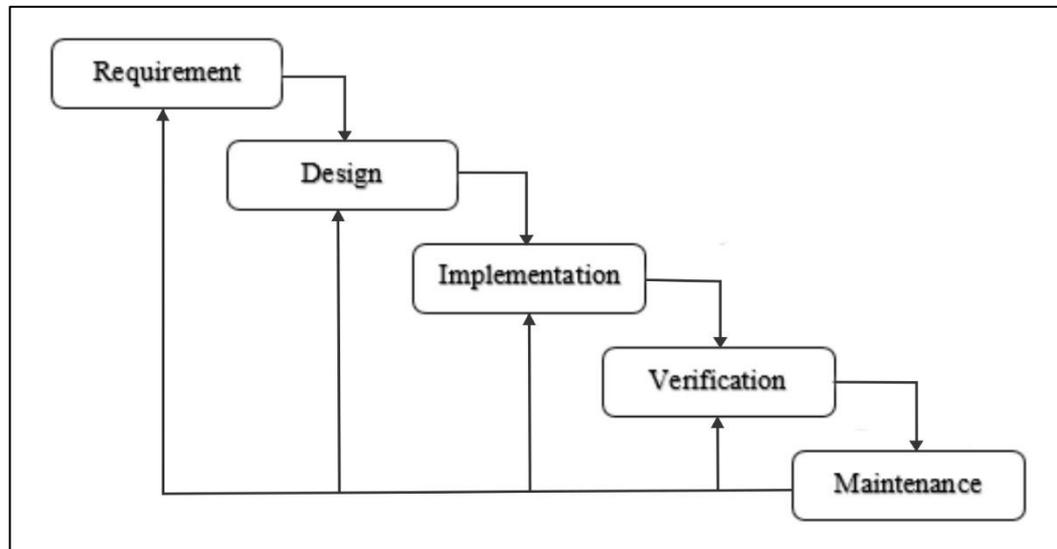
Penelitian Laporan Akhir ini menggunakan metode pengembangan sistem Waterfall. Menurut Pressman (dikutip Sanubari, dkk, 2020:41), menyatakan bahwa "Metode air terjun atau yang sering disebut metode waterfall sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan."

Menurut Teknika: Jurnal Sains dan Teknologi (2020:269), menyatakan bahwa "Metode Waterfall adalah suatu proses pengembangan perangkat lunak berurutan, dimana kemajuan teknologi dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa Metode Waterfall adalah suatu proses atau metode pengembangan sistem secara berurutan yang mengalir ke bawah (seperti air terjun) melalui tahapan-tahapan *Requirement Analysis, Design System, Implementation, Verification & Testing*, dan *Maintenance*.



Tahapan metode waterfall dapat dilihat pada gambar di bawah ini:



Sumber : Sanubari, dkk (2020:41)

Gambar 2.1 Tahapan-Tahapan Metode Waterfall

Berdasarkan gambar di atas, pengembangan sistem dengan menggunakan metode waterfall memiliki beberapa tahapan yang berurutan yaitu: *Requirement* (analisis kebutuhan), *Design System* (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian), Penerapan Program, Pemeliharaan. Tahapan-tahapan dari metode waterfall adalah sebagai berikut:

a. *Requirement Analisis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

b. *System Design*

Spesifikasi kebutuhan tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem dipersiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.



c. *Implementation*

Tahap ini dimana sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

d. *Integration & Testing*

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

e. *Operation & Maintenance*

Operation & Maintenance merupakan tahap akhir dalam model waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaiki implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Sumber : Sanubari, dkk (2020:41-42).

2.1.7 Studi Kelayakan

Menurut Sutrisno dikutip oleh Suprpto dalam Jurnal nya (Pemahaman Umum Studi Kelayakan Proyek *Infrastuktur*) mengemukakan bahwa Pengertian Studi Kelayakan (*Feasibility Study*) adalah suatu studi atau pengkajian apakah suatu usulan proyek/gagasan usaha apabila dilaksanakan dapat berjalan dan berkembang sesuai dengan tujuannya atau tidak. Objek atau subjeck materi studi kelayakan adalah usulan proyek/gagasan usaha. Usulan proyek/gagasan usaha tersebut dikaji, diteliti, dan diselidiki dari berbagai aspek tertentu apakah memenuhi persyaratan untuk dapat berkembang atau tidak. Dalam studi kelayakan yang distudi (diteliti) misalnya aspek pemasaran, aspek tehnik, aspek proses termasuk input, out put dan pemasaran, aspek komersial, aspek yuridis, aspek



social budaya, aspek paedagogis dan aspek ekonomi. Sedangkan Menurut Yacob Ibrahim dikutip oleh Suprpto dalam Jurnal nya (Pemahaman Umum Studi Kelayakan Proyek *Infrastuktur*) mengemukakan bahwa Studi Kelayakan (*feasibility study*) adalah kegiatan untuk menilai sejauh mana manfaat yang dapat diperoleh dalam melaksanakan suatu kegiatan usaha /proyek dan merupakan bahan pertimbangan dalam mengambil suatu keputusan, apakah menerima atau menolak dari suatu gagasan usaha /proyek yang direncanakan. Pengertian layak dalam penilaian ini adalah kemungkinan dari gagasan usaha/proyek yang akan dilaksanakan memberikan manfaat (*benefit*), baik dalam arti *financial benefit* maupun dalam arti *social benefit*.

Dari kedua pendapat tentang pengertian Studi Kelayakan diatas dapatlah disimpulkan bahwa Studi kelayakan adalah kegiatan menganalisa, mengkaji dan meneliti berbagai aspek tertentu suatu gagasan usaha/proyek yang akan dilaksanakan atau telah dilaksanakan, sehingga memberi gambaran layak (*feasible-go*) atau tidak layak (*no feasible-no go*) suatu gagasan usaha/proyek apabila ditinjau dari manfaat yang dihasilkan (*benefit*) dari proyek/gagasan usaha tersebut baik dari sudut *financial benefit* maupun *social benefit*.

2.1.8 Kebutuhan Fungsional

Menurut Hadiprakoso Raden Budiarto (2020:88), Kebutuhan Fungsional mendefinisikan fungsi atau *fitur* dari sebuah perangkat lunak, fungsi tersebut dapat berupa input dan output, proses perhitungan, rincian teknis, manipulasi dan pengolahan data atau fungsi spesifik lainnya yang seharusnya dipenuhi oleh sistem.

2.1.9 Kebutuhan Non-Fungsional

Menurut Hadiprakoso Raden Budiarto (2020:88), Kebutuhan Non-Fungsional merupakan batasan pada layanan atau fungsi yang ditawarkan oleh sistem (Kebutuhan Non-Fungsional merupakan syarat pendukung). Batasan ini meyangkut masalah kinerja, keandalan atau standar tertentu.



2.2 Teori Khusus

2.2.1 Kamus Data

Kamus data berisi katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Menurut Sukamto dan Shalahuddin (2016:73), mengungkapkan bahwa "Kamus Data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD yang berupa kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)."

Menurut Rusmawan (2019:36), mendefinisikan bahwa "Kamus data merupakan katalog fakta data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi."

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa Kamus Data merupakan katalog yang berisi kumpulan fakta data agar proses *input* dan *output* pada sistem perangkat lunak dapat dengan mudah dipahami.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

Tabel 2.1 Simbol-Simbol pada Kamus Data

No.	Simbol	Keterangan
1.	=	disusun atau terdiri dari
2.	+	dan
3.	[]	baik ...atau...
4.	{ } ⁿ	n kali diulang/ bernilai banyak
5.	()	data opsional
6.	*...*	batas komentar

Sumber : Sukamto dan Shalahuddin (2016:74).



Kamus data ini sangat membantu analisis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur.

2.2.2 *Unified Modeling Language (UML)*

2.2.2.1 Pengertian *Unified Modeling Language (UML)*



Sumber : commons.m.wikipedia.org (2021)

Gambar 2.2 Logo UML

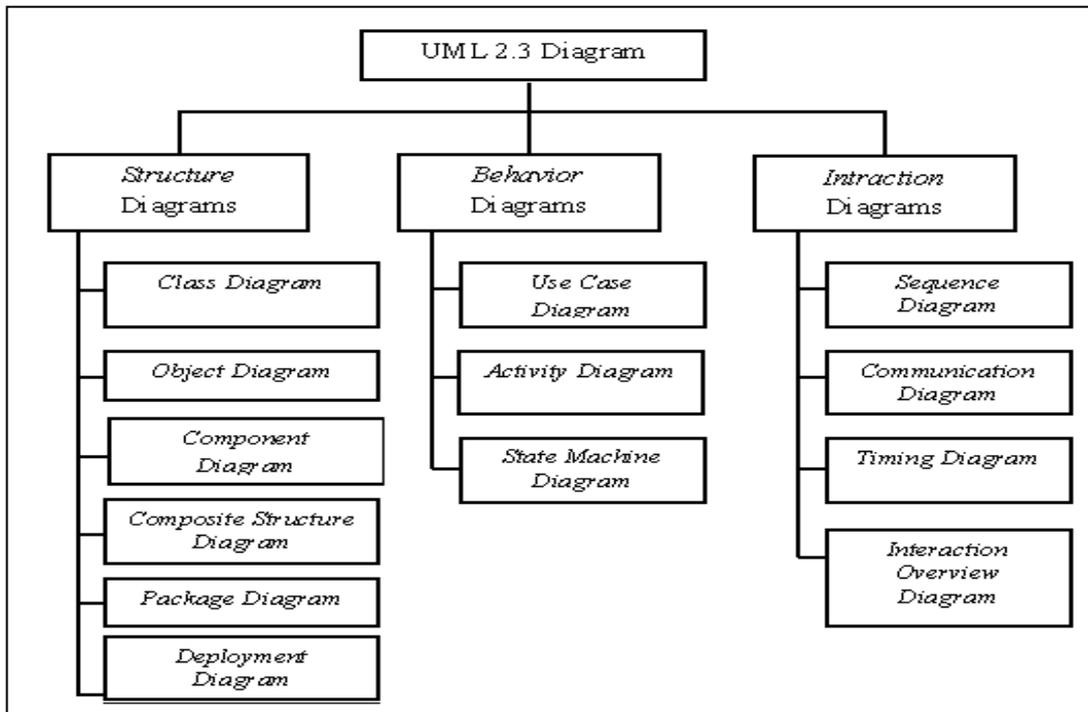
Menurut Haqi (2019:29), "UML merupakan singkatan dari *“Unified Modelling Language”* yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*."

Menurut Herlinah dan Musliadi (2019:50), "UML merupakan salah satu alat bantu pengembangan sistem berorientasi objek."

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa UML (*Unified Modelling Language*) adalah metode atau alat bantu dalam melakukan permodelan dan pengembangan sistem yang berorientasi objek.

2.2.2.2 Kategori *Unified Modeling Language (UML)*

Unified Modeling Language (UML) terdiri dari 13 macam diagram dan dikelompokkan menjadi 3 kategori sebagai berikut:



Sumber : Sukamto dan Shalahuddin (2016:140).

Gambar 2.3 Kategori dan Macam-Macam Diagram UML

Berikut penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2016:141):

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



2.2.3 Jenis-Jenis Diagram *Unified Modeling Language* (UML)

2.2.3.1 Pengertian *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2016:155), "*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat."

Menurut Setyawati, dkk (2021:13), "*Use case diagram* merupakan konsep informasi dengan menggunakan fungsi apa saja yang ada di dalam sistem informasi & siapa saja *actor* yang terlibat."

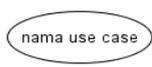
Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *Use Case Diagram* adalah suatu permodelan atau konsep informasi untuk mengetahui kelakuan sistem informasi dengan mendeskripsikan interaksi antara aktor dan sistem informasi.

Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

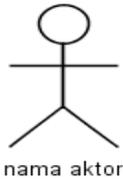
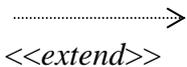
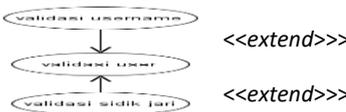
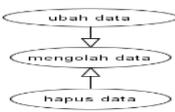
- a. **Aktor** merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. ***Use case*** merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Simbol-simbol pada *Use Case Diagram* adalah sebagai berikut:

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>

Lanjutan Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
2.	aktor / <i>actor</i> 	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
3.	asosiasi/ <i>association</i> 	komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki ingeraksi dengan actor.
4.	ekstensi / <i>extend</i> 	relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan sama dengan <i>use case</i> yang ditambahkan, misalnya  arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya
5.	Generalisasi / <i>generalization</i> 	hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).

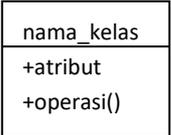
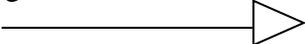
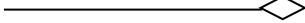


suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi yang terdapat pada sistem tersebut."

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *Class Diagram* adalah permodelan yang menggambarkan struktur sistem yang digunakan untuk menampilkan kelas atau paket mengenai sistem dan juga relasi yang ada pada sistem.

Simbol-simbol pada *Class Diagram* adalah sebagai berikut:

Tabel 2.3 Simbol-Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem.
2	antarmuka / <i>interface</i>  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .
4.	asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber : Sukamto dan Shalahuddin (2016:146-147).



2.2.3.2 Pengertian Activity Diagram

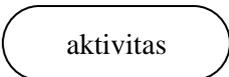
Menurut Sukamto dan Shalahuddin (2016:161), "Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem."

Menurut Haqi (2019:31), "*Activity diagram* atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem."

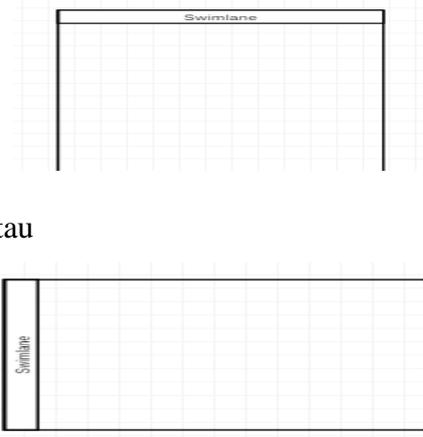
Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *Activity Diagram* merupakan permodelan yang menggambarkan aliran kerja mengenai proses-proses apa saja yang terjadi di dalam sebuah sistem yang ada pada perangkat lunak.

Simbol-simbol pada *Activity Diagram* adalah sebagai berikut:

Tabel 2.4 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
4.		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

Lanjutan Tabel 2.4 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Sukamto dan Shalahuddin (2016:162-163).

2.2.3.2 Pengertian *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2016:165), "Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat *scenario* yang ada pada *use case*."

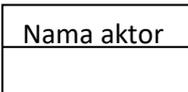
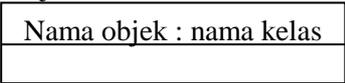
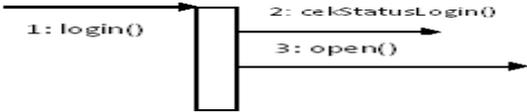
Menurut Haqi (2019:31), "*Sequence diagram* yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence diagram* juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*."



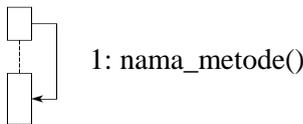
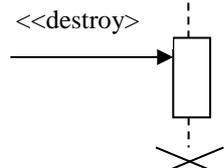
Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *Sequence Diagram* merupakan permodelan yang mendeskripsikan interaksi objek dengan waktu hidup yang dikirimkan dan diterima objek, dan dapat menggambarkan tahapan yang harus dilakukan oleh sistem untuk memenuhi kebutuhan yang ada pada *use case*.

Simbol-simbol pada *Sequence Diagram* adalah sebagai berikut:

Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p><i>Actor</i></p>  <p>Atau</p>  <p>tanpa waktu aktif</p>	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2.	<p>Garis hidup / <i>lifeline</i></p> 	menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cek Status Login () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif.</p>

Lanjutan Tabel 2.5 Simbol-Simbol *Sequence* Diagram

No	Simbol	Deskripsi
5.	Pesan tipe <i>create</i> 	menyatakan objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i>  	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe <i>send</i> 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Sukamto dan Shalahuddin (2016:165-167).



2.3 Teori Judul

2.3.1 Pengertian Perancangan

Perancangan memiliki peranan penting sebelum proses pembuatan aplikasi. Perancangan memberikan gambaran mengenai aplikasi yang akan dibuat, bagaimana perencanaan dan pembuatan sketsa dari beberapa elemen yang terpisah ke dalam satu kesatuan aplikasi yang utuh dan dapat berfungsi dengan baik.

Menurut Nur dan Suyuti (2018:5), mendefinisikan bahwa "Perancangan adalah suatu proses yang bertujuan untuk menganalisis, menilai, memperbaiki, dan menyusun suatu sistem, baik sistem fisik maupun non fisik yang optimum untuk waktu yang akan datang dengan memanfaatkan informasi yang ada."

Menurut Prasetyowibowo (Dikutip Prasnowo, dkk, 2020:22), mendefinisikan bahwa "Perancangan (*design*) merupakan suatu kegiatan atau rekayasa rancang bangun yang dimulai dari ide-ide inovasi desain, atau kemampuan untuk menghasilkan karya dan cipta yang benar-benar dapat menjabarkan permintaan pasar karena adanya penelitian dan pengembangan teknologi."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa perancangan adalah rekayasa rancang bangun untuk memberikan gambaran secara umum dan fungsionalitas dari aplikasi yang akan dibuat kepada pemakai.

2.3.2 Pengertian Aplikasi

Menurut Fathoroni, dkk (2020:1), menyatakan bahwa "Aplikasi adalah sebuah program siap pakai yang bisa digunakan untuk menjalankan sejumlah perintah dari pengguna aplikasi itu sendiri." Sedangkan, menurut Putra (2020), mendefinisikan "Aplikasi adalah sebuah perangkat lunak atau program yang diciptakan dan dikembangkan untuk melakukan tugas-tugas tertentu pada perangkat komputer, laptop, ataupun *smartphone*."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa aplikasi merupakan perangkat lunak (*software*) yang dibuat untuk menolong beberapa aktivitas manusia dalam melaksanakan tugas tertentu.



2.3.3 Pengertian Pengelolaan

Menurut Kamus Besar Bahasa Indonesia, "Pengelolaan adalah proses yang memberikan pengawasan kepada semua hal yang terlibat dalam pelaksanaan kebijaksanaan dan pencapaian tujuan". Sedangkan, menurut Tim Dosen PAI (2016:104), menyatakan bahwa "Pengelolaan adalah suatu proses untuk menggerakkan, mengorganisasikan, dan mengerahkan usaha manusia untuk mencapai tujuannya".

Maka, dapat disimpulkan bahwa Pengelolaan adalah suatu proses yang dilakukan oleh manusia untuk melakukan pengawasan, pengorganisasian, dan menggerakkan pelaksanaan kebijakan untuk mencapai tujuan.

2.3.4 Pengertian Data

Menurut Yusuf dan Daris (2018:3), mendefinisikan bahwa "Data didefinisikan sebagai deskripsi atau keterangan sebuah objek yang belum memiliki makna secara utuh, dapat berupa angka (*numeric*), karakter (*text*), gambar (*image*), suara (*sound*) ataupun lambang (*symbol*)". Sedangkan menurut Pane, dkk, (2020:17), "Data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan (*observasi*), dan dapat diolah menjadi bentuk yang lebih kompleks, seperti: informasi, database, atau solusi untuk masalah tertentu".

Dengan demikian, dapat disimpulkan bahwa data adalah sekumpulan keterangan atau deskripsi dasar sebuah objek atau kejadian yang belum memiliki makna secara utuh, dapat berupa angka (*numeric*), karakter (*text*), gambar (*image*), suara (*sound*) ataupun lambang (*symbol*) yang dapat diolah menjadi bentuk yang lebih kompleks agar menghasilkan suatu informasi.

2.3.5 Pengertian Pegawai

Menurut Soedaryono Dikutip oleh Pujiarsih dalam Buku Otomatisasi TataKelola Kepegawaian (2018:4) mendefinisikan bahwa "Pegawai adalah seorang yang melakukan penghidupan dengan bekerja dalam kesatuan organisasi



baik kesatuan pemerintah maupun kesatuan swasta". Sedangkan, menurut Robbins dalam Buku Perilaku Organisasi (2016), menjelaskan bahwa "Pegawai adalah orang pribadi yang bekerja pada pemberi kerja, baik sebagai pegawai tetap atau tidak, berdasarkan kesepakatan kerja baik tertulis maupun tidak tertulis, untuk melaksanakan suatu pekerjaan dalam jabatan atau kegiatan tertentu yang ditetapkan oleh pemberi kerja".

Berdasarkan penjelasan di atas dapat disimpulkan bahwa Pegawai adalah seseorang yang bekerja pada kesatuan organisasi, badan usaha baik pemerintah maupun swasta, baik sebagai pegawai tetap ataupun tidak yang diberikan imbalan kerja sesuai dengan pemberi kerja.

2.3.6 Pengertian Cuti

Menurut Pujiarsi dan Budiningsih (2019:55), menyatakan bahwa "Cuti adalah hak Pegawai Negeri Sipil yang diberikan dalam rangka usaha menjamin kesegaran jasmani dan rohani kepada PNS setelah bekerja selama jangka waktu tertentu perlu diberikan cuti". Sedangkan, Menurut Prawono, (2020:122), mendefinisikan bahwa "Cuti adalah keadaan tidak masuk kerja yang diizinkan dalam jangka waktu tertentu. Cuti diberikan dalam rangka usaha jasmani dan rohani".

Maka, dapat disimpulkan bahwa Cuti adalah Keadaan tidak masuk kerja yang diizinkan dalam jangka waktu tertentu yang diberikan kepada setiap pegawai ataupun karyawan".

2.3.7 Pengertian Kerja

Menurut Kamus Besar Bahasa Indonesia, "Kerja adalah kegiatan melakukan sesuatu yang dilakukan ataupun diperbuat". Sedangkan, dalam situs Wikipedia, (2020) "Kerja adalah Aktivitas yang dilakukan manusia untuk menghasilkan sesuatu".

Maka, dapat disimpulkan Kerja adalah Suatu kegiatan yang dilakukan setiap orang/ manusia guna untuk menghasilkan sesuatu biasanya berupa uang



sebagai balas jasa yang diberikan atas kegiatan atau perbuatan yang telah dilakukan”.

2.3.8 Pengertian *Website*

Menurut Sa’ad (2020:3), “*Website* merupakan halaman-halaman yang berisi informasi yang dapat diakses oleh *browser* dan mampu memberikan informasi yang berguna bagi para pemkainya”. Sedangkan, menurut Marisa (2017:1), “*Website* dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk mempublikasikan informasi berupa teks, gambar, dan program multimedia lainnya berupa animasi (gambar gerak, tulisan gerak), suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait antara satu halaman dengan halaman yang lain yang sering disebut sebagai *hyperlink*”.

Maka, dapat disimpulkan bahwa *Website* adalah halaman yang berisi informasi berupa teks, gambar, suara, dan video yang dapat disimpan pada komputer lokal maupun *server web*, yang terhubung melalui link *hypertext* dan dapat dijadikan atau ditampilkan pada browser.

2.3.9 Pengertian Aplikasi Pengelolaan Data Pegawai dan Pengajuan Cuti Kerja Berbasis *Website*

Aplikasi Pengelolaan Data Pegawai dan Pengajuan Cuti Kerja Berbasis *Website* adalah aplikasi untuk memudahkan perusahaan dalam melakukan proses pengelolaan data pegawai dan data cuti kerja yang lebih terstruktur, efektif, dan efisien serta memudahkan pegawai dalam proses pengajuan cuti kerja. Dengan adanya aplikasi ini, perusahaan dan pegawai dapat melakukan proses pengelolaan data pegawai, data cuti kerja, dan pengajuan cuti kerja menggunakan koneksi internet yang dimana aplikasi ini mempunyai *Security* yang aman. Perusahaan dan pegawai pun dapat mengetahui mengenai perhitungan data pegawai tetap maupun *Honorer* dan perhitungan jumlah cuti kerja setiap pegawai.



2.4 Teori Program

2.4.1 Pengertian HTML



Sumber : wikipedia.org (2021)

Gambar 2.4 Logo *JavaScript*

Pengertian HTML menurut Surya dan Jannah (2020:1), "HTML adalah bahasa markah yang digunakan peramban untuk menafsirkan dan menulis teks, gambar, dan bahan lainnya ke dalam halaman web secara virtual maupun suara."

Menurut Devi (2020:1), "*Hypertext Markup Language* (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi dalam sebuah penjelajah web internet dan memformat hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang teritegrasi."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa HTML (*Hypertext Markup Language*) adalah skrip yang berupa tag-tag untuk membuat dan mengatur struktur serta menandai bagian-bagian dari sebuah halaman website untuk menampilkan konten di web.

2.4.2 Pengertian *JavaScript*



Sumber : pngwing.com (2021)

Gambar 2.5 Logo *JavaScript*

Menurut Siahaan dan Sianipar, (2020:1), "Javascript adalah sebuah skript dinamis yang dapat dipakai untuk membangun interaktifitas pada halaman-halaman HTML statis."



Menurut Sulistiono (2018:3), "Javascript adalah bahasa pemrograman tingkat tinggi untuk membuat website dinamis."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa JavaScript adalah bahasa pemrograman web yang pemrosesannya dilakukan di sisi client untuk memberikan efek dinamis, interaktif dan bersifat independen, yang dieksekusi di browser.

2.4.3 Pengertian CSS



Sumber : pngegg.com (2021)

Gambar 2.6 Logo CSS

Menurut Abdulloh (2016:2), "CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain website."

Menurut Enterprise (2016:94), "CSS merupakan kumpulan kode untuk mendesain atau mempercantik tampilan halaman *website*."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa CSS adalah bahasa pengkodean atau dokumen yang berfungsi mengatur elemen HTML dengan berbagai *property* untuk menata gaya tampilan halaman web agar lebih cantik dan indah saat di tampilkan di web browser.

2.4.4 Pengertian PHP



Sumber : php.net (2021)

Gambar 2.7 Logo PHP



Menurut Supono dan Putratama (2018:3), "PHP (PHP: *Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML."

Menurut Abdulloh (2016:3), "PHP singkatan dari *Hypertext Preprocessor*, yaitu bahasa pemrograman yang diproses di sisi server."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa PHP adalah bahasa pemrograman web berbasis *server-side programming* (bahasa pemrograman yang diproses disisi server) yang memparsing kode PHP dari kode web dengan ekstensi (.php), yang kemudian akan dikirim ke browser web.

2.4.5 Pengertian CodeIgniter



Sumber : forum.codeigniter.com (2021)

Gambar 2.8 Logo CodeIgniter

Menurut Supono dan Putratama (2018:109), mendefinisikan bahwa "CodeIgniter adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP."

Menurut Habibi dan Aprilian (2019:9), mendefinisikan bahwa "CodeIgniter adalah sebuah web *application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi php dinamis."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa CodeIgniter adalah sebuah *framework* yang bersifat *open source* untuk membangun *website* / aplikasi dengan menggunakan bahasa pemrograman PHP.



2.4.6 Pengertian XAMPP



Sumber : mrenesiauw.web.id (2021)

Gambar 2.9 Logo XAMPP

Menurut Habibi, dkk (2020:5), menyatakan bahwa "XAMPP merupakan perangkat lunak bebas, yang mendukung banyak *system* operasi merupakan kompilasi dari beberapa program, *xampp* adalah perangkat yang menggabungkan tiga aplikasi ke dalam satu paket yaitu *Apache*, *MySQL*, 6 dan *PHP my admin* dengan *xampp* pekerjaan anda sangat dimudahkan karena dapat menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dengan sekaligus dan otomatis."

Menurut Nurhidayah, dkk (2020:40), menyatakan bahwa "XAMPP adalah aplikasi yang bersifat *open source* yang bisa digunakan di banyak sistem operasi seperti Windows, Linux, dan Mac OS."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa XAMPP merupakan aplikasi pendukung bersifat *open source* yang menggabungkan tiga aplikasi ke dalam satu paket yaitu *Apache*, *MySQL*, 6 dan *PHP my admin* untuk membaca file yang memiliki format PHP.

2.4.7 MySQL

2.4.7.1 Pengertian MySQL



Sumber : stickpng.com (2021)

Gambar 2.10 Logo MySQL

Menurut Fathoroni, dkk (2020:55), "MySQL adalah *database management system* yang menggunakan bahasa *SQL* sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*."



Menurut Fitri (2020:7), "MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database SQL* sebagai bahasa interaktif dalam mengelola data."

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa MySQL adalah *database management system* sebagai bahasa interaktif penghubung perangkat aplikasi dengan *database server* yang digunakan dalam melakukan pengelolaan data.

2.4.7.2 Fungsi-Fungsi MySQLi

Versi terkini, berintegrasi dengan bahasa pemrograman PHP versi terbaru (5.6-7.2), MySQL berganti fungsi menjadi MySQLi (*MySQL improved*). Berikut adalah fungsi-fungsi dari MySQLi.

Tabel 2.6 Fungsi-Fungsi MySQLi

No.	Fungsi MySQLi	Keterangan
1.	<code>mysqli_affected_rows()</code>	Mengembalikan jumlah baris yang berpengaruh dalam operasi MySQL sebelumnya.
2.	<code>mysqli_autocommit()</code>	Menghidupkan atau mematikan modifikasi basis data yang dilakukan secara otomatis.
3.	<code>mysqli_change_user()</code>	Mengubah pengguna koneksi <i>database</i> yang ditentukan.
4.	<code>mysqli_character_set_name()</code>	Mengembalikan <i>set</i> karakter default untuk koneksi <i>database</i> .
5.	<code>mysqli_close()</code>	Menutup koneksi <i>database</i> yang sebelumnya dibuka.
6.	<code>mysqli_commit()</code>	Melakukan transaksi saat ini.
7.	<code>mysqli_connect_errno()</code>	Mengembalikan kode kesalahan dari kesalahan koneksi terakhir.



Lanjutan Tabel 2.6 Fungsi-Fungsi MySQLi

No.	Fungsi MySQLi	Keterangan
8.	<code>mysqli_connect_error()</code>	Mengembalikan deskripsi kesalahan dari kesalahan koneksi terakhir.
9.	<code>mysqli_connect()</code>	Membuka koneksi baru ke server MySQL.
10.	<code>mysqli_data_seek()</code>	Menyesuaikan pointer hasil ke baris arbitrer di set hasil.
11.	<code>mysqli_debug()</code>	Melakukan operasi <i>debugging</i> .
12.	<code>mysqli_dump_debug_info()</code>	Tuang info debug ke dalam log.
13.	<code>mysqli_free_result()</code>	Membebaskan memori yang terkait dengan hasil.
14.	<code>mysqli_errno()</code>	Mengembalikan kode kesalahan terakhir untuk panggilan fungsi terbaru.
15.	<code>mysqli_error_list()</code>	Mengembalikan daftar kesalahan untuk panggilan fungsi terbaru.
16.	<code>mysqli_error()</code>	Mengembalikan deskripsi kesalahan terakhir untuk panggilan fungsi terbaru.
17.	<code>mysqli_fetch_all()</code>	Mengambil semua baris hasil sebagai <i>array asosiatif</i> , <i>array numerik</i> , atau keduanya.
18.	<code>mysqli_fetch_array()</code>	Mengambil baris hasil sebagai <i>asosiatif</i> , <i>array numerik</i> , atau keduanya.
19.	<code>mysqli_fetch_assoc()</code>	Mengambil baris hasil sebagai <i>array asosiatif</i> .
20.	<code>mysqli_fetch_field_direct()</code>	Mengembalikan meta-data untuk bidang tunggal di <i>set</i> hasil, sebagai objek.
21.	<code>mysqli_fetch_field()</code>	Mengembalikan bidang berikutnya dalam <i>set</i> hasil, sebagai objek.



Lanjutan Tabel 2.6 Fungsi-Fungsi MySQLi

No.	Fungsi MySQLi	Keterangan
22.	<code>mysqli_fetch_fields()</code>	Mengembalikan <i>array</i> objek yang mewakili bidang dalam <i>set</i> hasil.
23.	<code>mysqli_init()</code>	Menginisialisasi MySQLi dan mengembalikan sumber daya untuk digunakan dengan <code>mysqli_real_connect()</code> .
24.	<code>mysqli_fetch_lengths()</code>	Mengembalikan panjang kolom dari baris saat ini di <i>set</i> hasil.
25.	<code>mysqli_fetch_object()</code>	Mengembalikan baris saat ini dari hasil yang ditetapkan, sebagai objek.
26.	<code>mysqli_field_count()</code>	Mengembalikan jumlah kolom untuk kueri terbaru.
27.	<code>mysqli_field_tell()</code>	Mengembalikan posisi kursor bidang.
28.	<code>mysqli_fetch_row()</code>	Mengambil satu baris dari hasil- <i>set</i> dan mengembalikannya sebagai <i>array</i> .
29.	<code>mysqli_insert_id()</code>	Mengembalikan id yang dibuat secara otomatis yang digunakan dalam permintaan terakhir.
30.	<code>mysqli_info()</code>	Mengembalikan informasi tentang kueri yang baru saja dieksekusi.
31.	<code>mysqli_insert_id()</code>	Mengembalikan id yang dibuat secara otomatis yang digunakan dalam permintaan terakhir.
32.	<code>mysqli_more_results()</code>	Cek apakah ada lebih banyak hasil dari kueri multi.
33.	<code>mysqli_multi_query()</code>	Melakukan satu / lebih <i>query</i> pada <i>database</i> .
34.	<code>mysqli_next_result()</code>	Mempersiapkan <i>set</i> hasil selanjutnya dari <code>mysqli_multi_query()</code> .



Lanjutan Tabel 2.6 Fungsi-Fungsi MySQLi

No.	Fungsi MySQLi	Keterangan
35.	<code>mysqli_get_charset()</code>	Mengembalikan objek <i>set</i> karakter.
36.	<code>mysqli_get_client_info()</code>	Mengembalikan versi pustaka MySQL.
37.	<code>mysqli_fetch_array()</code>	Mengambil baris hasil sebagai asosiatif, array numerik, atau keduanya.
38.	<code>mysqli_fetch_assoc()</code>	Mengambil baris hasil sebagai <i>array asosiatif</i> .
39.	<code>mysqli_fetch_field_direct()</code>	Mengembalikan meta-data untuk bidang tunggal di <i>set</i> hasil, sebagai objek.
40.	<code>mysqli_fetch_field()</code>	Mengembalikan bidang berikutnya dalam <i>set</i> hasil, sebagai objek.
41.	<code>mysqli_fetch_fields()</code>	Mengembalikan <i>array</i> objek yang mewakili bidang dalam <i>set</i> hasil.
42.	<code>mysqli_init()</code>	Menginisialisasi MySQLi dan mengembalikan sumber daya untuk digunakan dengan <code>mysqli_real_connect()</code> .
43.	<code>mysqli_fetch_lengths()</code>	Mengembalikan panjang kolom dari baris saat ini di <i>set</i> hasil.
44.	<code>mysqli_fetch_object()</code>	Mengembalikan baris saat ini dari hasil yang ditetapkan, sebagai objek.
45.	<code>mysqli_field_count()</code>	Mengembalikan jumlah kolom untuk kueri terbaru.
46.	<code>mysqli_field_tell()</code>	Mengembalikan posisi kursor bidang.
47.	<code>mysqli_fetch_row()</code>	Mengambil satu baris dari hasil-set dan mengembalikannya sebagai array.
48.	<code>mysqli_insert_id()</code>	Mengembalikan id yang dibuat secara otomatis yang digunakan dalam permintaan terakhir.



Lanjutan Tabel 2.6 Fungsi-Fungsi MySQLi

No.	Fungsi MySQLi	Keterangan
49.	<code>mysqli_info()</code>	Mengembalikan informasi tentang kueri yang baru saja dieksekusi.
50.	<code>mysqli_insert_id()</code>	Mengembalikan id yang dibuat secara otomatis yang digunakan dalam permintaan terakhir.
51.	<code>mysqli_more_results()</code>	Cek apakah ada lebih banyak hasil dari kueri multi.
52.	<code>mysqli_multi_query()</code>	Melakukan satu atau lebih query pada <i>database</i> .
53.	<code>mysqli_next_result()</code>	Mempersiapkan <i>set</i> hasil selanjutnya dari <code>mysqli_multi_query()</code> .
54.	<code>mysqli_get_charset()</code>	Mengembalikan objek <i>set</i> karakter.
55.	<code>mysqli_get_client_info()</code>	Mengembalikan versi pustaka MySQL.
56.	<code>mysqli_get_client_stats()</code>	Mengembalikan statistik tentang klien per-proses.
57.	<code>mysqli_get_client_version()</code>	Mengembalikan versi pustaka klien MySQL sebagai <i>integer</i> .
58.	<code>mysqli_get_connection_stats()</code>	Mengembalikan statistik tentang koneksi klien.
59.	<code>mysqli_get_host_info()</code>	Mengembalikan nama <i>host server</i> MySQL dan jenis koneksi.
60.	<code>mysqli_get_proto_info()</code>	Mengembalikan versi protokol MySQL.
61.	<code>mysqli_get_server_info()</code>	Mengembalikan versi server MySQL.
62.	<code>mysqli_get_server_version()</code>	Mengembalikan versi <i>server</i> MySQL sebagai bilangan bulat.
63.	<code>mysqli_kill()</code>	Meminta <i>server</i> untuk membunuh utas MySQL.



Lanjutan Tabel 2.6 Fungsi-Fungsi MySQLi

No.	Fungsi MySQLi	Keterangan
64.	<code>mysqli_num_fields()</code>	Mengembalikan jumlah bidang dalam <i>set</i> hasil.
65.	<code>mysqli_refresh()</code>	Me- <i>refresh</i> tabel atau <i>cache</i> , atau me- <i>reset</i> informasi server replikasi.
66.	<code>mysqli_rollback()</code>	Putar kembali transaksi saat ini untuk basis data.
67.	<code>mysqli_select_db()</code>	Mengubah <i>database</i> default untuk koneksi.
68.	<code>mysqli_set_charset()</code>	Menetapkan <i>set</i> karakter klien default.
69.	<code>mysqli_set_local_infile_</code> <code>default()</code>	<i>Unsets handler</i> yang ditentukan pengguna untuk memuat perintah <i>infile</i> lokal.
70.	<code>mysqli_set_local_infile_</code> <code>handler()</code>	Atur fungsi panggilan balik untuk perintah <i>LOAD DATA LOCAL INFILE</i> .
71.	<code>mysqli_sqlstate()</code>	Mengembalikan kode kesalahan <i>SQLSTATE</i> untuk operasi MySQL terakhir.
72.	<code>mysqli_num_rows()</code>	Mengembalikan jumlah baris dalam hasil yang ditetapkan.
73.	<code>mysqli_options()</code>	Menetapkan opsi koneksi tambahan dan memengaruhi perilaku koneksi.
74.	<code>mysqli_ping()</code>	Ping koneksi <i>server</i> , atau mencoba menyambung kembali jika koneksi sudah turun.
75.	<code>mysqli_prepare()</code>	Mempersiapkan pernyataan SQL untuk dieksekusi.
76.	<code>mysqli_query()</code>	Melakukan kueri terhadap basis data.



Lanjutan Tabel 2.6 Fungsi-Fungsi MySQLi

No.	Fungsi MySQLi	Keterangan
77.	<code>mysqli_real_connect()</code>	Membuka koneksi baru ke <i>server</i> MySQL.
78.	<code>mysqli_real_escape_string()</code>	Melepaskan karakter khusus dalam <i>string</i> untuk digunakan dalam pernyataan SQL.
79.	<code>mysqli_real_query()</code>	Menjalankan <i>query</i> SQL.
80.	<code>mysqli_reap_async_query()</code>	Mengembalikan hasil dari kueri <i>async</i> .
81.	<code>mysqli_thread_id()</code>	Mengembalikan ID utas untuk koneksi saat ini.
82.	<code>mysqli_ssl_set()</code>	Digunakan untuk membuat koneksi yang aman menggunakan SSL.
83.	<code>mysqli_stat()</code>	Mengembalikan status sistem saat ini.
84.	<code>mysqli_stmt_init()</code>	Menginisialisasi pernyataan dan mengembalikan objek untuk digunakan dengan <code>mysqli_stmt_prepare()</code> .
85.	<code>mysqli_store_result()</code>	Mentransfer hasil yang ditetapkan kueri terakhir.
86.	<code>mysqli_thread_safe()</code>	Mengembalikan apakah pustaka klien dikompilasi sebagai aman-utas.
87.	<code>mysqli_use_result()</code>	pengambilan hasil yang disetel dari kueri terakhir yang dieksekusi <code>mysqli_real_query()</code> .
88.	<code>mysqli_warning_count()</code>	Mengembalikan jumlah peringatan dari kueri terakhir dalam koneksi.

Sumber : W3school (2019).



2.4.8 Pengertian *Sublime Text* dan Karakteristiknya



Sumber : en.m.wikipedia.org (2021)

Gambar 2.11 Logo *Sublime Text*

Menurut Ardhana (dikutip Sa'ad, 2020:39), "*Sublime Text* adalah salah satu program yang digunakan untuk melakukan *editor*, seperti *HTML*, *PHP*, *CSS*."

Menurut Harani dan Hasanah (2020:84), menyatakan bahwa "Aplikasi *Sublime Text* merupakan sebuah aplikasi editor digunakan pada kode dan teks yang dapat berfungsi di berbagai platform *operating system* (OS) dengan menggunakan teknologi *Phyton API*".

Sublime Text menjunjung berbagai jenis bahasa pemrograman dan dapat menyuguhkan fitur *syntax highlight* hampir pada semua bahasa pemrograman yang didukung maupun dikembangkan oleh komunitas seperti; C, C++, SQL, TCL, Textile, CSS, D, LaTeX, Lisp, Lua, PHP, Python, XML, Dylan, Erlang, Java, JavaScript, OCaml, Perl, C#, HTML, Markdown, MATLAB, Groovy, Ruby, R and Haskell.

Berikut karakteristik yang diunggulkan dari aplikasi *Sublime Text* :

1. *Go to Anything*

Go to anything digunakan untuk membuka file yang menarik dengan satu project file yang sedang kita kerjakan/lakukan pada *sublime* lalu dengan cara menekan CTRL+P maka kita akan dapat mencari file apa yang akan kita buka dengan menuliskan nama file nya, lalu kita dapat mencari baris kata dengan keyword # atau @ untuk melompat ke simbol setelah menekan CTRL+P.

2. *Multiple Selections*

Multiple selections digunakan untuk membuat perubahan *code* pada saat yang sama dalam beberapa baris yang berbeda. Dapat dicoba dengan cara menekan



CTRL+L, atau dapat juga dengan mengganti kata yang sama dalam baris berbeda dengan memblok kata yang akan diganti kemudian tekan CTRL+D.

3. *Command Pallete*

Command pallete digunakan untuk fungsi yang satu ini jarang digunakan sebenarnya namun fungsi ini banyak sekali hal yang bisa dilakukan seperti menutup semua file, *convert file: lower case*, *remove tag*, dan masih banyak lagi. Dengan menekan CTRL+SHIFT+P.

4. *Distraction Free Mode*

Distraction Free Mode digunakan untuk merubah tampilan menjadi layar penuh, dibutuhkan ketika *user* akan fokus pada kerjaan yang dikerjakan maka dengan fungsi ini akan sangat membantu dengan menekan SHIFT+F11.

5. *Split Editing*

Split editing untuk mendapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan *editing* perpecahan. Melakukan *editing* di sisi file dengan sisi, atau mengedit dua lokasi di satu file.

6. *Instant Project Switch*

Mengambil seluruh file yang dimasukkan kedalam project pada aplikasi ini terhubung dengan fitur *Go to anything* untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam project lainnya dengan cepat.

7. *Plugin API*

Aplikasi ini memiliki plugin API berbasis *Python* sehingga membuat aplikasi ini sangat tangguh.

8. *Customize Anything*

Aplikasi sublime ini memiliki dan memberikan *user/pengguna* fleksibilitas dalam hal pengaturan/pengarahan fungsional dalam aplikasi ini.

9. *Cross Platform*

Aplikasi ini bisa berjalan hampir pada semua *operating system modern* seperti *Windows*, *Linux based operating system*, dan *OS X*.

Sumber : Harani dan Hasanah (2020:84-87).