



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Perangkat Lunak

Menurut Kadir (2017:2), “Perangkat Lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

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”.

2.1.2 Komputer

Menurut Kadir (2017:2), “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Fauzi (2018:1) merupakan mengatakan bahwa, “Komputer adalah suatu peralatan elektronik yang dapat menerima input, mengolah input (processing), memberika informasi dengan menggunakan suatu program yang tersimpan di memori komputer dan dapat menyimpan program dari hasil pengolahan yang bekerja secara otomatis”.

2.1.3 Data

Menurut Pamungkas (2017:1), “Data Merupakan nilai yang mempresentasikan deskripsi dari suatu objek atau kejadian”.

Menurut Pane, dkk, (2020:7), "Data adalah kumpulan 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.



2.1.4 Basis Data

Menurut Fitri (2020:1), “Basis Data adalah kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer”.

Pamungkas (2017:2) mengemukakan bahwa, “Basis Data merupakan Pengertian suatu 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”.

2.1.5 Metode Pengembangan Sistem

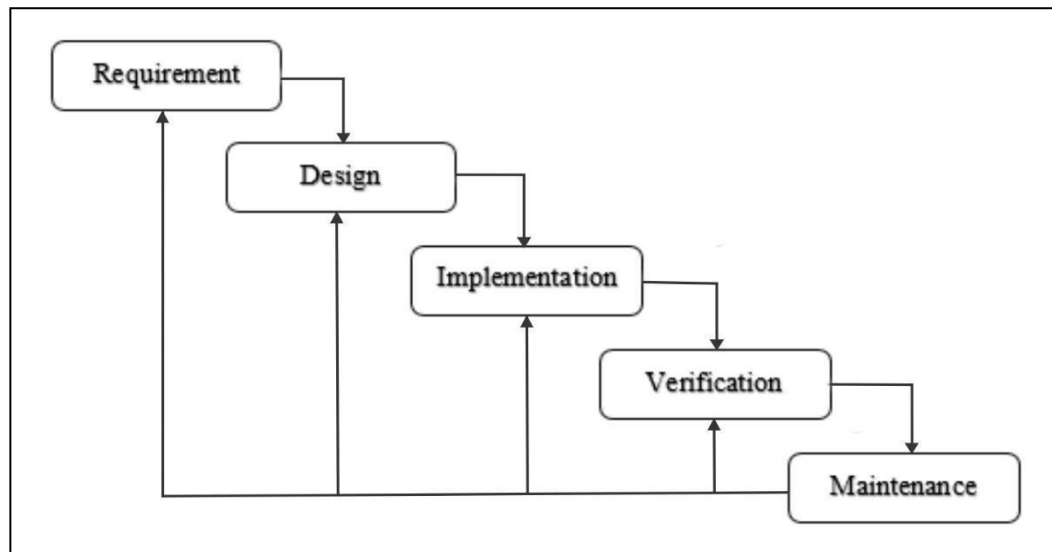
Penelitian Laporan Akhir ini menggunakan metode pengembangan sistem Waterfall. Menurut Pressman (dikutip Sanubari, dkk, 2020:41), menyatakan bahwa "Metode air terjun atau yang sering disebut metode waterfall sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan 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.

2.2 Teori Khusus

2.2.1 Kamus Data

Menurut Rianto (2021:133), “Kamus data adalah kumpulan informasi terpusat terkait data”.

Menurut Rosa dan Shalahuddin (2018:73-74), “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Kamus data biasanya berisi :

- a. Nama-nama dari data
- b. Digunakan pada proses-proses yang terkait data
- c. Deskripsi data
- d. Informasi tambahan seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

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).

2.2.2 *Unified Modeling Language* (UML)

2.2.2.1 Pengertian *Unified Modeling Language* (UML)



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

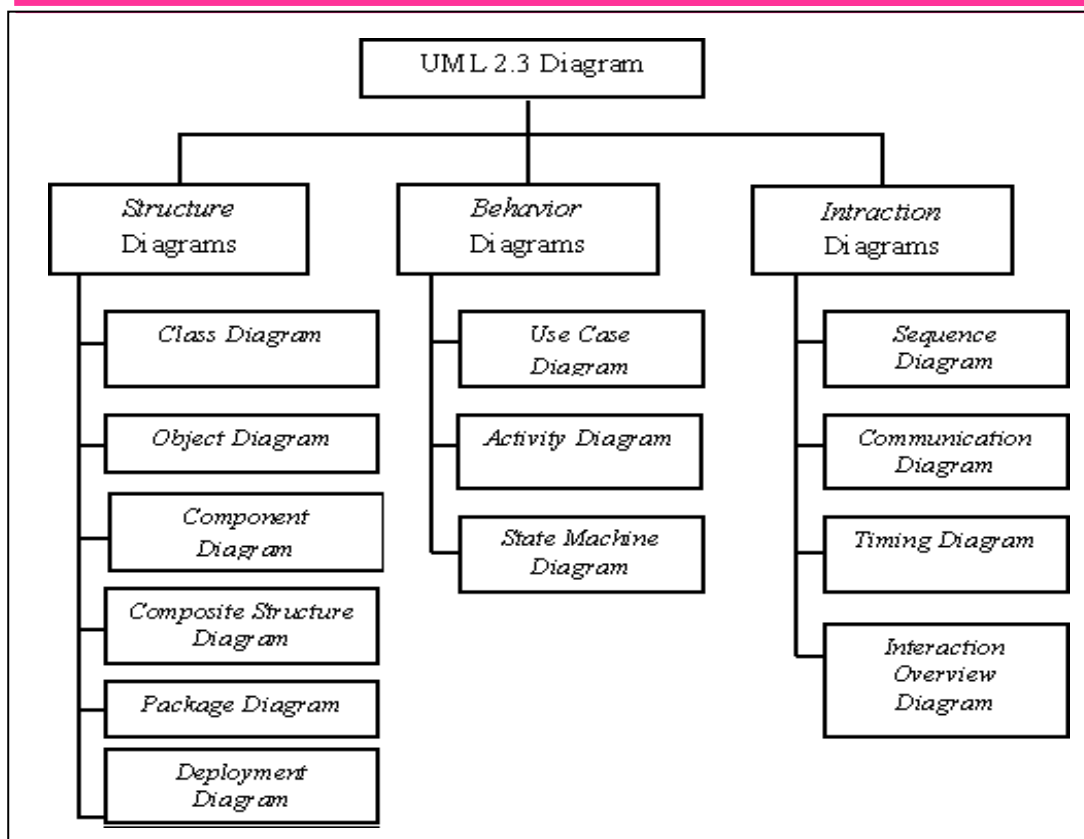
Gambar 2.2 Logo UML

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

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*".

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."

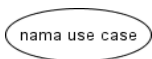
Menurut Setyawati, dkk (2021:13), "*Use case diagram* merupakan konsep informasi dengan menggunakan fungsi apa saja yang ada di dalam sistem informasi & siapa saja *actor* yang terlibat."

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

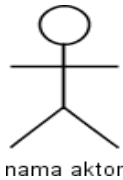

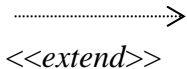

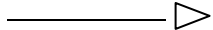
- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas 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>  nama aktor	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).



Lanjutan Tabel 2.2 Simbol-Simbol Use Case Diagram

No	Simbol	Deskripsi
6.	<p>menggunakan</p> <p>/include / uses</p> <p>.....></p> <p><<include>></p> <p><<uses>></p>	<p>relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <p>a. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:</p> <pre> graph TD login((login)) -- "<<include>>" --> validasi_username((validasi username)) </pre> <p>b. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p> <pre> graph TD ubah_data((ubah data)) -- "<<include>>" --> validasi_user((validasi user)) </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Sukamto dan Shalahuddin (2016:156-158).

2.2.3.2 Pengertian Class Diagram

Menurut Sukamto dan Shalahuddin (2018:141-142), "*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.



Haqi (2019:32), "Class diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi yang terdapat pada sistem tersebut". 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).


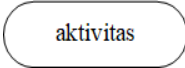
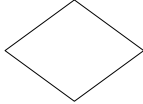



2.2.3.2 Pengertian Activity Diagram

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".


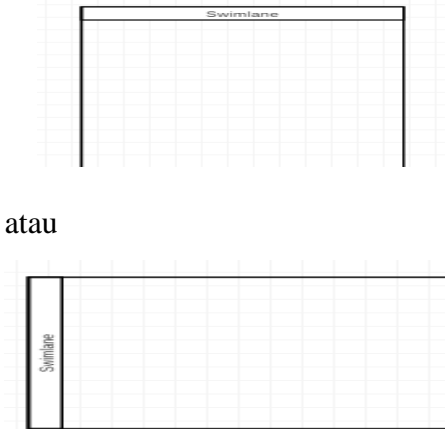
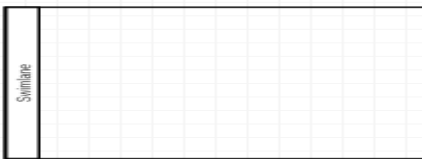
Menurut Haqi (2019:31), "*Activity diagram* atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem". Simbol-simbol pada *Activity Diagram* adalah sebagai berikut :

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.



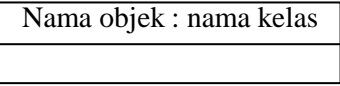
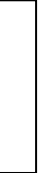
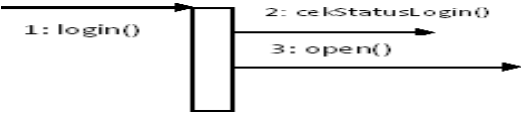
Sumber : Sukamto dan Shalahuddin (2016:162-163).

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*."

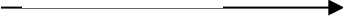

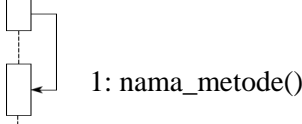
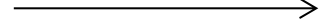
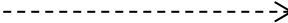
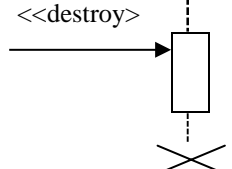
Menurut Haqi (2019:31), "*Sequence diagram* yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence diagram* juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*." Simbol-simbol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


Tabel 2.5 Simbol-Simbol *Sequence* Diagram

No	Simbol	Deskripsi
1.	<p><i>Actor</i></p>  <p>nama aktor</p> <p>Atau</p> <p>Nama aktor</p> <p>tanpa waktu aktif</p>	<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>.</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>menyatakan kehidupan suatu objek.</p>
3.	<p>Objek</p> 	<p>menyatakan objek yang berinteraksi pesan.</p>
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> <<create>> 	menyatakan objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 1: nama_metode() 	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> 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 1: keluaran 	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> <<destroy>> 	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

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".

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*".

2.3.3 Pengertian Cuti

Menurut Pujiasri 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".

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".



2.3.4 Pengertian Pegawai

Menurut Soedaryono Dikutip oleh Pujiarsih dalam Buku Otomatisasi Tata Kelola Kepegawaian (2018:4) mendefinisikan bahwa "Pegawai adalah seorang yang melakukan penghidupan dengan bekerja dalam kesatuan organisasi baik kesatuan pemerintah maupun kesatuan swasta".

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".

2.3.5 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".

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 halamandengan halaman yang lain yang sering disebut sebagai *hyperlink*".

2.3.6 Pengertian Aplikasi Permohonan Cuti Pegawai Berbasis Web Pada UPTB Pengelolaan Pendapatan Daerah Kota Pagar Alam

Aplikasi Permohonan Cuti Pegawai Berbasis *Web* adalah aplikasi untuk mengajukan permohonan cuti pegawai secara online. Dengan adanya aplikasi ini, pegawai dapat mengajukan proses permohonan cuti pegawai menggunakan koneksi internet agar lebih efektif dan efisien.

Aplikasi Permohonan Cuti Pegawai Berbasis *Web* adalah aplikasi untuk mengajukan permohonan cuti pegawai secara online. Dengan adanya aplikasi ini, pegawai dapat mengajukan proses permohonan cuti pegawai menggunakan koneksi internet agar lebih efektif dan efisien.



2.4 Teori Program

2.4.1 Pengertian HTML



Sumber : wikipedia.org (2021)

Gambar 2.4 Logo HTML

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".

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".



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*".

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."



2.4.5 Pengertian CodeIgniter



Sumber : shasaninfotech.com (2020)

Gambar 2.8 Logo CodeIgniter

Menurut Putra (2021:13) “CodeIgniter merupakan suatu kerangka kerja (*framework*) pengembangan aplikasi yang digunakan untuk membangun aplikasi berbasis web menggunakan bahasa pemrograman PHP (*Perl Hypertext Processor*)”.

Menurut Harani dan Sunandhar (2020:43), mendefinisikan bahwa "CodeIgniter merupakan sebuah toolkit yang ditunjukkan orang yang ingin membangun aplikasi web dalam bahasa pemrograman PHP”.

2.4.6 Pengertian XAMPP



Sumber : mrenesia.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”.



2.4.7 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".

2.4.8 Pengertian Visual Studio Code



Sumber : code.visualstudio.com (2016)

Gambar 2.11 Logo Visual Studio Code

Menurut Salamah (2021:1), menyatakan bahwa "Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh microsoft untuk sistem informasi multiflatform, artinya tersedia juga untuk versi Linux, Mac dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java dst)".