



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Harmayani, dkk. (2021:2) “Komputer adalah alat elektronik yang terdiri dari rangkaian berbagai komponen yang saling terhubung sehingga membentuk suatu sistem kerja. Sistem di dalam komputer tersebut dapat melakukan pekerjaan secara otomatis berdasarkan program yang diperintahkan kepadanya sehingga mampu menghasilkan informasi berdasarkan data dan program yang ada”.

Menurut Situmorang & Maudiarti (2020:4) “Komputer secara sederhana adalah sebuah alat untuk melakukan proses perhitungan aritmatika, sedangkan secara umum adalah peralatan elektronika yang berfungsi sebagai peng-*input* data kemudian mengolahnya dan memberikan keluaran informasi dalam bentuk teks, gambar, suara, maupun video”.

Menurut Robert H. Blissmer dalam Medcoms (2019:3) “Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, seperti menerima dan memproses input, menyimpan perintah - perintah dan menyediakan output dalam bentuk informasi”.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa Pengertian Komputer adalah suatu alat elektronik yang mengelola data menjadi suatu informasi yang berguna dan juga dapat melakukan perhitungan sehingga dapat mempermudah pekerjaan manusia di berbagai bidang.

2.1.2 Pengertian Perangkat Lunak

Menurut Wibawanto (2018:193) menyatakan bahwa perangkat lunak merupakan program-program komputer yang digunakan untuk menjalankan pekerjaan sesuai dengan yang dikehendaki dan ditulis dengan bahasa khusus yang dimengerti oleh komputer.



Menurut Harmayani, dkk. (2021:7) “*Software* dalam komputer adalah komponen penting karena berperan menjalankan segala perintah yang masuk ke *hardware*. *Software* dapat diartikan juga sebagai suatu kumpulan data elektronik yang tersimpan dan diatur oleh komputer, bisa berupa program ataupun koneksi untuk menjalankan berbagai macam instruksi perintah”.

Harmayanti, dkk (2021:43) mengatakan bahwa Perangkat lunak merupakan kumpulan data elektronik, yang tersimpan dan kemudian dikendalikan oleh perangkat keras komputer. Data elektronik tersebut berupa instruksi atau program yang nantinya akan menjalankan perintah khusus.

Berdasarkan pengertian diatas dapat disimpulkan bahwa perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model *desain*, dan cara penggunaan untuk membantu proses sistem informasi.

2.1.3 Pengertian Data

Menurut Siswandari (dalam Makbul, 2021:3) “Data dapat berarti fakta dari suatu objek yang diamati yang dapat berupa angka-angka dan kata-kata. Sedangkan jika dipandang dari sisi statistika”.

"Data merupakan bentuk yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model yang nantinya akan dikelompokkan dan diproses untuk menghasilkan informasi" (Kristanto, 2018:8).

Menurut Rochman, dkk. (2019:2) “Data adalah suatu bahan mentah yang kelak dapat diolah lebih lanjut untuk menjadi suatu yang lebih bermakna. Data inilah yang nantinya akan disimpan dalam *database*”.

Jadi, berdasarkan pengertian di atas data adalah fakta atau informasi mentah yang belum dapat memberikan manfaat bagi penerimanya.

2.1.4 Pengertian Basis Data

Menurut Jayanti dan Sumiarti (2018:2) menyatakan bahwa secara konsep basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu



berkas (file) yang saling terhubung (*relation*) dengan tata cara tertentu untuk membentuk suatu data baru atau sebuah informasi.

Menurut Fathansyah (2018:2), menjelaskan bahwa basis data (*database*) merupakan kumpulan dari kelompok data (arsip) dan saling berhubungan serta diorganisasikan sedemikian rupa supaya nanti dapat digunakan lagi dengan cepat dan mudah.

Rosa dan Shalahuddin (2018:43), "Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia saat dibutuhkan".

Berdasarkan definisi di atas, basis data adalah kumpulan data yang saling berhubungan dan disimpan secara terkomputerisasi yang bertujuan agar informasi tersedia saat dibutuhkan.

2.1.5 Pengertian Sistem

Menurut Yunita & Devitra (2017) mengatakan bahwa "Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu".

Sitinjak. dkk (dalam Hutahean 2020) mengatakan bahwa "Sistem merupakan suatu jaringan kerja yang terdiri dari prosedur-prosedur yang saling berhubungan dan bersama-sama melakukan kegiatan atau melakukan sasaran tertentu".

Menurut Prehanto (2020:3) mengatakan bahwa "Sistem adalah bagian-bagian komponen yang dikumpulkan yang memiliki hubungan satu sama lain baik fisik maupun non-fisik yang bersama-sama dalam bekerja demi tujuan yang dituju secara harmonis".

Berdasarkan pengertian sistem tersebut dapat diambil suatu pendapat bahwa pengertian sistem dalam komputer adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

2.1.6 Pengertian Informasi

Informasi dapat didefinisikan sebagai data yang telah disusun atau diproses sedemikian rupa sehingga mempunyai makna dan bermanfaat kepada seseorang yang akan menggunakannya untuk mengambil suatu keputusan.

Menurut Prehanto (dalam Sutabri 2020) mengatakan bahwa "Informasi merupakan pengelolaan data yang diinterpretasikan maupun diklasifikasi yang dipakai dalam proses untuk mengambil keputusan".

"Informasi dapat didefinisikan sebagai data yang telah terorganisasi dan memiliki kegunaan ataupun manfaat". (Rahmasari dalam Susanto, 2019).

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan yang nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Informasi dapat diperoleh dari sistem informasi dari penjelasan di atas dapat di ambil kesimpulan bahwa informasi adalah hasil dari pengolahan data dalam suatu bentuk yang mempunyai arti bagi si penerima dan mempunyai nilai yang nyata, sehingga dapat dipakai sebagai dasar mengambil keputusan untuk sekarang maupun yang akan mendatang.

2.1.7 Pengertian Sistem Informasi

Menurut Yunita & Devitra (2017) mengatakan bahwa "Sistem informasi merupakan satuan komponen yang saling berhubungan yang mengumpulkan (atau mendapatkan kembali), memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi".

Menurut Yunaeti dan Irvani (dalam Sitinjak 2020) mengatakan bahwa "Sistem Informasi meruokan kombinasi teratur dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah serta menyebarkan informasi dalam sebuah organisasi".

Menurut Rahmasari (2019) mengatakan bahwa "Sistem informasi merupakan kumpulan dari sub sistem yang bekerja sama dalam mengolah data untuk mencapai tujuantertentu dan menghasilkan informasi yang bermanfaat".



2.2 Teori Judul

2.2.1 Aplikasi

Menurut Kamus Besar Bahasa Indonesia (2018), “Aplikasi adalah program komputer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu.”

"Aplikasi merupakan suatu program komputer yang digunakan untuk membantu pengguna dalam membuat suatu tugas tertentu" (Alfaridzi, 2019:10).

Menurut Eviana *et al.* (2017:22), "Aplikasi adalah kumpulan perintah program yang dapat melakukan pekerjaan-pekerjaan tertentu dan melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju."

2.2.2 Penerima Beasiswa

Penerima adalah proses, cara, perbuatan menerima, penyambutan. Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan. Pemberian beasiswa dapat dikategorikan pada pemberian cuma- cuma ataupun pemberian dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan. Lama ikatan dinas ini berbeda-beda, tergantung pada lembaga yang memberikan beasiswa tersebut.

Jadi, penerima beasiswa adalah proses pemberian bantuan pendidikan kepada seseorang baik berupa uang maupun jaminan pendidikan demi keberlangsungan pendidikan yang ditempuh.

2.2.3 Dhuafa

Kaum dhuafa terdiri dari orang-orang yang terlantar, fakir miskin, anak-anak yatim dan orang cacat. Kaum dhuafa ialah orang-orang yang menderita hidupnya secara sistematis.



2.2.4 Berprestasi

Arti berprestasi adalah mempunyai prestasi dalam suatu hal atau dalam suatu bidang yang merupakan hasil kerja atau hasil usaha. Definisi berprestasi adalah tindakan seseorang yang berasal dari dorongan dari dalam dirinya atau dari luar dirinya untuk melakukan sesuatu dengan hasil terbaik untuk memperoleh predikat unggul.

2.2.5 Website

Menurut Awal (2019:1) “*Website* adalah halaman informasi yang disediakan melalui jaringan internet sehingga bisa di akses diseluruh dunia Selama masih tetkoneksi dengan intrenet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi.”

Menurut Elgamar (dalam Novria, dkk., 2022) *Website* merupakan sebuah media yang memiliki banyak halaman yang saling terhubung (*hyperlink*), berfungsi memberikan informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya. Karakteristik utama dari *website* adalah halaman-halaman yang saling terhubung dan dilengkapi dengan domain sebagai alamat (url) atau *World Wide Web* (www) dan juga *hosting* sebagai media penyimpanan banyak data.

Dari beberapa pengertian di atas, penulis menyimpulkan bahwa *website* adalah halaman yang saling terhubung dan berisi informasi data digital yang berfungsi memberikan informasi berupa teks, gambar, video, suara, atau berbagai bentuk media lainnya.

2.2.6 Yayasan

Yayasan (Inggris: *foundation*) adalah suatu badan hukum yang mempunyai maksud dan tujuan bersifat sosial, keagamaan dan kemanusiaan, didirikan dengan memperhatikan persyaratan formal yang ditentukan dalam undang-undang.



2.2.7 Amil

Amil dalam zakat adalah semua pihak yang bertindak mengerjakan yang berkaitan dengan pengumpulan, penyimpanan, penjagaan, pencatatan, dan penyaluran atau distribusi harta zakat. Mereka diangkat oleh pemerintah dan memperoleh izin darinya atau dipilih oleh instansi pemerintah yang berwenang atau oleh masyarakat Islam untuk memungut dan membagikan serta tugas lain yang berhubungan dengan zakat, seperti penyadaran atau penyuluhan masyarakat tentang hukum zakat, menerangkan sifat-sifat pemilik harta yang terkena kewajiban membayar zakat dan mereka yang menjadi mustahiq, mengalihkan, menyimpan dan menjaga serta menginvestasikan harta zakat sesuai dengan ketentuan yang ditetapkan dalam rekomendasi pertama Seminar Masalah Zakat Kontemporer Internasional ke-3, di Kuwait. Lembaga-lembaga dan panitia-panitiapengurus zakat yang ada pada zaman sekarang ini adalah bentuk kontemporer bagi lembaga yang berwenang mengurus zakat yang ditetapkan dalam syari'at Islam. Oleh karena itu, petugas (amil) yang bekerja di lembaga tersebut harus memenuhi syarat-syarat yang ditetapkan.

2.2.8 Zakat

Zakat adalah jumlah harta tertentu yang wajib dikeluarkan oleh orang yang beragama Islam dan diberikan kepada golongan yang berhak menerimanya (fakir miskin dan sebagainya) menurut ketentuan yang telah ditetapkan oleh syarak. Zakat merupakan rukun ketiga dari Rukun Islam. Zakat dari segi prakteknya adalah kegiatan bagi-bagi yang diwajibkan bagi umat islam. Zakat berbeda dengan gratifikasi. Gratifikasi adalah kegiatan bagi-bagi yang tidak diperkenankan oleh negara atau ketentuan pemerintah.

2.2.9 Pengertian Aplikasi Pendaftaran Registrasi Ulang Penerima Beasiswa Dhuafa Berprestasi Yayasan YAZRI Berbasis Website menggunakan Framework Codeigniter.



Aplikasi Pendaftaran Registrasi Ulang Penerima Beasiswa Dhuafa Berprestasi Yayasan Yazri Berbasis *Website* Menggunakan *Framework Codeigniter* adalah sebuah aplikasi yang digunakan untuk melakukan registrasi ulang berbasis *website*, yang bertujuan agar dapat mempermudah pihak Yayasan dan memperlancar proses anak asuh Beasiswa Dhuafa Berprestasi Yayasan Yazri.

2.2.10 Framework

Framework merupakan kumpulan fungsi–fungsi/prosedur–prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal (Supardi dan Hermawan, 2018).

Sedangkan Menurut Naista (2017) mengemukakan bahwa *Framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang bersifat kompleks.

2.2.11 CodeIgniter

CodeIgniter merupakan sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi *web*. Tujuan utama pengembangan *codeigniter* adalah untuk membantu *developer* dalam pengerjaan aplikasi yang lebih cepat dari pada menulis semua *code* dari awal dan *codeigniter* merupakan salah satu *framework* tercepat saat ini (Novianto, 2017).

Menurut Yanuarsyah (dalam Suharman 2021), “*Codeigniter* merupakan kerangka kerja untuk mengembangkan aplikasi berbasis web menggunakan bahasa pemrograman PHP. *Codeigniter* memungkinkan mengembangkan sebuah aplikasi menjadi lebih cepat karena fasilitas-fasilitas untuk mengembangkan aplikasi sudah disediakan dan dokumentasi yang tersedia lengkap”.

Menurut (Anton Subagia, 2018) sebagai *web framework* populer yang menggunakan bahasa pemrograman PHP, *CodeIgniter* mempunyai beberapa keunggulan yaitu:



1. *Free*, karena berada di bawah lisensi *open source*, kita dapat melakukan apapun dengan *CodeIgniter*. Lisensi lengkapnya dapat dilihat di halaman dokumentasi.
2. *Light weight*, sistem inti *CodeIgniter* memerlukan *library* yang sedikit. Sangat berbeda dengan *framework* lainnya yang membutuhkan banyak sumber daya tambahan. *Library* tambahan akan digunakan ketika *request* secara dinamis, membuat *system* yang dibangun menjadi efisien dan cukup cepat.
3. *Fast*, menurut dokumentasi, performa yang dimiliki *CodeIgniter* terbukti cepat setelah dibandingkan dengan *framework* lainnya.
4. Menggunakan kaidah MVC, dengan menggunakan *Model-View-Controller*, kita dapat memisahkan bagian *logic* dan *presentation* dari aplikasi yang kita bangun. Hal ini tentu sangat cocok dan bagus untuk proyek yang memfokuskan, desainer focus pada template file dan programmer focus pada pembangunan logic dari aplikasi yang dibangun.
5. Menghasilkan URL yang bersih, URL yang dihasilkan oleh *CodeIgniter* bersih dan ramah terhadap *search engine*. *CodeIgniter* menggunakan pendekatan *segment-based* dibandingkan dengan *query string* yang biasa digunakan oleh programmer yang tidak menggunakan *web framework*.
6. *Packs a Punch*, *CodeIgniter* hadir dengan berbagai *library* yang akan membantu tugas-tugas di pengembangan *web* yang sudah umum dan sering dilakukan seperti mengakses *database*, mengirim email, validasi data dari *form*, mengelola *session*, memanipulasi gambar, dan masih banyak lagi.
7. *Extensible*, kita dapat menambahkan *library* atau *helper* yang kita ciptakan sendiri ke dalam *CodeIgniter*. Selain itu kita dapat juga menambahkan fitur lewat *class extension* atau *system hooks*.
8. *Thoroughly Documented*, hampir semua fitur, *library* dan *helper* yang ada di *CodeIgniter* telah terdokumentasi dengan lengkap dan tersusun dengan baik.

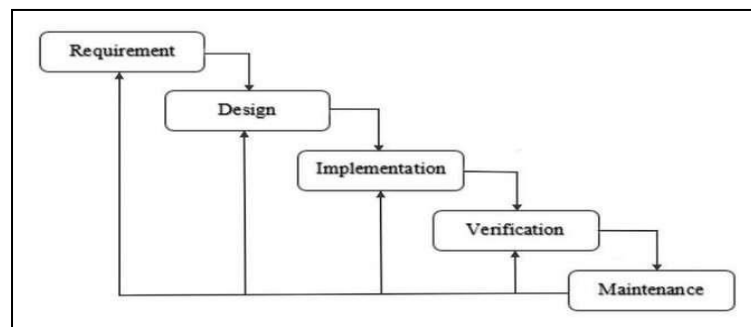
2.3 Teori Khusus

2.3.1 Metode Pengembangan Aplikasi

Metode pengembangan sistem yang dilakukan dalam penelitian ini adalah Metode *Waterfall* yang mana metode tersebut menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut (Casro, dkk., 2020).

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."

Sukamto dan Shalahuddin (2018:28), "Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). "Metode air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis desain pengodean, pengujian, dan tahap pendukung (*support*).



Sumber: Sanubari, dkk (2020:41)

Gambar 2.1 Tahapan-Tahapan Metode Waterfall

Adapun penjelasan untuk masing-masing tahapan adalah sebagai berikut:

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak ini perlu didokumentasikan.

b. *Desain*

Desain perangkat lunak adalah proses multi langkah yang fokus pada *desain* pembuatan program sistem termasuk struktur data, arsitektur sistem, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan sistem dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. *Desain* perangkat lunak ini yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain. Desain perangkat lunak ini juga perlu didokumentasikan.

d. Pengujian

Pengujian fokus pada perangkat lunak dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung dan pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk perangkat lunak yang baru.

2.3.2 Pengertian *Unified Modeling Language* (UML)

Sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek yaitu *Unified Modelling Language* (UML). UML muncul karena adanya kebutuhan



pemodelan visual untuk menspesifikasikan , menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

Menurut Sukamto dan Shalahuddin (2018:133), mendefinisikan bahwa “UML merupakan sebuah standar Bahasa yang digunakan untuk menganalisis dan merancang serta menggambarkan arsitektur program dalam pemrograman *object oriented*”.

Pada UML dikelompokkan dalam 3 macam kategori sebagai berikut:

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 padasebuah sistem.
3. *Interactions diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun sistem interaksiantar subsistem pada suatu sistem.

2.3.3 Jenis-Jenis Diagram UML

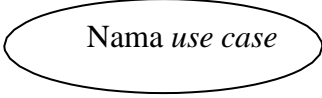
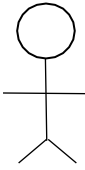

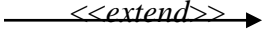
2.3.3.1 Pengertian Use Case Diagram

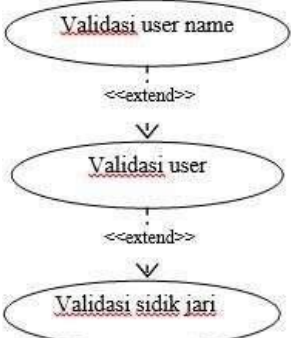
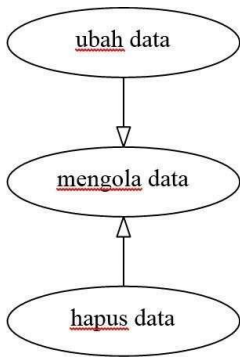
Sukamto dan Shaluddin (2018:155), menjelaskan tentang *use case diagram* sebagai berikut:

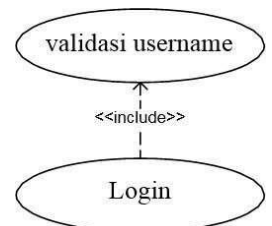
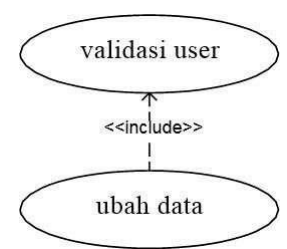
“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem”.

Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2. 1 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.	Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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 frame nama aktor.
3.	Assosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan 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 yang sama dengan <i>use case</i> yang ditambahkan, misalnya

		 <pre>graph TD; A([Validasi user name]) -- <<extend>> B([Validasi user]); B -- <<extend>> C([Validasi sidik jari]);</pre> <p>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.</p>
5.	Generalisasi / <i>generalization</i> →	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <pre>graph TD; A([hapus data]) --> B([mengola data]); B --> C([ubah data]);</pre> <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>

6.	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>→</p> <p><<include>></p> <p><<uses>></p> <p>→</p>	<p>Relasi <i>use case</i> 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.</p> <p>ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <ul style="list-style-type: none">• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tamb ahan dijalankan, missal pada kasus berikut :  <pre>graph BT; Login((Login)) -.-> <<include>> validasi_username((validasi username));</pre> <ul style="list-style-type: none">• <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankansebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut :  <pre>graph BT; 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 (2018:156-158)

Ada dua hal utama pada *use case* yaitu:

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


2.3.3.2 Pengertian *Activity Diagram*

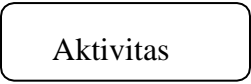
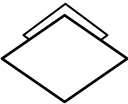


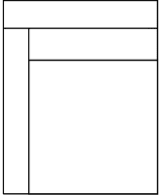
Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut:

“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”.

Adapun simbol- simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2. 2 Simbol-Simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Sukamto dan Shalahuddin (2018:162-163)

2.3.3.3 Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2018:141), menjelaskan tentang *class diagram* sebagai berikut:

“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron”.

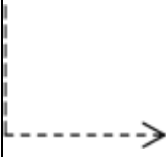
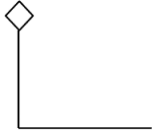


Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

Tabel 2. 3 Simbol-Simbol *Class Diagram*

No	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	Antarmuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/ <i>association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah/ <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus).



6.	Kebergantungan/ <i>dependens</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber : Sukamto dan Shalahuddin (2018:146-147)

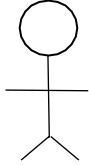
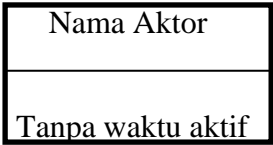

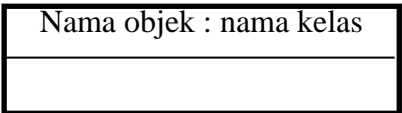

2.3.3.4 Pengertian *Sequence Diagram*

“*Sequence Diagram* 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 beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*”. (Sukamto dan Shalahuddin, 2018:165).

Dapat penulis simpulkan bahwa *Sequence Diagram* adalah penggambaran skenario dari sebuah objek yang ada pada *use case* yang meliputi rangkaian Langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada *Sequence Diagram*:

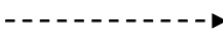
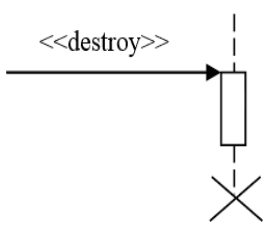
Tabel 2. 4 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Nama Aktor</p> <p>Atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</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, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>



		<p>Maka cek status <i>login()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i> aktor tidak memiliki waktu aktif.</p>
5.	Pesan tipe <i>create</i> 	Menyatakan suatu 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, <p>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.</p>
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/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> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i> .

Sumber: : Sukamto dan Shalahuddin (2018:165-167)

2.3.4 Kamus Data

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

Sukamto 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 menulis)."

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:

Gambar 2. 2 Simbol-Simbol Kamus Data

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

(Sumber : Sukamto dan Shaluddin (2018:74))

2.3.5 Pengertian *Blackbox*

Teknik pengujian yang penulis gunakan adalah *Black-Box Testing*. Menurut Sukamto (2016:275) *Black Box Testing* yaitu menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau model secara offline dan online melalui publik, kemudian diamati apakah hasil dari unit itu sesuai dengan proses yang diinginkan.

Menurut Rosa (2016:275) *Black Box Testing* yaitu menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Menurut Sutinah et al. (2019) "*Blackbox* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak apakah sesuai dengan fungsinya".

2.4 Teori Program

2.4.1 Pengertian Database



Gambar 2. 3 Logo Database

Menurut Fathansyah (2018:2), menjelaskan bahwa basis data (*database*) merupakan kumpulan dari kelompok data (arsip) dan saling berhubungan serta diorganisasikan sedemikian rupa supaya nanti dapat digunakan lagi dengan cepat dan mudah.

Menurut Abdulloh (2018:103), “*Database* atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi”.

Jadi, dapat ditarik kesimpulan bahwa basis data (*database*) adalah sekumpulan informasi yang disimpan di dalam komputer secara sistematis dan membuat informasi tersedia saat dibutuhkan agar dapat diakses dengan mudah dan cepat.

2.4.2 Pengertian PHP



Gambar 2. 4 Logo PHP



Menurut Abdulloh (2018:127), "PHP merupakan kependekan dari PHP *Hypertext Preprocessor* yaitu bahasa pemrograman *web* yang dapat disisipkan dalam skrip HTML dan bekerja di sisi *server*. Tujuan dari bahasa ini adalah membantu para pengembangan *web* untuk membuat *web* dinamis dengan cepat".

Menurut Kadir (2018:358), "PHP merupakan bahasa pemrograman skrip yang diletakkan dalam *server* yang biasa digunakan untuk membuat aplikasi *web* yang bersifat dinamis".

Risawandi (2019:1), Pada awalnya PHP digunakan sebagai bahasa pemrograman untuk sebuah server-side HTML-embedded dengan nama Personal Home Pages. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari *web*.

Dapat disimpulkan bahwa, PHP merupakan bahasa pemrograman yang dapat disisipkan dalam membuat suatu aplikasi *web*.

2.4.3 Pengertian HTML



Gambar 2. 5 Logo HTML

Abdulloh (2018:7), "HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar *web* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*". Menurut (Endra & Aprilita, 2018), "HTML atau *Hypertext Markup Language* merupakan salah satu bahasa yang biasa digunakan oleh pengguna dalam membuat tampilan yang digunakan oleh *web application*".

Menurut Hadi (2018:1), “HTML (*Hyper Text Markup Language*) adalah sebuah bahasa formatting yang digunakan untuk membuat sebuah halaman *website*. Di dalam dunia pemrograman berbasis *website* (*Web Programming*), HTML menjadi pondasi dasar pada halaman *website*”.

Berdasarkan pendapat diatas, (*HyperText Markup Language*) HTML adalah bahasa markup berupa tag-tag yang digunakan untuk membuat halaman *web* dan menampilkan informasi didalam sebuah *web* internet.

2.4.4 Pengertian CSS



Gambar 2. 6 Logo CSS

Menurut Abdulloh (2019:45) “CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen *web* yang berfungsi mengatur elemen HTML dengan berbagai *property* yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan”.

Menurut Lewenusa (2020:1) “CSS (*Cascading Style Sheet*) secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan tag HTML, seperti *font*, *color*, *text* dan tabel menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan”.

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa CSS adalah sebuah Bahasa sederhana yang digunakan untuk menambahkan gaya/ *styling* (misalnya *font*, warna, spasi dan lain-lain) ke sebuah halaman *web*.



2.4.5 Pengertian *Bootstrap*

"*Bootstrap* merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada. *Bootstrap* memungkinkan desain sebuah *web* menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik" (Abdulloh, 2018:261).

Menurut Sulistiono (2018:17), "*Bootstrap* adalah sebuah pustaka *open source* yang merupakan *framework* CSS dan *Javascript* untuk membuat *website* yang responsif."

2.4.6 Pengertian *MySQL*



Gambar 2. 7 Logo MySQL

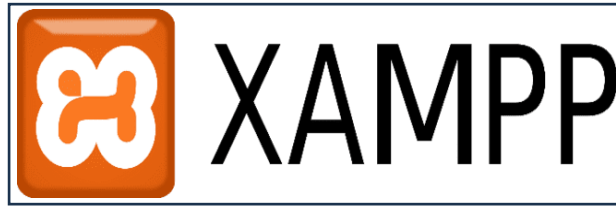
Menurut Teguh dan Wagiyati (2022), "*MySQL* merupakan salah satu jenis *database* yang bersifat *open source*."

Menurut Rusmawan (2019:97), "*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Manajemen System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia."

Menurut Mundzir (2018:217) "*MySQL* adalah sistem manajemen *database* SQL yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem *database MySQL* mampu mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan SQL *database management system* (DBMS)".

Menurut Madcoms (dalam Ayu dan Permatasari, 2018:21), "*MySQL* adalah sistem manajemen *Database SQL* yang bersifat *open source* dan paling populer saat ini. Sistem *Database MySQL* mendukung beberapa fitur seperti *multithreaded*, *multiuser* dan SQL *Database managemen system* (DBMS)."

2.4.7 Pengertian XAMPP



Gambar 2. 8 Logo XAMPP

Menurut Putra (2020:29) XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan komplikasi dari beberapa program yang berfungsi sebagai *server* yang berdiri sendiri (*local host*), yang terdiri atas *Apache*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Hidayatullah dan Kawistara (2017:125) dalam bukunya mengatakan bahwa, "*XAMPP support* untuk banyak sistem operasi seperti *Windows*, *Linux*, *Mac OS* dan *Solaris* sehingga tidak terdapat masalah ketika melakukan perpindahan sistem operasi."

Menurut Madcoms (dalam Ayu dan Permatasari, 2018:19), "XAMPP adalah sebuah paket kumpulan *software* yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *PHP*, *Perl*, *Filezilla*, dan lain."

2.4.8 Pengertian Sublime Text



Gambar 2. 9 Logo Sublime Text

Menurut Syifani dan Ardiansyah (2018:25) "*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi Phyton API".

Sedangkan menurut Murya (2017:9), "*sublime* atau *sublime text* merupakan *text editor* sama halnya dengan *Notepad++* hanya saja fitur dan *user interface* yang lengkap yang membuat editor ini banyak dilirik oleh *kalangan programmer*."

2.4.9 Pengertian Framework CodeIgniter



Gambar 2. 10 Logo CodeIgniter

Menurut Sallaby & Kanedi (dalam Ridwan, dkk., 2022) mengatakan bahwa *CodeIgniter* adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*. *CodeIgniter* memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya. *CodeIgniter* bersifat *open source* dan menggunakan model basis MVC (*Model View Controller*), yang merupakan model konsep modern saat ini. Metode MVC (*Model View Controller*) terdapat tiga komponen menurut Yesputra, dkk. (dalam Ridwan, dkk., 2022) yaitu:

1. *Model*, mengelola basis data (*RDBMS*) seperti *MySQL* ataupun *Oracle RDMS*. *Model* berhubungan dengan *database* sehingga biasanya dalam model akan berisi class ataupun fungsi untuk membuat (*create*), melakukan pembaruan (*update*), menghapus data (*delete*), mencari data (*search*), dan mengambil data (*select*) pada *database*. Selain itu juga *model* akan berhubungan dengan perintah-perintah *query* sebagai tindak lanjut dari fungsi-fungsi di atas.
2. *View*, bagian *User Interface* atau bagian yang nantinya merupakan tampilan untuk *end-user*. *View* bisa berupa halaman *HTML*, *CSS*, *Javascript*, *JQuery* dan *AJAX*, karena metode yang dipakai merupakan MVC sehingga *view* tidak boleh terdapat pemrosesan data ataupun pengaksesan yang



berhubungan dengan *database*, sehingga *view* hanya menampilkan data-data hasil dari *Model* dan *Controller*.

3. *Controller*, penghubung antara *view* dan *model*, maksudnya ialah karena *model* tidak dapat berhubungan langsung dengan *view* ataupun sebaliknya, jadi *controller* inilah yang digunakan sebagai jembatan keduanya. Sehingga tugas *controller* ialah sebagai pemrosesan data atau *Alur Logic Program*, menyediakan variabel yang akan ditampilkan di *view*, pemanggilan *model* sehingga *model* dapat mengakses *database*, *error handling* validasi atau cek terhadap suatu input data.

Berdasarkan uraian di atas, dapat disimpulkan bahwa *Framework CodeIgniter* adalah sebuah *framework* yang menggunakan bahasa pemrograman PHP yang digunakan untuk mempermudah *programmer* dalam mengembangkan atau membuat suatu *web*. *CodeIgniter* menggunakan model basis MVC atau terdiri dari *Model*, *View* dan *Controller*.

