

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori Umum

Teori umum merupakan kumpulan dari teori-teori yang menjelaskan hal yang berhubungan dengan penelitian yang sedang dikerjakan dengan bersifat umum.

2.1.1 Data

Peristiwa berkaitan dengan kejadian asli, gambar, angka, atau tanda yang ditunjukkan pada suatu konsep disebut dengan data (Lesmana & Silalahi, 2020) sedangkan menurut (Rapita et al., 2022) data adalah metode untuk menunjukkan bahwa temuan dari hasil penelitian dapat menjadi dasar untuk studi lebih lanjut. Data dapat dikumpulkan dengan cara wawancara, kajian literatur, dan observasi (Sama & David, 2021). Kesimpulannya, data dapat didefinisikan sebagai sumber daya mentah yang diolah dan digunakan sebagai panduan untuk mengambil keputusan.

2.1.2 Informasi

Hasil dari transformasi data menjadi sesuatu yang berharga dan praktis untuk dimanfaatkan sebagai pedoman pengambilan keputusan disebut dengan informasi (Waidah & Hursali, 2020). Jika data diolah sedemikian rupa sehingga memiliki arti bagi penerima dan memiliki efek pada keputusan, tindakan, atau keduanya. Maka pada saat ini atau yang akan datang, dapat disebut sebagai data yang telah diolah berguna dalam mendukung sistem informasi (Rizkiya, 2019).

Dapat disimpulkan bahwa informasi merupakan kumpulan data olahan yang berguna bagi penerimanya. tiga indikator yang menjadikan informasi dikatakan baik (Alvionita, 2022), antara lain:

1. Akurat

Agar sebuah informasi dapat dikatakan akurat informasi harus jelas dari sumber ke penerima karena gangguan dapat menyebabkan perubahan pada sebuah informasi.

2. Tepat Pada Waktu

Tepat pada waktu yang dimaksud adalah informasi tersebut sampai ke penerima tepat pada waktunya. Informasi yang telah lama dianggap sebagai informasi yang tidak berguna atau tidak mampu berfungsi sebagai dasar pengambilan sebuah keputusan.

3. Relevan

Informasi yang bermanfaat bagi penerimanya dapat disebut sebagai informasi yang relevan.

2.1.3 Sistem

Kumpulan komponen-komponen termasuk data, jaringan proses yang berhubungan, perangkat keras, dan perangkat lunak yang bekerjasama untuk mencapai tujuan tertentu disebut dengan sistem (Talia, 2022). Hal tersebut sejalan dengan (Junita, 2021) yang mengatakan bahwa sistem merupakan sejumlah elemen data yang terhubung dan dihubungkan oleh prosedur agar menghasilkan aktivitas dengan tujuan tertentu.

Sistem terdiri dari sejumlah komponen termasuk *input*, pemrosesan, dan, *output* yang bersatu untuk mencapai tujuan tertentu (Maulana et al., 2022), dapat ditarik kesimpulan secara sederhana bahwa sistem ini merupakan bagian-bagian yang bekerja sama untuk memfasilitasi aliran informasi untuk tujuan tertentu.

2.1.4 Sistem Informasi

Secara umum, komponen yang terdiri dari orang, teknologi, komputer, dan proses kerja yang digunakan untuk menyelesaikan tugas dikenal dengan sebutan sistem informasi (Waidah & Hursali, 2020), sedangkan dalam pengertian lain untuk menentukan pengambilan sebuah keputusan, sistem informasi berfungsi untuk pengumpulan, pemrosesan, penyimpanan, dan juga menyebarkan informasi agar dapat dijadikan sebagai pengambil keputusan. Kinerja juga dapat ditingkatkan dengan menggunakan sistem informasi (Ananda Kusuma, Supono, 2019). Sistem informasi merupakan sistematis yang digunakan pada organisasi untuk melakukan proses kebutuhan transaksi, mendukung operasional manajerial, dan mengatur kegiatan strategis organisasi sehingga laporan tersedia untuk pihak eksternal tertentu (Zega & Irmayani, 2022). Hal tersebut sejalan dengan pendapat (Sanjaya et al., 2021) yang mengatakan tujuan dari sistem informasi adalah untuk menyalurkan komunikasi melalui kombinasi orang, sumber daya, teknologi, media, proses dan kontrol.

2.1.5 Perancangan

Rancang bangun merupakan langkah-langkah pada proses informasi yang diperlukan untuk menyelesaikan tugas tertentu dari pengguna (Angraina Fitri &

Sulistio, 2020). Dalam perencanaan sistem akan direncanakan untuk membuat program yang mempermudah pengguna (Putri et al., 2022).

Hal ini sejalan dengan penelitian yang dilakukan (Subagio & Samsir, 2022) yang mengatakan rancang bangun bertujuan untuk mempermudah membuat sebuah aplikasi agar memberikan bentuk dan tata letak yang jelas. Dalam melakukan perancangan sebuah sistem informasi diperlukan alat bantu yang berfungsi untuk mempermudah perancangan sebuah sistem informasi, berikut contoh alat bantu sistem informasi antara lain: *Data Flow Diagram (DFD)*, dan *Entity Relationship Diagram (ERD)*.

2.1.6 SDLC

System Development Life Cycles merupakan pekerjaan pada sistem yang melibatkan serangkaian proses, dimulai dengan analisis sistem dan diakhiri dengan pemrograman sistem (Silalahi & Lesmana, 2022). *SDLC* adalah metode untuk membuat sistem perangkat lunak yang meliputi tahap perencanaan, analisis, desain, implementasi, pengujian, dan pengelolaan (Kurniawan & Saptadi, 2022). Terdapat beberapa model dalam mengembangkan *SDLC*, antara lain:

1. *Waterfall*.
2. *Prototype*
3. RAD
4. *Agile*
5. Dan model lainnya.

Pada penelitian yang sedang dilakukan, penulis tertarik untuk menggunakan model pengembangan *agile*. Adapun beberapa model metode *agile* (Nova et al., 2022), sebagai berikut:

1. *Scrum*

Model *scrum* adalah model yang mengutamakan kecepatan pengembangan.

2. *Agile*

Model *agile* adalah model pendekatan untuk mengembangkan sistem yang mungkin terjadi perubahan setiap saat.

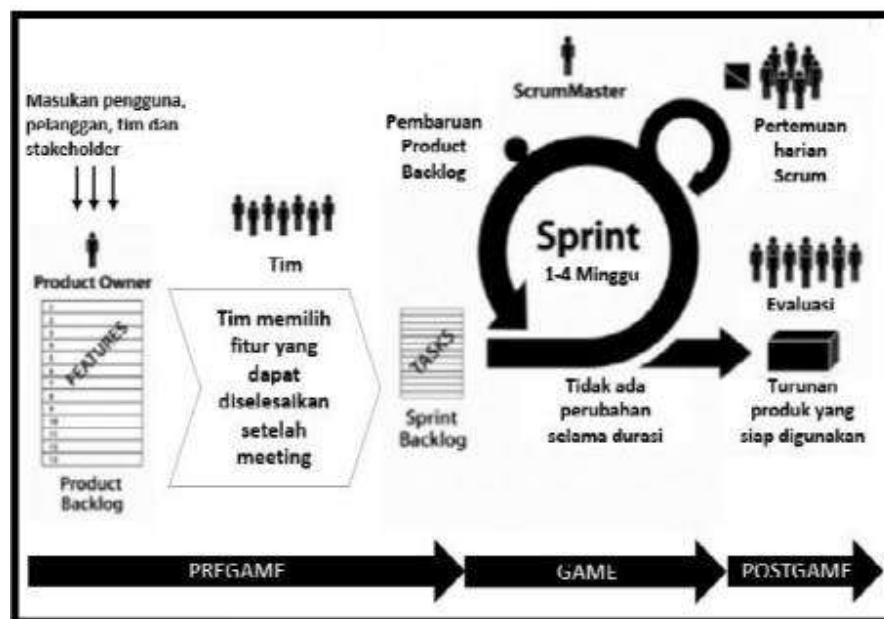
3. *Extreme Programming (XP)*

Model *XP* adalah pengembangan *software* seperti sistem informasi dengan sasaran pembangunan dengan ketetapan kebutuhan belum jelas dan tim yang kecil sampai menengah.

4. *Spiral*

Model *spiral* adalah model dengan menekankan pada analisis resiko. Dari beberapa jurnal mengenai *agile*, model *scrum* merupakan model yang efektif untuk implementasi pengembangan sistem informasi berbasis *website* yang cenderung cepat. Metode *scrum* menghasilkan perangkat lunak berkualitas tinggi sesuai dengan preferensi pengguna yang dapat digunakan untuk proyek kecil, besar, dan mudah untuk diubah (Hadji et al., 2019). Metode *scrum* memiliki beberapa tahapan (Nisef & Octaviano, 2022), antara lain:

1. *Product Backlog*, daftar komponen penyusun produk yang menjadi masukan oleh pengguna, tim, maupun stakeholder dan dijadikan dasar spesifikasi yang dibutuhkan untuk memproduksinya.
2. *Sprint Backlog*, kelompok item *product backlog* yang telah dipilih untuk dikerjakan pada *Sprint*.
3. *Sprint*, batasan waktu selama kurang lebih sebulan, di mana sebuah pekerjaan dianggap selesai dan dapat digunakan maupun dirilis.
4. *Working Increment Of The Software*, tahapan dimana evaluasi terhadap hasil dari fase *sprint* yang sudah dilakukan.



Gambar 2.1 Tahap Metode *Scrum*

Beberapa komponen *scrum* memiliki tujuan dan fungsi yang berkontribusi pada penggunaan *scrum* secara efektif. Adapun komponen *scrum* antara lain:

- a. Tim *scrum*, terdiri dari master *scrum*, tim pengembangan, dan pemilik. Tim *scrum* mengatur dirinya sendiri untuk menyelesaikan tugas seefisien mungkin tanpa melibatkan siapa pun kecuali anggota tim.
- b. *Scrum* artefak, dirancang untuk meningkatkan transparansi informasi sehingga semua pemangku kepentingan dapat memahaminya.
- c. *Scrum event*, aktivitas yang harus ada agar membantu *scrum* untuk merencanakan, mengkoordinasikan, dan meminimalkan acara terkait *scrum* lainnya.
- d. *Sprint*, merupakan tenggat waktu satu bulan untuk penyelesaian fungsional dari produk akhir incremental yang dapat dianggap selesai di dalam *sprint*.
- e. *Sprint Planning*, untuk mengatur tugas-tugas yang dilakukan dalam *sprint*.

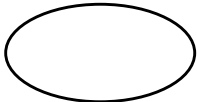



2.1.7 Unified Modeling Language (UML)

Unified Modeling Language adalah teknik mengembangkan sistem menggunakan bahasa grafis dalam mendokumentasikan dan spesifikasi pada sistem (Albhantany et al., 2022), sedangkan menurut (Musthofa & Adiguna, 2022) UML adalah bahasa yang diperuntukan untuk pemodelan bisnis, misalnya untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan sistem perangkat lunak. Adapun yang termasuk kedalam perancangan *UML* antara lain: perancangan Diagram *Use Case*, Diagram *Activity*, Diagram *Sequence*, dan Diagram *Class*.


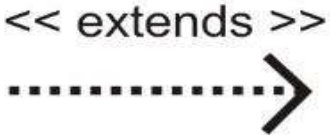
1. Diagram *Use Case*

Diagram *Use Case* berfungsi untuk menemukan fungsi sistem dan bagaimana menggunakannya. Berikut beberapa contoh simbol Diagram *Use Case*, dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol Diagram *Use Case*

SIMBOL	KETERANGAN
<p><i>Use Case</i></p> 	<p><i>Use case</i> memberikan gambaran tentang tujuan sistem sebagai satuan pertukaran pesan satuan dengan verba yang digunakan untuk menyatakan aktor.</p>
<p>Aktor</p> 	<p>Seseorang atau aktor yang memicu fungsi sistem target disebut sebagai aktor.</p>
<p><i>Association</i></p> 	<p>Garis tanpa panah yang menunjukka siapa yang mencari keterlibatan langsung dan tidak menunjukkan data digunakan untuk mewakili asosiasi anatar aktor dan <i>use case</i>.</p>
<p><i>Directed Association</i></p> 	<p>Panah terbuka menandakan asosiasi terarah antara aktor dan <i>use case</i></p>


Tabel 2.2 Lanjutan

	menunjukkan interaksi pasif antara aktor dan sistem.
<p><i>Include</i></p> 	<i>Use case</i> yang menyertakan <i>use case</i> lain atau memanggil <i>use case</i> lain, seperti memanggil fungsi dalam perangkat lunak.
<p><i>Extend</i></p> 	Menunjukkan bahwa <i>usecase</i> target memperluas perilaku <i>use case</i> sumber ke sesuatu yang spesifik.


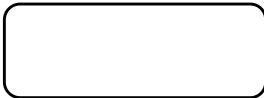
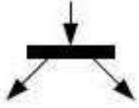

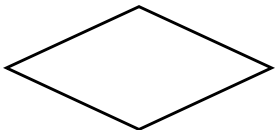
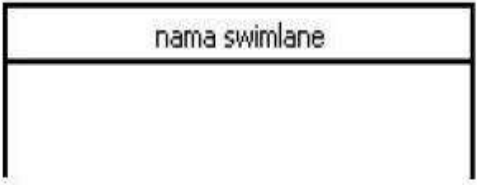
2. Diagram *Activity*

Diagram *Activity* menggambarkan bagaimana kerja sistem dilakukan dalam program. Berikut adalah beberapa contoh simbol yang digunakan dalam diagram aktivitas, dapat dilihat pada tabel 2.2.

Tabel 2.2 Simbol Diagram *Activity*

SIMBOL	KETERANGAN
<p><i>Start Point</i></p> 	<i>Start Point</i> diposisikan di sudut kiri atas, yang berfungsi sebagai awal aktivitas.

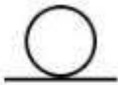
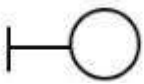


Tabel 2.2 Lanjutan

<p><i>End Point</i></p> 	<p><i>End Point</i> berfungsi sebagai akhir dari aktivitas.</p>
<p><i>Activities</i></p> 	<p><i>Activities</i> menjelaskan suatu prosedur.</p>
<p>Percabangan/<i>Fork</i></p> 	<p>Ketika dua tugas paralel digabungkan menjadi satu, percabangan digunakan untuk menunjukkan bahwa tugas tersebut dilakukan secara bersamaan.</p>
<p>Penggabungan/<i>Join</i></p> 	<p>Dekomposisi ditunjukkan dengan terjadinya penggabungan.</p>
<p><i>Decision Points</i></p> 	<p><i>Decision Points</i> menggambarkan keputusan untuk membuat pilihan yang benar atau salah.</p>
<p><i>Swimlane</i></p> 	<p>Memisahkan entitas organisasi yang bertanggung jawab atas kegiatan yang berlangsung.</p>

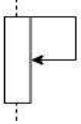


3. Diagram *Sequence*

Diagram ini menggambarkan bagaimana berbagai hal bekerja sama secara dinamis. Tujuannya adalah untuk menampilkan rangkaian komunikasi antar objek, dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol Diagram *Sequence*

SIMBOL	KETERANGAN
<p data-bbox="341 860 501 891"><i>Entity Class</i></p> 	<p data-bbox="880 860 1409 1249"><i>Entity Class</i> merupakan sebuah komponen dari sistem yang terdiri dari sekelompok kelas dalam bentuk entitas yang berfungsi sebagai gambaran awal sistem dan landasan untuk membangun basis data.</p>
<p data-bbox="341 1292 549 1323"><i>Boundary Class</i></p> 	<p data-bbox="880 1292 1409 1464"><i>Boundary Class</i> terdiri dari satu set kelas untuk interaksi antara satu atau lebih aktor dan sistem.</p>
<p data-bbox="341 1509 523 1541"><i>Control Class</i></p> 	<p data-bbox="880 1509 1409 1608"><i>Control Class</i> merupakan logika aplikasi dalam objek tidak ditetapkan ke entitas.</p>
<p data-bbox="341 1650 459 1682"><i>Message</i></p> 	<p data-bbox="880 1650 1409 1749"><i>Message</i> merupakan sarana komunikasi antar kelas.</p>


Tabel 2.3 Lanjutan

<i>Recursive</i>		<i>Recursive</i> merupakan penulisan dan pengiriman pesan pada diri sendiri.
<i>Activation</i>		<i>Activation</i> merupakan pelaksanaan operasi pada objek yang berkorelasi erat dengan berapa lama operasi berlangsung.
<i>LIFELINE</i>		<i>Lifeline</i> merupakan garis hidup pada objek.

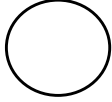






4. Diagram *Class*

Kelas sistem digambarkan dalam diagram ini bersama dengan hubungan, atribut, dan operasinya. Berikut beberapa contoh simbol Diagram *Class*, dapat dilihat pada tabel 2.4.

Tabel 2.4 Simbol Diagram *Class*

SIMBOL	KETERANGAN
<i>Class</i> 	Kelas tentang arsitektur sistem dan elemen bangunan untuk membangun program berorientasi objek.

Tabel 2.4 Lanjutan

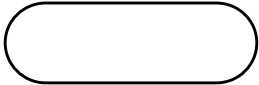


<p><i>Interface</i></p> 	<p>Antarmuka adalah komponen kunci dari pemrograman berorientasi objek.</p>
<p><i>Association</i></p> 	<p>Asosiasi dan hubungan antar kelas yang memiliki konotasi luas biasanya disertai dengan multiplisitas.</p>
<p><i>Directed Association</i></p> 	<p>Asosiasi antar kelas juga disertai dengan multiplisitas, dengan kelas yang satu menggunakan makna kelas yang lain, begitu pula sebaliknya.</p>
<p>Generalisasi</p> 	<p>Hubungan kelas sehubungan dengan generalisasi-spesialisasi (kelas umum).</p>
<p><i>Dependency</i></p> 	<p>Hubungan antar kelas yang memperhitungkan semua aspek (<i>whole-part</i>).</p>
<p><i>Aggregation</i></p> 	<p>Mengidentifikasi setiap aspek dari hubungan dan sering digunakan untuk merujuk ke suatu hubungan</p>
<p><i>Nary Association</i></p> 	<p>Hal ini bertujuan untuk mencegah keterkaitan yang melibatkan lebih dari dua hal.</p>

Berdasarkan pengertian *UML* dari beberapa sumber dapat ditarik kesimpulan bahwa *UML* merupakan bahasa untuk memodelkan sistem yang akan dibuat menggunakan notasi yang sudah disepakati (Amelia, 2021).

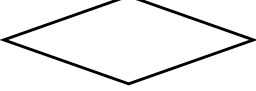


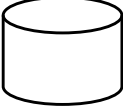



2.1.8 Aliran Sistem Informasi (ASI)

Sistem informasi bagian yang menggambarkan aliran logis dari data yang akan ditangani dari awal program hingga kesimpulannya disebut aliran. Sebuah *flowchart* terdiri dari banyak simbol yang perlu dikerjakan (Arnomo, 2022). Pendapat lain dari (Musthofa & Adiguna, 2022) yang mengatakan bahwa (ASI) merupakan arus laporan maupun formulir termasuk dengan tembusannya, oleh karena itu dibutuhkan yang namanya pedoman dalam membuat aliran sistem informasi. Contoh simbol dari aliran sistem informasi dapat dilihat pada tabel 2.5.

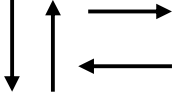
Tabel 2.5 Simbol Aliran Sistem Informasi

SIMBOL	KETERANGAN
Terminal 	Menentukan awal atau akhir dari program.
<i>Input / Output</i> 	Tergantung dari jenis peralatannya, mengidentifikasi proses <i>input</i> atau <i>output</i> .
<i>Process</i> 	Menyatakan proses yang dilakukan oleh komputer.

Tabel 2.5 Lanjutan

<p><i>Decision</i></p> 	<p>Menunjukkan kondisi tertentu yang menghasilkan sebuah kemungkinan dengan jawaban ya atau tidak.</p>
<p><i>Off- Page Connector</i></p> 	<p>Sambungan dari proses-proses dalam halaman yang berbeda.</p>
<p>Arsip</p> 	<p>Arsip dokumen yang disimpan secara manual.</p>
<p><i>Database</i></p> 	<p>Menunjukkan <i>input</i> atau <i>output</i> menggunakan <i>database</i>.</p>
<p>Rangkap Dokumen</p> 	<p>Rangkap dokumen merupakan dokumen yang mempunyai lebih dari satu rangkap.</p>
<p><i>Manual Operation</i></p> 	<p>Simbol yang mendefinisikan proses manual.</p>
<p><i>Document</i></p> 	<p>Mencetak <i>output</i> dalam bentuk dokumen.</p>

Tabel 2.5 Lanjutan

<i>Flow</i> 	Menyatakan arus dari suatu proses.
--	------------------------------------

2.2 Tinjauan Teori Khusus

Teori khusus merupakan kumpulan dari teori-teori yang menjelaskan hal yang berhubungan dengan penelitian yang sedang dikerjakan secara spesifik.

2.2.1 E-Administrasi

Administrasi bisa didefinisikan sebagai pekerjaan tulis menulis atau ketatausahaan, meliputi kegiatan menerima, mencatat, menghimpun, mengolah, mengadakan, mengirim, dan juga menyimpan (Goleman et al., 2019), administrasi dapat juga disebut dengan pelayanan terhadap subjek tertentu (Gunawan et al., 2021). Dalam pengertian luasnya, administistrasi merupakan keseluruhan proses bekerjasama antara dua orang atau lebih berdasarkan rasionalitas tertentu untuk menggapai tujuan, secara garis besar dapat disimpulkan bahwa adminstrasi merupakan proses kerjasama minimal antara dua orang untuk mencapai tujuan.

Salah satu contoh penggunaan administrasi ada pada instansi pendidikan. Administrasi sekolah merupakan segala usaha bersama untuk mendayagunakan sumber-sumber, baik personal maupun material, secara efektif dan efisien guna menunjang tercapainya tujuan pendidikan di sekolah secara optimal (Arnita, Aulia, 2019). Dapat disimpulkan bahwa administrasi merupakan seluruh ruang lingkup dari pengaturan hingga pengurusan kelompok yang memiliki perbedaan pekerjaan untuk mencapai tujuan yang sama (Lumbanraja, 2021). Administrasi saat ini juga dapat dilakukan dengan memanfaatkan teknologi, yang dikenal dengan kata “E-Administrasi”.

E-Administrasi merupakan perubahan surat dalam bentuk digital yang diciptakan untuk mempercepat waktu dalam merespon informasi yang akan masuk, dengan adanya e-administrasi pengarsipan dokumen menjadi lebih tertata karena *database* yang sudah tersimpan dapat diakses kapanpun dan dimanapun. Sejalan dengan peneliti sebelumnya (Hilma, 2018) mengatakan mengelola administrasi dengan komputer sering disebut dengan e-administrasi, karena dengan adanya e-administrasi menunjang efektifitas dan kelancaran kegiatan administrasi itu sendiri.

2.2.2 PHP

PHP merupakan bahasa skrip sisi server yang menggabungkan *HTML* agar terbentuk halaman *web* dinamis dengan tujuan agar perintah dan *sintaks* dengan memberikan eksekusi seluruhnya di *server* tetapi disertakan dalam dokumen *HTML* (Lesmana & Silalahi, 2020). Menurut (Moutaouakkil & Mbarki, 2020) bahasa *PHP* telah menjadi bahasa yang digunakan dalam mengembangkan sebuah aplikasi *web*. *PHP* juga menyediakan banyak fungsi yang umum digunakan seseorang saat ingin mengembangkan situs *web* dinamis (Rasid, 2022).

2.2.3 CodeIgniter

Framework yang digunakan untuk bahasa pemrograman *PHP* yang dibuat oleh Rick Elis pada tahun 2006 bernama *CodeIgniter* (Kharisma, 2022). *CodeIgniter* adalah *framework* yang diklaim memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya (Akbar & Rais, 2022). Menurut (Anggryani, 2022) *CodeIgniter* merupakan *framework* yang menimplementasikan pola desain MVC. Adapun MVC yang dimaksud sebagai berikut:

1. Model adalah struktur data yang membantu dalam proses berhubungan dengan *database*.
2. *View* merupakan informasi yang disampaikan kepada *user* berupa *output* dari suatu model.
3. *Controller* merupakan komponen dengan tugas mengambil *input* dari *user* dan nantinya akan diubah menjadi sebuah perintah untuk model dan *view*.

2.2.4 Framework

Framework merupakan komponen yang bisa digunakan ulang kapan saja, sehingga programmer tidak harus membuat ulang skrip yang sama pada tugas yang sama (Angraina Fitri & Sulistio, 2020). *Framework* dapat juga disebut sebagai pembangun sebuah *website*, *framework* adalah bahasa dasar pemrograman dasar yang telah dikembangkan dan dibuat untuk digunakan sehingga sebuah *website* dapat diselesaikan dalam waktu yang relatif singkat (Akbar & Rais, 2022).

2.2.5 Website

Web didefinisikan sebagai sekelompok halaman yang berguna untuk memunculkan informasi, *text*, *silent picture*, animasi, suara, dan gabungan statis dan dinamis berbentuk pola bangunan yang saling berhubungan ke masing-masing jaringan halaman *web (hyperlink)* (Lesmana & Silalahi, 2020). Adapun definisi *website* menurut (Angraina Fitri & Sulistio, 2020) adalah kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video maupun gabungan dari semuanya.

2.2.6 Mysql

Mysql merupakan *software RDBMS (server database)* yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah besar sehingga dapat diakses oleh banyak *user* dan dapat melakukan sinkronisasi (Angraina Fitri & Sulistio, 2020). *Mysql* sangat baik untuk digunakan karena dapat menangani lapisan *security*, seperti nama *host*, izin akses dalam mengakses *level subnetmask*, kata sandi, dan terpenting bersifat gratis (Sama & David, 2021).

2.2.7 XAMPP

XAMPP merupakan *software* yang digunakan untuk menjalankan sebuah *website* dengan basis *PHP* dengan menggunakan *Mysql* sebagai pengolah data utama di *local computer* (Kusuma et al., 2022), *XAMPP* sendiri merupakan singkatan dari *Apache*, *Mysql*, *PHP* dan juga *Perl* (Waidah & Hursali, 2020). Secara luas defenisi *XAMPP* merupakan *web* lengkap untuk melakukan pemrograman *web*, *XAMPP* memudahkan *web developer* untuk mengembangkan *website* di *local computer*, sehingga pembuatan *website* lebih aman dan cepat (Dirgantara & Suryadarma, 2022).