



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Perangkat Lunak

##### 2.1.1.1. Pengertian Perangkat Lunak (*Software*)

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

Menurut Mulyani (dalam Fitriadi, 2018) mendefinikan bahwa “*Software* yaitu kumpulan program komputer yang digunakan untuk memproses data”.

Menurut Rianto (2021:7) mendefinisikan bahwa “*Software* (Perangkat Lunak) merupakan instruksi (program komputer) yang mana bila ia dieksekusi akan memberikan fungsi dan unjuk kerja seperti yang diinginkan”.

Menurut Kadir (dalam Rini dan Saputra, 2021:70) mendefinisikan bahwa “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

Menurut Subakti *et al* (2022:1), “Secara umum perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi-informasi yang ada”.

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



### 2.1.1.2. Jenis- Jenis dan Contoh Perangkat Lunak

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

1. Sistem Operasi, merupakan perangkat lunak yang mengoperasikan komputer serta menyediakan antarmuka dengan perangkat lunak lain atau dengan pengguna. Contoh sistem operasi: MS DOS, MS Windows (dengan berbagai generasi), Macintosh, OS2, UNIX (dengan berbagai versi), LINUX (dengan berbagai distribusi), NetWare, dan lain-lain.
2. Program Utilitas, merupakan program khusus yang berfungsi sebagai perangkat pemeliharaan komputer, seperti anti virus, partisi hardisk, manajemen hardisk, dan lain-lain. Contoh produk program utilitas: Norton Utilities, Partition Magic, McAfee, dan lain-lain.
3. Program Aplikasi, merupakan program yang dikembangkan untuk memenuhi kebutuhan yang spesifik. Contoh: aplikasi akuntansi, aplikasi perbankan, aplikasi manufaktur, dan lain-lain.
4. Program Paket, merupakan program yang dikembangkan untuk kebutuhan umum, seperti:
  - a. Pengolah kata / editor naskah: Wordstar, MS Word, Word Perfect, AmiPro, dan lain-lain.
  - b. Pengolah angka / lembar kerja: Lotus123, MS Excell, Quattro Pro, dan yang lain.
  - c. Presentasi: MS PowerPoint, dan lain-lain.
  - d. Desain Grafis: Corel Draw, Adobe PhotoShop, dan lain-lain.
5. Bahasa Pemrograman, merupakan perangkat lunak untuk pembuatan atau pengembangan perangkat lunak lain.

### 2.1.2. Komputer

#### 2.1.2.1. Pengertian Komputer

Menurut Sanders (dalam Sindu *et al.*, 2018:17) “Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan secara otomatis untuk menerima dan menyimpan data input, memprosesnya, dan menghasilkan output dibawah pengawasan suatu

---



langkah-langkah instruksi-instruksi program yang tersimpan di memori (stored program)”.  

---

Menurut Wibowo (2021:4) “Komputer berasal dari bahasa latin *computer* yang berarti menghitung (*to compute*), karena pada awalnya komputer pertama yang dirancang digunakan untuk keperluan perhitungan. Inspirasinya diambil dari alat hitung tertua bernama “*Abaccus*” (300 SM) atau lebih dikenal dengan Sipoa yang berasal dari Negeri Cina”.

Menurut Fuori (dalam Anam *et al.*, 2023:27) mendefinisikan “Komputer adalah suatu alat pemroses data yang bisa melakukan perhitungan secara besar dan cepat, termasuk perhitungan aritmatika serta operasi logika, dan tidak ada campur tangan manusia”.

Berdasarkan beberapa definisi komputer diatas, dapat disimpulkan bahwa Komputer merupakan suatu perangkat untuk perhitungan dan mengendalikan operasi-operasi yang dapat menyimpan program dan hasil pengolahan serta bekerja secara otomatis untuk mempermudah kebutuhan manusia.

#### **2.1.2.2. Fungsi Komputer**

Menurut Zakaria (2019) Komputer memiliki 4 fungsi utama, yaitu :

##### *1. Input*

Fungsi pertama adalah *input*, yaitu menerima data atau informasi dari sumber luar. Contoh paling mudah dari fungsi ini adalah informasi dari ketikan keyboard atau klik mouse. Komputer juga bisa mendapatkan informasi dari peralatan lain atau komputer lain.

##### *2. Processing*

Fungsi utama dari komputer adalah melakukan pemrosesan. Yang diproses adalah berbagai macam data dan informasi yang diberikan oleh perangkat input. Kemudian data yang tersimpan dalam memori internal akan diproses untuk menghasilkan informasi baru yang nantinya akan dikirim ke perangkat *output*.



### 3. *Output*

Setelah diproses, informasi yang dihasilkan akan menjalankan fungsi *output*. Hasilnya akan tersedia untuk digunakan oleh pengguna atau perangkat lainnya. Contoh dari perangkat output adalah monitor komputer, *printer* dan *speaker*. Perangkat tersebut yang menyajikan output dari komputer untuk dimanfaatkan oleh pengguna atau perangkat lain.

### 4. *Storage*

Fungsi komputer yang terakhir adalah sebagai tempat untuk menyimpan informasi. Penyimpanan yang dilakukan berbeda-beda tergantung bagaimana informasinya akan digunakan. Ada yang disimpan dalam memori utama komputer, ada juga yang disimpan dalam memori tambahan. Informasi yang disimpan juga bisa dibedakan menjadi data pengguna dan data instruksi. Data instruksi inilah yang biasanya dikenal dengan sebutan program atau *software* komputer.

#### 2.1.2.3. Elemen – Elemen Sistem Komputer

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

1. Perangkat Keras (*Hardware*) adalah peralatan di sistem komputer yang secara fisik terlihat dan dapat dijamah, seperti *monitor*, *keyboard*, *mouse*, dan lain- lain.
2. Perangkat Lunak (*Software*) adalah program yang berisi perintah-perintah untuk melakukan pengolahan data, *Software* terdiri dari
  - a. Sistem operasi, DOS Ijmix, Windows, Mac, dan lain-lain.
  - b. Bahasa pemrograman, Visual Basic, C++, Pascal, Java, Visual C, dan lain-lain.
  - c. Aplikasi, MS Office, Antivirus, Winamp, Mozilla, dan lain-lain.
3. Brainware adalah manusia yang terlibat dalam mengoperasikan serta mengatur sistem komputer.

#### 2.1.3. *Internet*

##### 2.1.3.1. Pengertian *Internet*



Menurut Suwarya (2021:7) mengatakan bahwa “*Internet* singkatan dari *Interconnected Network* merupakan sistem jaringan komputer yang saling terhubung secara global untuk menghubungkan perangkat diseluruh dunia”.

Menurut Apriyanti *et al* (2022:14) “*internet* adalah sebuah jaringan komunikasi global yang terbuka. Internet adalah salah satu bukti dari kecanggihan teknologi. Internet bisa memberikan dampak positif ataupun negatif tergantung pada penggunaannya. Internet sudah menjadi komponen utama yang menyokong aktivitas masyarakat modern sekarang”.

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

*Internet* menggunakan protokol komunikasi yang sama yaitu TCP/IP (Transmission Control Protocol/Internet Protocol).

#### **2.1.3.2. Istilah – istilah *Internet***

Adapun beberapa istilah- istilah *internet*, yaitu :

##### 1. *Domain Name System* (DNS)

DNS adalah *server* yang berfungsi untuk menerjemahkan *IP address* ke sebuah nama alamat dan sebaliknya dari nama alamat ke *IP address*. DNS *Server* memberikan nama sebuah komputer dalam suatu jaringan komputer, sedangkan dalam *internet* nama yang diterjemahkan oleh DNS server merupakan *IP address* dimana *web* dapat diakses. DNS adalah *distribute database* sistem yang digunakan untuk pencarian nama komputer di jaringan yang menggunakan TCP/IP (Transmission Control Protocol/Internet Protocol).

##### 2. *Web Server*

Menurut Abdulloh (2018:4), “*Web server* merupakan perangkat lunak yang berfungsi untuk menerima permintaan (request) melalui protocol HTTP (Hypertext Transfer Protocol) atau HTTPS (Hypertext Transfer Protocol Secure) dari client kemudian mengirimkan kembali dalam bentuk halaman-halaman web”.

##### 3. *World Wide Web* (WWW)



Menurut Ibrahim, *et al* (2018), *World Wide Web* merupakan sekumpulan komputer yang menyediakan berbagai layanan informasi yang biasa disebut server dan di dalamnya terdapat sekumpulan komputer yang terintegrasi satu sama lainnya dengan menggunakan jaringan telekomunikasi yang rumit sehingga mereka dapat berkomunikasi dengan cepat.

### **2.1.3.3. Fungsi dan Tujuan *Internet***

*Internet* memiliki beberapa fungsi dan tujuan sebagai berikut:

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

### **2.1.4. Pengertian Data**

Menurut Rusmawan (2019:34) “Data adalah catatan atas kumpulan fakta”.

Menurut Saefuddin (dalam Heridiawanto *et al.*, 2021:110) “Data adalah sejumlah informasi yang dapat memberikan gambaran tentang suatu keadaan atau masalah.

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa data adalah informasi yang mencatat fakta atau kejadian untuk memberikan gambaran tentang suatu keadaan atau masalah. Data bisa berupa angka, teks, atau gambar yang terkumpul dari berbagai sumber untuk digunakan dalam analisis atau pengambilan keputusan.

Sebuah data dianggap baik apabila memenuhi syarat-syarat, sebagai berikut:

1. Objektif, yaitu data yang diperoleh dari hasil penelitian harus menggambarkan keadaan yang sebenarnya.
  2. Relevan, yaitu data yang diperoleh harus ada kaitannya dengan permasalahan yang akan diteliti.
  3. *Up to date*, yaitu data harus sesuai zaman.
  4. Representatif, yaitu data yang diperoleh dari hasil penelitian sampel harus memiliki atau menggambarkan keadaan populasinya.
  5. Dapat dipercaya, yaitu sumber data harus diperoleh dari sumber yang tepat.
- Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa



Berdasarkan sumbernya, data dibagi menjadi dua, yaitu :

1. Data Primer

Menurut Sugiyono (2020:193), data primer merupakan Sumber data primer atau sumber data utama adalah sumber data yang didapat secara langsung oleh pengumpul tanpa melalui perantara. Sedangkan data primer menurut Fuadah (2021), merupakan sumber data yang langsung memberikan data kepada pengumpul data.

2. Data Sekunder

Menurut Sugiyono (2019:193), data sekunder adalah sumber yang tidak langsung memberikan data kepada pengumpulan data.

### 2.1.5. Pengertian Basis Data (Database)

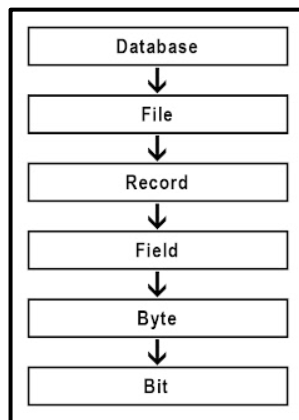
Menurut Jayanti dan Sumiari (2018:1) “Basis data yang juga dikenal *database*, terdiri dari kata basis dan data. Data merupakan catatan atas kumpulan fakta yang mewakili suatu objek. Data memiliki ciri bersifat mentah dan tidak memiliki konteks. Sedangkan basis atau basedapat diartikan sebagai markas, tempat berkumpul dari suatu objek atau representasi objek”.

Anggoro *et al* (2021:1) mengungkapkan bahwa “*Database* merupakan sekumpulan data yang saling berhubungan dan disimpan dalam computer secara sistematis dan mempunyai arti secara implisit serta dapat diolah dan diperiksa”.

Berdasarkan pengertian di atas dapat disimpulkan Basis Data merupakan kumpulan data yang tersambung dalam suatu penyimpanan secara sistematis dengan tujuan untuk mempermudah menangani informasi tersebut. Data tersebut merepresentasikan objek atau kejadian dan memiliki arti implisit yang dapat diolah dan diperiksa.



### Hirarki Basis Data



**Gambar 2.1** Hirarki Basis Data

Sumber : cikgushelly.blogspot.com

- a) **Basis data/database**, merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar *record*.
- b) **Berkas/file**, merupakan sekumpulan rekaman data yang berkaitan dengan suatu objek.
- c) **Record**, merupakan sekumpulan *field* atau atribut yang saling berhubungan terhadap obyek tertentu *fixed length record*, semua *field* dalam *record* memiliki ukuran yang tetap. Variabel *length record*, *field-field* dalam *record* dapat memiliki ukuran berbeda (metode penandaan yang digunakan adalah : *end of record marker*, indikator panjang, dan tabel posisi *record*).
- d) **Field/atribut**, merupakan unit terkecil yang disebut data, yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. *Fixed length field*, memiliki ukuran yang tetap, variabel *length field*, *field-field* dalam *record* dapat memiliki ukuran berbeda.
- e) **Byte**, adalah bagian terkecil yang dialamatkan dalam memori. *Byte* merupakan sekumpulan *bit* yang secara konvensional terdiri atas kombinasi delapan *bit* yang menyatakan sebuah karakter dalam memori ( $1 \text{ byte} = 1 \text{ karakter}$ ).
- f) **Bit**, adalah sistem *binner* yang terdiri atas dua macam nilai, yaitu 0 dan 1. sistem *binner* merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin, yang merupakan serangkaian komponen elektronik dan





hanya dapat membedakan 2 macam keadaan, yaitu ada tegangan dan tidak ada tegangan yang masuk ke rangkaian tersebut

## **2.1.6. Metode Pengembangan**

### **2.1.6.1. Metode Pengembangan *Waterfall***

Metode adalah tahap-tahap ataupun aturan untuk melakukan sesuatu *System Development Life Cycle* (SDLC) adalah sebuah proses logika yang digunakan oleh seorang system analyst untuk mengembangkan sebuah sistem informasi yang melibatkan *requirements, validation, training*, dan pemilik sistem *System Development Life Cycle* (SDLC) atau siklus hidup pengembangan sistem dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Model *waterfall* merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Pada laporan akhir ini, penulis menggunakan metode *waterfall*.

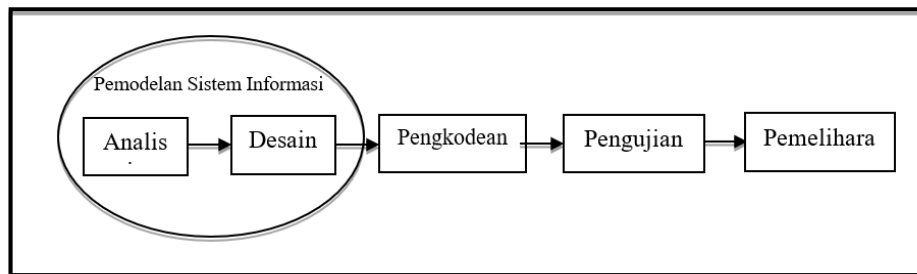
Model SDLC *waterfall* (air terjun) sering juga disebut model sekuensial linier (sequential linear) atau *classic life cycle* (alur hidup klasik). Metode air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan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

Menurut Samala dan Fajri (2020) Metode pengembangan dengan pendekatan *waterfall* merupakan pendekatan yang sistematis dan berurutan, dimulai dengan analisis kebutuhan sistem dan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan seperti desain atau perancangan sistem dan *database*, pengkodean, pengujian dan pemeliharaan dari sistem yang dikembangkan.

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa metode pengembangan sistem *waterfall* merupakan metode pengembangan sistem informasi yang sistematis, yang dimulai dengan melakukan analisis kebutuhan sistem, serta metode ini dilakukan secara urut dan berkelanjutan.



Berikut adalah gambar model *Waterfall* :



**Gambar 2.2** Ilustrasi Model *Waterfall*

Sumber : Sukamto dan Shalahuddin (2018:29).

- a. Analisis kebutuhan perangkat lunak
 

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat di pahami perangkat lunak seperti apa yang di butuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.
- b. Desain
 

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur, data, arsitektur perangkat lunak representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya, desain perangkat lunak yang di hasilkan pada tahap ini juga perlu di dokumentasikan.
- 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.
- d. Pengujian
 

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



e. 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 atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak baru.

### 2.1.7. Studi Kelayakan

“Studi kelayakan (feasibility study) merupakan suatu studi yang akan digunakan untuk menentukan kemungkinan apakah pengembangan proyek sistem layak diteruskan atau dihentikan. Studi kelayakan disebut juga dengan istilah *high point review* Jogianto ( dalam Alakel *et al.*, 2019:44).

Menurut Subair (2020: 40) studi kelayakan adalah suatu proses analisis dan evaluasi yang sistematis dan komprehensif terhadap faktor-faktor teknis, ekonomi, sosial, dan lingkungan suatu proyek untuk menentukan kelayakan proyek tersebut.

Dari dua definisi yang disebutkan, dapat disimpulkan bahwa studi kelayakan adalah suatu proses analisis dan evaluasi yang sistematis dan komprehensif terhadap faktor-faktor teknis, ekonomi, sosial, dan lingkungan suatu proyek untuk menentukan kelayakan proyek tersebut.

### 2.1.8. Kebutuhan Fungsional

Menurut Aini *et al* (2019), kebutuhan fungsional yaitu suatu kebutuhan yang berisi proses dimana sistem tersebut melakukan perintah sesuai yang diinginkan pengguna berdasarkan hasil analisis kebutuhan sistem sebelumnya.

### 2.1.9. Kebutuhan Non Fungsional

Menurut Purwantoro dan Sutjahjo (2019) mengatakan bahwa kebutuhan non-fungsional mencakup fungsi-fungsi yang membantu sistem untuk berjalan dengan baik serta dapat digunakan dengan mudah.



## 2.2. Teori Khusus

### 2.2.1. Pengertian Kamus Data

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

Menurut Kristanto (2018:7) “ kamus data merupakan kumpulan elemen-elemen yang dimanfaatkan dalam membantu penggambaran atau pengidentifikasian setiap *field* atau file di dalam sistem”.

Menurut Rusmawan (2019:36) “kamus data (data dictionary) adalah suatu penjelasan tertulis tentang suatu data yang berada di dalam *database*”.

Dari pernyataan di atas, dapat disimpulkan bahwa kamus data merupakan penjelasan tertulis tentang suatu data yang dimanfaatkan dalam membantu penggambaran setiap field di dalam sistem.

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

**Tabel 2.1.** Simbol- simbol Kamus Data

No.	Simbol	Keterangan
	=	Disusun atau terdiri dari
2.	+	Dan
3.	[   ]	Baik...atau...
4.	{ } <sup>n</sup>	N kali diulang/ bernilai banyak
5.	( )	Data opsional
6.	* ... *	Batas Komentar

Sumber : Sukamto dan Shalahuddin (2018:74).

### 2.2.2. *Unified Modelling Language (UML)*

#### 2.2.2.1. Pengertian *Unified Modelling Language (UML)*



**Gambar 2.3** Logo UML

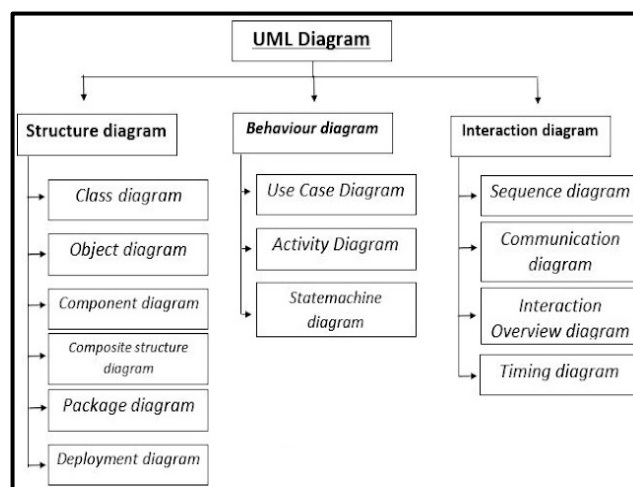
Sumber : [www.wikipedia.org](http://www.wikipedia.org)

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

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

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

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



**Gambar 2.4** Kategori dan Diagram UML

Sumber : [www.ansoriweb.com](http://www.ansoriweb.com)



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.

### 2.2.3. Jenis- Jenis Diagram Unified Modelling Language (UML)

#### 2.2.3.1. Pengertian Use Case Diagram

Menurut Sukamto dan Shalahuddin (2018:155) *Use case* merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

Menurut Haqi (2019:29) “*Use case* yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case* diagram juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

Berdasarkan pernyataan diatas, dapat disimpulkan bahwa *use case* diagram adalah sebuah pemodelan perilaku sistem informasi yang melibatkan interaksi antara aktor dan sistem. I digunakan untuk mengetahui fungsi-fungsi yang ada di dalam sistem informasi dan siapa saja aktor yang terlibat dalam penggunaan sistem tersebut.

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

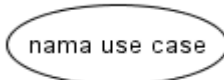
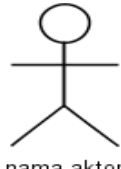




itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

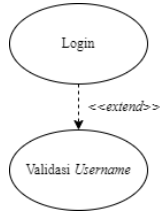
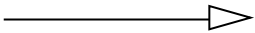
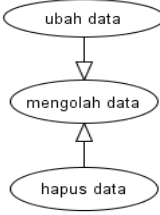
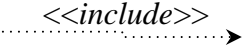
- b. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

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

**Tabel 2.2** Simbol-simbol *Use Case Diagram*

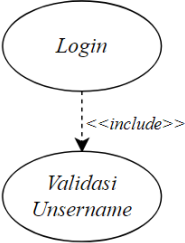
No	Simbol	Deskripsi
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i> .
2.		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang 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; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
3.		Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .
4.		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

Lanjutan Tabel 2.2 Simbol-simbol *Use Case Diagram*

No	Simbol	Deskripsi
		<p>pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya :</p>  <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.	<p>generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p> <p>misalnya:</p>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
6	<p>Menggunakan <i>include/uses</i></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 <i>include</i> di <i>use case</i>:</p>



Lanjutan Tabel 2.2 Simbol-simbol *Use Case Diagram*

No	Simbol	Deskripsi
		<p><i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan <i>use case</i> ini untuk menjalankan tugasnya.</p> <p>Misalnya pada kasus berikut:</p>  <pre> graph TD     Login((Login)) -.-&gt; &lt;&lt;include&gt;&gt;  ValidasiUsername((Validasi Username))   </pre>

Sumber : Sukamto dan Shalahuddin (2018:156).

### 2.2.3.2. Pengertian *Class Diagram*

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

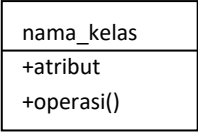



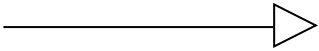
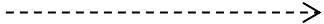
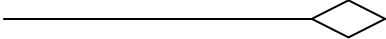
Menurut Haqi (2019:32) “*Class diagram* yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi yang terdapat pada sistem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

Berdasarkan penjelasan di atas dapat disimpulkan bahwa *Class Diagram* merupakan jenis diagram dalam UML yang digunakan untuk menggambarkan struktur sistem berdasarkan kelas-kelas dan paket-paket yang ada. *Class diagram* memberikan gambaran tentang hubungan dan relasi di dalam sistem, membantu dalam pemahaman dan pengembangan sistem secara efisien.



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> 	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 (2018:146-147).



### 2.2.3.3. Pengertian *Activity Diagram*






Menurut Sukamto dan Shalahuddin (2018:161) *Activity Diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

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

Berdasarkan penjelasan di atas dapat disimpulkan bahwa *Activity diagram* atau diagram aktivitas digunakan untuk memodelkan proses-proses pada sistem, sementara class diagram digunakan untuk menunjukkan kelas-kelas dalam sistem beserta hubungan, atribut, dan operasi yang terkait dengan kelas tersebut.

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

**Tabel 2.4** Simbol-simbol *Activity Diagram*

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

Tabel 2.4 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
6.	<p>atau</p>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

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

#### 2.2.3.4. Pengertian *Sequence Diagram*

*Sequence* diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima objek. Oleh karena itu, untuk menggambarkan diagram *sequence* harus diketahui objek-objek yang terlibat dalam sebuah use case beserta mode-mode yang dimiliki kelas yang diinstansi menjadi objek (Sukamto dan Shalahuddin, 2018:165).


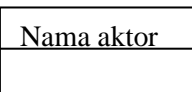
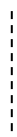
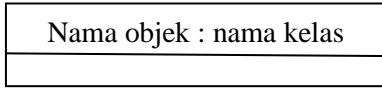
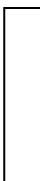
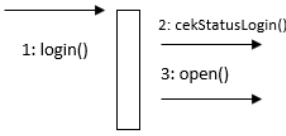
Menurut Haqi (2019:31) “*Sequence* diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case* diagram”.

Berdasarkan pernyataan diatas, dapat disimpulkan bahwa *Sequence Diagram* merupakan salah satu jenis diagram pada UML yang digunakan untuk menjelaskan interaksi antara objek berdasarkan urutan waktu. *Sequence Diagram* juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk mencapai tujuan tertentu pada *Use Case Diagram*.


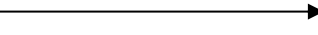
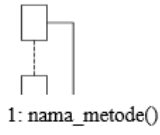
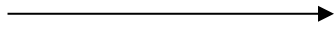
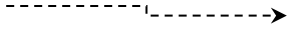


Simbol-simbol pada *Sequence Diagram* adalah sebagai berikut:

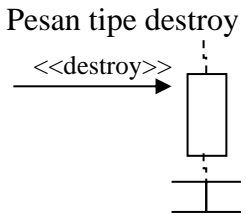
**Tabel 2.5** Simbol-simbol *Sequence Diagram*

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>Nama objek : nama kelas</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,.</p> <p>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 suatu 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 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.

Lanjutan Tabel 2.5 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
9.		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 (2018:165-167).

#### 2.2.3.4. Pengertian *Entity Relationship Diagram (ERD)*

Menurut Riski (2020) “*Entity Relationship Diagram (ERD)* adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Di dalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek *Entity* (data) dan *relationship* (hubungan), yang ada pada *Entity* berikutnya.”

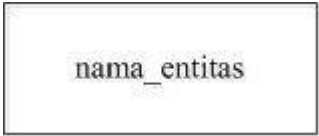
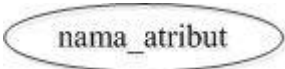
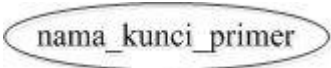
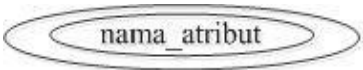

Menurut Anamisa *et al* (2020:15), “*Entity Relationship Diagram (ERD)* adalah suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi. ERD digunakan untuk menyusun struktur data dan hubungan antar data, dan untuk menggambarkannya digunakan notasi, simbol, bagan, dan lain sebagainya.”

Berdasarkan pernyataan diatas, dapat disimpulkan bahwa *Entity Relationship Diagram* merupakan model teknik pendekatan yang menyatakan atau menggambarkan hubungan antar objek data atau *entity* dan hubungan antar relasi pada struktur data. ERD digunakan untuk memudahkan pemahaman terhadap struktur data dan hubungan antar data



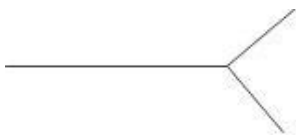
Simbol- simbol *Entity Relationship Diagram* adalah sebagai berikut:

**Tabel 2.6** Simbol-simbol *Entity Relationship Diagram*

No	Simbol	Deskripsi
1.	Entitas/ Entity 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas.
3.	Atribut Kunci Primer 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan, biasanya berupa id, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.	Atribut Multinilai/Multi Value 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
6.	Assosiasi/association	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah



Lanjutan Tabel 2.6 Simbol-simbol *Entity Relationship Diagram*

No	Simbol	Deskripsi
		pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.

Sumber : Sukamto dan Shalahuddin (2018: 50-51)

Kardinalitas *Entity Relationship Diagram* adalah sebagai berikut:

- a. Satu ke satu (one to one atau 1:1 )

Satu ke satu (one to one), Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.



**Gambar 2.5** Kardinalitas *one to one*

Sumber : fairuzelsaid.upy.ac.id

- b. Satu ke banyak (one to many atau 1:n)

Satu ke banyak (one to many), Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.



**Gambar 2.6** Kardinalitas *one to many*

Sumber : fairuzelsaid.upy.ac.id



c. Banyak ke banyak (many to many atau n:n)

Banyak ke banyak (many to many), Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.



**Gambar 2.7** Kardinalitas *many to many*

Sumber : fairuzelsaid.upy.ac.id

## 2.3. Teori Judul

### 2.3.1. Pengertian Aplikasi

Menurut Setyawan dan Munari (2020:28) “aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Aplikasi dibuat untuk memudahkan pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan, dan penambahan data yang dibutuhkan”.

Menurut (Setiawan dalam Solikin, 2022) “aplikasi adalah sebuah perangkat lunak (software) yang berisi sebuah *coding* atau perintah yang dimana bisa diubah sesuai dengan keinginan”.

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa Aplikasi adalah perangkat lunak (software) yang menggabungkan beberapa fitur tertentu dengan cara yang dapat diakses oleh pengguna.

Klasifikasi dan Jenis Aplikasi :

1. *Real Time Software*

Merupakan klasifikasi aplikasi yang dapat berguna dalam mengamati, mengendalikan, maupun menganalisa sebuah keadaan di dunia nyata secara langsung.

2. *System Software*

Merupakan aplikasi yang berguna untuk mengendalikan serta mengelola proses operasi internal yang ada pada sebuah sistem komputer.

3. *Business Software*



Merupakan aplikasi yang dibuat untuk membantu keperluan bisnis seseorang. Salah satu contohnya adalah mengatur sistem keuangan, mendata jumlah pesanan, dan masih banyak lagi.

4. *Web Based Software*

Merupakan aplikasi dengan fungsi sebagai media penghubung pengguna pada internet secara langsung.

5. *Engineering and scientific software*

Merupakan aplikasi yang dibuat dan dikembangkan guna membantu manusia menyelesaikan berbagai masalah yang ada. Hanya saja masalah tersebut bersifat secara khusus, yakni non algoritmik.

### **2.3.2. Pengertian Pemetaan**

Menurut Indrawati (2019) pemetaan adalah proses dari pembuatan peta yang mengelompokkan suatu kumpulan wilayah yang berkaitan dengan beberapa letak geografis wilayah yang meliputi dataran tinggi, pegunungan, sumber daya dan potensi penduduk yang berpengaruh terhadap sosial kultural yang memiliki ciri khas khusus dalam penggunaan skala yang tepat.

Menurut Badan Informasi Geospasial Indonesia (BIG) mendefinisikan bahwa “Pemetaan adalah kegiatan pengumpulan, pengolahan, penyajian, dan penyimpanan data geospasial serta informasi lainnya dalam bentuk peta, gambar, ataupun data digital.

Menurut *International Society for Photogrammetry and Remote Sensing* (2021) mendefinisikan bahwa “Pemetaan adalah proses pengumpulan, pemrosesan, analisis, dan penyajian data geospasial dalam bentuk peta, gambar, atau model digital untuk mendukung pengambilan keputusan dan manajemen sumber daya”.

Dari beberapa pernyataan di atas, maka dapat ditarik kesimpulan bahwa pemetaan merupakan proses pengumpulan, pengolahan, analisis, dan penyajian data geospasial dalam bentuk peta, gambar, atau model digital dengan tujuan untuk mendukung pengambilan keputusan dan manajemen sumber daya, serta memperoleh informasi terkait dengan suatu kumpulan wilayah yang berkaitan dengan beberapa letak geografis wilayah yang meliputi dataran tinggi, pegunungan,



sumber daya, potensi penduduk, serta ciri khas khusus dalam penggunaan skala yang tepat.

### **2.3.3. Pengertian *School Mapping* (Pemetaan Sekolah)**

Harjanto *et al* (2019:29) mengartikan *School Mapping* sebagai proses pemetaan dan analisis data sekolah yang meliputi lokasi, jumlah siswa dan guru, fasilitas, dan kondisi lingkungan sekitar sekolah.

Menurut Wibowo (2019) *School Mapping* adalah proses identifikasi, pemetaan, dan analisis data terkait fasilitas pendidikan, kebutuhan pendidikan, dan karakteristik masyarakat di suatu wilayah.

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *School Mapping* (Pemetaan Sekolah) adalah proses pemetaan dan analisis data terkait kondisi dan potensi sekolah, serta lingkungan sekitarnya. Tujuan utama dari *School Mapping* adalah untuk membantu dalam perencanaan dan pengambilan keputusan terkait kebijakan pendidikan.

### **2.3.4. Pengertian Persebaran**

Menurut Sukirman (2018:80) “Persebaran adalah pola penyebaran objek atau fenomena dalam suatu wilayah yang dapat diukur dan dianalisis secara spasial.

Menurut Kasim (2019:27) persebaran adalah penyebaran atau distribusi suatu fenomena atau objek dalam ruang dan waktu.”

Menurut Badan Pusat Statistik (2020:1) “Persebaran adalah cara atau pola penyebaran suatu objek atau fenomena dalam suatu wilayah, yang dapat diukur dengan menggunakan indikator seperti kepadatan, frekuensi, atau proporsi.”

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa “Persebaran merujuk pada pola penyebaran objek atau fenomena dalam suatu wilayah yang dapat diukur dan dianalisis secara spasial. Persebaran juga dapat diukur menggunakan indikator seperti kepadatan, frekuensi, atau proporsi.”

### **2.3.5. Pengertian Sekolah Menengah Pertama (SMP) Negeri**

Menurut Wahyudin (2018:32) SMP Negeri adalah jenjang pendidikan menengah pertama yang diselenggarakan oleh pemerintah daerah dan dibiayai oleh



APBN/APBD yang bertujuan untuk memberikan pendidikan yang berkualitas kepada siswa.

Menurut Sari (2020:11) SMP Negeri adalah jenjang pendidikan menengah pertama yang diwajibkan oleh negara dan diperuntukkan bagi siswa yang telah menyelesaikan pendidikan dasar.

Menurut Anwar (2021:45) SMP Negeri adalah lembaga pendidikan menengah pertama yang diselenggarakan oleh pemerintah daerah dengan tujuan untuk membantu siswa mengembangkan potensi akademik, sosial, dan emosional mereka.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa SMP Negeri adalah jenjang pendidikan menengah pertama yang diselenggarakan oleh pemerintah daerah dan dibiayai oleh APBN/APBD, yang bertujuan untuk memberikan pendidikan yang berkualitas kepada siswa dan membantu mereka mengembangkan potensi akademik, sosial, dan emosional mereka.

### **2.3.6. Pengertian *Website Geographyc Information System (WeGIS)***

Rusman dalam (Rini 2020) “*WebGIS* adalah suatu sistem yang kompleks yang dapat diakses di internet, untuk mengakuisi, menyimpan, mengintegrasikan, memanipulasi, menganalisis dan menampilkan data tanpa memerlukan perangkat lunak *GIS*. *Web mapping* memanfaatkan fungsi interaktifitas yang ada aplikasi *GIS* dalam bentuk *web*”

Menurut Purwanto (2020) *WebGIS* adalah sistem informasi geografis yang diakses melalui internet dan berbasis teknologi *web* untuk mengolah, memvisualisasikan, dan membagikan data geospasial.

Menurut Mustofa (2021) *WebGIS* adalah teknologi yang mengintegrasikan antara sistem informasi geografis (SIG) dengan *web* sehingga pengguna dapat mengakses data geospasial dengan mudah melalui *internet*.

Berdasarkan pengertian dari beberapa ahli di atas, *WebGIS* dapat diartikan sebagai suatu sistem informasi geografis yang dikembangkan dengan teknologi *web* sehingga pengguna dapat mengakses, menganalisis, dan mengelola data geospasial melalui *internet*.



Ada tiga hal pokok yang harus dipahami dan dipersiapkan untuk mengelola sebuah *website* (Tedyyana dan Kurniati, 2016)

1. *Domain Name*

Alamat di internet, dapat diibaratkan sebagai alamat rumah. Contohnya *buletindo.com*, *google.com*, *ristek.go.id*, atau *sctv.co.id*.

2. *Web Hosting*

Tempat menyimpan *file-file* untuk *website* anda, ibaratnya ini adalah fisik bangunan rumah.

3. *Web Content*

*File-file* halaman *web*, gambar, atau *file* pendukung lainnya yang dirangkai menjadi sebuah tampilan halaman *web* yang terstruktur. Ini ibaratnya perabot yang ada di dalam rumah.

### **2.3.7. Pengertian Aplikasi *School Mapping* Persebaran Sekolah Menengah Pertama (SMP) Negeri di Dinas Pendidikan Kota Palembang Berbasis *WebGIS***

Aplikasi *School Mapping* Persebaran Sekolah Menengah Pertama (SMP) Negeri di Dinas Pendidikan Kota Palembang Berbasis *WebGIS* adalah sebuah aplikasi yang digunakan untuk memetakan dan menganalisis persebaran SMP Negeri di Kota Palembang dalam lingkungan Dinas Pendidikan Kota Palembang menggunakan teknologi *WebGIS*. Aplikasi ini akan memberikan informasi yang lengkap dan akurat mengenai lokasi serta informasi profil SMP Negeri yang ada di wilayah tersebut. Dengan menggunakan *WebGIS*, aplikasi ini dapat diakses dan dikelola melalui internet sehingga memudahkan *user* atau pengguna dalam mengakses data dan melakukan analisis secara cepat dan efektif.

## **2.4. Teori Program**

### **2.4.1. Pengertian *Sublime Text***



**Gambar 2.8** Logo *Sublime Text*

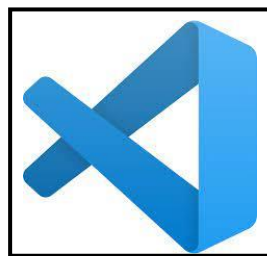
Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Setyawan dan Pratiwi (2019.88) “*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim Aplikasi ini sangatlah fleksibel dan *powerfull*”.

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

Dari penjelasan di atas dapat ditarik kesimpulan, *Sublime Text* adalah aplikasi editor teks yang sangat fleksibel, *powerfull*, dan cepat dalam mengedit kode, serta dapat berjalan di berbagai *platform operating system*. *Sublime Text* dilengkapi dengan fitur-fitur yang memudahkan proses penulisan kode seperti *auto-complete* dan *snippet*. Selain itu, aplikasi ini mendukung berbagai jenis bahasa pemrograman dan fitur *syntax highlight* hampir pada semua bahasa pemrograman yang didukung maupun dikembangkan oleh komunitas.

#### **2.4.2. Pengertian *Visual Studio Code***



**Gambar 2.9** Logo *Visual Studio Code*

Sumber : [www.wikipedia.org](http://www.wikipedia.org)



---

Menurut Rahman *et al* (2020:3) “*Visual Studio Code* adalah versi ringan dari lingkungan pengembangan *Microsoft* resmi yang secara eksklusif berfokus pada editor kode. Alat ini bersifat multiplatform dan mendukung sintaksis untuk banyak bahasa pemrograman”.

Menurut Salamah (2021:1) “*Visual Studio Code (VS Code)* ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, 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)”.

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan *Visual Studio Code* adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* yang tersedia untuk sistem operasi *multiplatform*, yaitu *Linux*, *Mac*, dan *Windows*. Teks editor ini memiliki dukungan langsung untuk bahasa pemrograman seperti *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya melalui plugin yang dapat diinstal melalui *marketplace VS Code*.

#### 2.4.3. Pengertian *Hypertext Preprocessor (PHP)*



Gambar 2.10 Logo PHP

Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Haqi *et al* (2019:9) “*PHP (Hypertext Preprocessor)* adalah bahasa skrip pemrograman yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dapat digunakan untuk membangun sebuah *CMS*. *phpMyAdmin* adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi *MySQL* melalui *World Wide Web (WWW)*. *phpMyAdmin* mendukung beerbagai operasi *MySQL*, di antaranya





mengelola basis data, tabel-tabel, bidang (fields), relasi (relations), indeks, pengguna (users), perijinan (permissions), dan lain-lain ”.

Menurut Anamisa *et al* (2020:4) “Bahasa pemrograman PHP disebut sebagai bahasa script server-side karena PHP diproses pada komputer server. Bahasa ini didesain untuk pengembangan sebuah website dan saat ini menjadi bahasa pemrograman yang banyak digunakan”.

Berdasarkan beberapa definisi di atas dapat disimpulkan , PHP adalah bahasa pemrograman script open-source yang digunakan untuk pengembangan aplikasi web dengan kemampuan menghasilkan halaman web dinamis dan interaktif yang dapat diintegrasikan dengan HTML, CSS, *JavaScript*, dan berbagai macam *database*. Bahasa pemrograman ini mudah dipelajari dan fleksibel sehingga banyak digunakan oleh para pengembang *web*.

#### 2.4.4. Pengertian *Hypertext Markup Language* (HTML)



**Gambar 2.11** Logo HTML

Menurut Suendri (2021:9) “ *Hyper Text Markup Language* adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah browser internet. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML berupa kode-kode *tag* yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan”.

Menurut (Dean dalam Wandira, 2022:4) HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML mendukung *tag* markup untuk pemformatan dan kemampuan dalam dalam menentukan struktur pada galaman web. HTML



merupakan tingkatan paling dasar dalam membangun web yang memberi arti dan struktur dasar konten *web*. *Hypertext* merujuk pada link-link yang digunakan untuk menghubungkan halaman-halaman web, baik dalam satu situs maupun antar situs web. HTML menggunakan “markup” untuk anotasi text, gambar, dan konten lainnya untuk ditampilkan dalam peramban *web*.

Dari kedua pernyataan tersebut, maka dapat ditarik kesimpulan HTML adalah bahasa markup yang digunakan untuk membuat struktur dan konten halaman *web* agar dapat diakses melalui *internet*.

#### 2.4.5. Pengertian *Cascading Style Sheet* (CSS)



**Gambar 2.12** Logo CSS

Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Sulistiono (2018:2) “CSS (*Cascading Style Sheet*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam”. CSS bukan bahasa pemrograman. Sama halnya seperti *style* dalam aplikasi pengolahan kata, *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*)”.

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 *table* menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan. CSS digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format berbeda”.



Dari beberapa definisi, dapat disimpulkan bahwa CSS adalah bahasa *stylesheet* yang digunakan untuk memberikan tampilan visual pada halaman *web*, seperti mengatur warna, jenis huruf, dan posisi elemen. CSS juga digunakan untuk memisahkan tampilan (*style*) dari struktur dan konten *HTML*, sehingga memungkinkan pengembangan web yang lebih modular dan efisien.

#### 2.4.6. Pengertian *Cross Platform, Apache, MySQL, PHP, Perl* (XAMPP)



**Gambar 2.13** Logo XAMPP

Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Haqi *et al* (2019:8) “XAMPP adalah perangkat lunak bebas (free software) yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri sebagai server yang berdiri sendiri (localhost), yang terdiri dari beberapa program, antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *perl*”.

Menurut Haerulah dan Ismiyati (dalam Ramadhani *et al.*, 2021:37) “XAMPP adalah *software* yang berfungsi untuk menjalankan website berbasis *PHP* dan menggunakan pengolah data *MySQL* di komputer lokal. XAMPP berperan sebagai *server* pada komputer. XAMPP juga dapat disebut sebagai panel server virtual, yang dapat membantu untuk melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*”.

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa XAMPP merupakan perangkat lunak bebas yang berfungsi sebagai server yang berdiri sendiri (localhost) dan digunakan untuk menjalankan *website* berbasis *PHP* dan *MySQL database*. XAMPP terdiri dari beberapa program, antara lain: *Apache*



*HTTP Server, MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *perl*.

#### 2.4.7. Pengertian *My Structured Query Language (MySQL)*



**Gambar 2.14** Logo MySQL

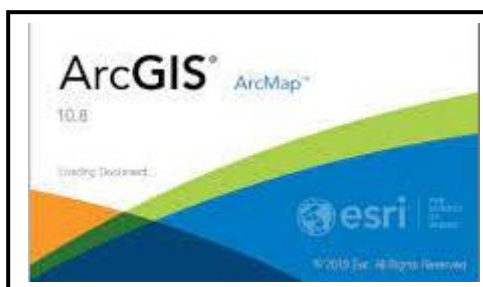
Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Fitri (2020:2) "MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database SQL* sebagai bahasa interaktif dalam mengelola data".

Menurut (Raharjo dalam Manu, 2020:17) "MySQL merupakan RDBMS (*server database*), yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user*".

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa MySQL adalah sebuah sistem manajemen basis data yang menggunakan bahasa SQL sebagai bahasa penghubung antara aplikasi dengan server *database*. MySQL dapat dijalankan pada berbagai *platform* sistem operasi dan merupakan salah satu RDBMS yang paling populer di dunia.

#### 2.4.8. Pengertian *ArcGis*



**Gambar 2.15** Logo *ArcGIS*

Sumber : [www.wikipedia.org](http://www.wikipedia.org)



Menurut ESRI (*Environment Science dan Research Institute*) “*ArcGIS* adalah sebuah platform SIG yang terdiri dari perangkat lunak SIG, aplikasi web, dan layanan yang digunakan untuk mengumpulkan, mengintegrasikan, dan menganalisis data geospasial”.

Menurut Farida *et al* (2021) “*ArcGIS* adalah perangkat lunak Sistem Informasi Geografis (SIG) yang memungkinkan pengguna untuk melakukan berbagai tugas seperti pemetaan, pemodelan spasial, analisis geografis, serta pengolahan dan visualisasi data geospasial”. Aplikasi ini dapat digunakan oleh berbagai sektor, mulai dari pemerintah, perusahaan, hingga akademisi, untuk mendukung pengambilan keputusan yang lebih baik”.

Berdasarkan beberapa definisi di atas, dapat disimpulkan *ArcGIS* adalah sebuah perangkat lunak SIG yang sangat populer dan digunakan secara luas oleh berbagai sektor. Aplikasi ini menyediakan berbagai fitur dan kemampuan untuk mengumpulkan, mengintegrasikan, dan menganalisis data geospasial, serta membantu para pengguna dalam memetakan, memodelkan, menganalisis, serta memvisualisasikan data geospasial untuk mendukung pengambilan keputusan yang lebih baik.

#### 2.4.9. Pengertian *CodeIgniter*



**Gambar 2.16** Logo *CodeIgniter*

Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Taufani (2021:6) “*CodeIgniter* adalah sebuah *web application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi php dinamis.” *CodeIgniter* menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan



menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat. *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya”.

Menurut Harani *et al* (2020:7) “*CodeIgniter* adalah sebuah aplikasi *open source* yang berupa kerangka kerja atau *framework* untuk membangun *website* menggunakan bahasa pemrograman PHP. Tujuannya memungkinkan pengembangan proyek yang lebih cepat daripada penulisan kode dasar atau kode terstruktur, dengan menyediakan banyak *library* yang biasanya digunakan dalam pengerjaan”.

Berdasarkan beberapa definisi di atas, dapat disimpulkan *CodeIgniter* adalah sebuah *framework* PHP *open source* yang menggunakan model MVC untuk membangun aplikasi web dinamis. *Framework* ini memiliki keunggulan dalam kecepatan, keringanan, dan menyediakan dokumentasi lengkap serta contoh implementasi kode. Dengan menyediakan banyak *library* yang umum digunakan, *CodeIgniter* mempercepat proses pengembangan aplikasi web dengan menghilangkan kebutuhan menulis kode dasar atau terstruktur secara manual.

#### 2.4.10. Pengertian *Application Programming Interface* (API)



**Gambar 2.17** Logo API

Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Siswanto (2019: 1) "API atau *Application Programming Interface* adalah suatu teknologi atau fitur pada platform atau sistem yang memungkinkan pengembang untuk mengakses dan memanipulasi data atau layanan yang disediakan oleh platform tersebut”.



Menurut Mulesoft (2021) “API adalah kumpulan protokol dan instruksi yang digunakan oleh pengembang perangkat lunak untuk mengintegrasikan aplikasi dan memungkinkan aplikasi berkomunikasi satu sama lain”.

Berdasarkan definisi di atas, maka dapat disimpulkan bahwa *Application Programming Interface* (API) adalah sebuah teknologi yang memungkinkan pengembang perangkat lunak untuk mengakses dan memanipulasi data atau layanan dari suatu platform atau sistem.

#### 2.4.11. Pengertian Mapbox



**Gambar 2.18** Logo Mapbox

Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Atmojo (2018:37) “*Mapbox* adalah sebuah *webservice* yang menyediakan informasi tentang geocoding dan direction dari dua buah node”.

Menurut Pratama (2020:3) “*Mapbox* merupakan salah satu penyedia *map custom* terbesar di situs ternama seperti *Foursquare*, *Pinterset*, *Evernote*, *Financial Times*, dan *Uber Technologies*. *Mapbox* memperbanyak pilihan *map custom* untuk mengisi keterbatasan yang dimiliki penyedia *map* seperti *Google Map*. Data *Mapbox* diambil dari sumber-sumber data terbuka seperti *OpenStreetMap* dan NASA, dan sumber-sumber data berbayar seperti *Digital Globe*”.

Menurut Ma'shum *et al* (2022) “*Mapbox* merupakan salah satu fitur paling populer yang membedakannya dari para pesaingnya seperti *Google Map*. *Mapbox API* adalah penyedia layanan yang memungkinkan pengguna untuk memilih dari berbagai pilihan penyesuaian. Dengan menggunakan alat yang disediakan oleh *Mapbox*, maka dapat mengunggah dan menerbitkan set data dan mengintegrasikannya dengan data *Mapbox* sendiri”.

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *Mapbox API* adalah sebuah layanan web yang kuat dan populer yang



memungkinkan pengguna untuk mengakses berbagai fitur pemetaan seperti geokoding, perutean, dan pilihan peta kustom. Dengan dukungan dari berbagai sumber data, Mapbox menyediakan fleksibilitas dan kemampuan penyesuaian yang luas untuk pengembangan aplikasi dan pemetaan *digital*.

#### 2.4.12. Pengertian *Google Maps*



**Gambar 2.19** Logo *Google Maps*

Sumber : [www.wikipedia.org](http://www.wikipedia.org)

Menurut Wahyudi *et al* (2021:123) “*Google Maps* adalah alat yang digunakan untuk menampilkan lokasi atau peta dari suatu tempat dengan mengakses server *Google Maps*”.

Menurut Santoso (2023: 51) “*Google Maps* adalah suatu peta dunia yang dapat digunakan untuk melihat suatu daerah dengan menggunakan *browser*. Kita dapat menambahkan fitur *google maps* dalam *web* yang telah dibuat baik berbayar maupun gratis sekalipun dengan *google maps api*. *Google Maps API* adalah suatu *library* yang berbentuk *javascript* dengan menggunakan *google maps API*, dapat menghemat waktu dan biaya untuk membangun aplikasi peta *digital*, sehingga fokus hanya pada data-data yang akan di tampilkan”.

Berdasarkan dua definisi di atas, maka dapat disimpulkan bahwa *Google Maps API* merupakan alat yang digunakan oleh pengembang suatu aplikasi untuk menampilkan peta dari *Google Maps* pada aplikasi yang dibuat.