



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Sistem Informasi Geografis

Menurut Prahasta dalam Mukti Wibowo, Kanedi dan Jumadi (2015:52), Sistem Informasi Geografis adalah sistem komputer yang digunakan untuk mengumpulkan, memeriksa, mengintegrasikan dan menganalisis informasi-informasi yang berhubungan dengan permukaan bumi.

Sedangkan menurut Adil dalam Tonggiroh, Jusmawati, dan Lestari (2020:2) Sistem Informasi Geografis (SIG) adalah suatu komponen yang terdiri dari perangkat keras, perangkat lunak, data geografis dan sumber daya manusia yang bekerja bersama secara efektif untuk memasukkan, menyimpan, memperbaiki, memperbarui, mengelola, memanipulasi, mengintegrasikan, menganalisis dan menampilkan data dalam suatu sistem informasi berbasis geografis.

Dari dua definisi di atas dapat disimpulkan bahwa Sistem Informasi Geografis merupakan sistem komputer yang digunakan untuk memasukkan, menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang berhubungan dengan posisi-posisi di permukaan bumi.

2.1.2 Pengertian *Monitoring*

Menurut Moerdiyanto dalam Yusrina Widyantika dan Suliyati (2019:193), *monitoring* merupakan aktivitas yang dilakukan pimpinan untuk melihat, memonitor jalannya organisasi selama kegiatan berlangsung dan menilai ketercapaian tujuan, melihat faktor pendukung dan penghambat.

Sedangkan menurut Fattah dalam Iyudhia Ekawati, Effendy dan Rasyad (2015:15), *monitoring* adalah prosedur analisis kebijakan yang digunakan untuk menghasilkan informasi tentang sebab dan akibat dari kebijakan publik.

Dari dua definisi di atas dapat disimpulkan bahwa *monitoring* adalah aktivitas melihat dan memonitor jalannya organisasi selama kegiatan berlangsung untuk menghasilkan informasi dan mencapai ketercapaian tujuan.

2.1.3 Pengertian Evaluasi

Evaluasi menurut Arikunto dalam Nardjosoeripto, Kurnia Sari dan Mitta Purbosari (2017:270) adalah kegiatan untuk mengumpulkan informasi tentang bekerjanya sesuatu, yang selanjutnya informasi tersebut digunakan untuk menentukan alternatif yang tepat dalam mengambil keputusan.

Sedangkan menurut Bryan & White dalam Choiriyah, Sugiyanti dan Hidayah (2018:150), evaluasi adalah upaya untuk mendokumentasi dan melakukan penilaian tentang apa yang terjadi dan juga mengapa hal itu terjadi, evaluasi yang paling sederhana dengan mengumpulkan informasi tentang keadaan sebelum dan sesudah pelaksanaan suatu program/rencana.

Dari dua definisi di atas dapat disimpulkan bahwa evaluasi adalah kegiatan mengumpulkan informasi, mendokumentasi dan melakukan penilaian tentang sesuatu yang dilakukan secara sistematis untuk mengetahui hasil pekerjaan karyawan dan kinerja organisasi.

2.2 Teori Khusus

2.2.1 Pengertian Metode *Rapid Application Development* (RAD)

Menurut Kendall dan Kendall dalam Habibi dan Aprilian (2019:69) *Rapid Application Development* (RAD) adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak.

Sedangkan menurut Hariyanto dalam Habibi dan Aprilian (2019:69) *Rapid Application Development* (RAD) adalah model proses perkembangan software sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis pada komponen. Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60 sampai 90 hari).

Dari dua definisi di atas dapat disimpulkan bahwa *Rapid Application Development* (RAD) merupakan metode pengembangan sistem informasi yang

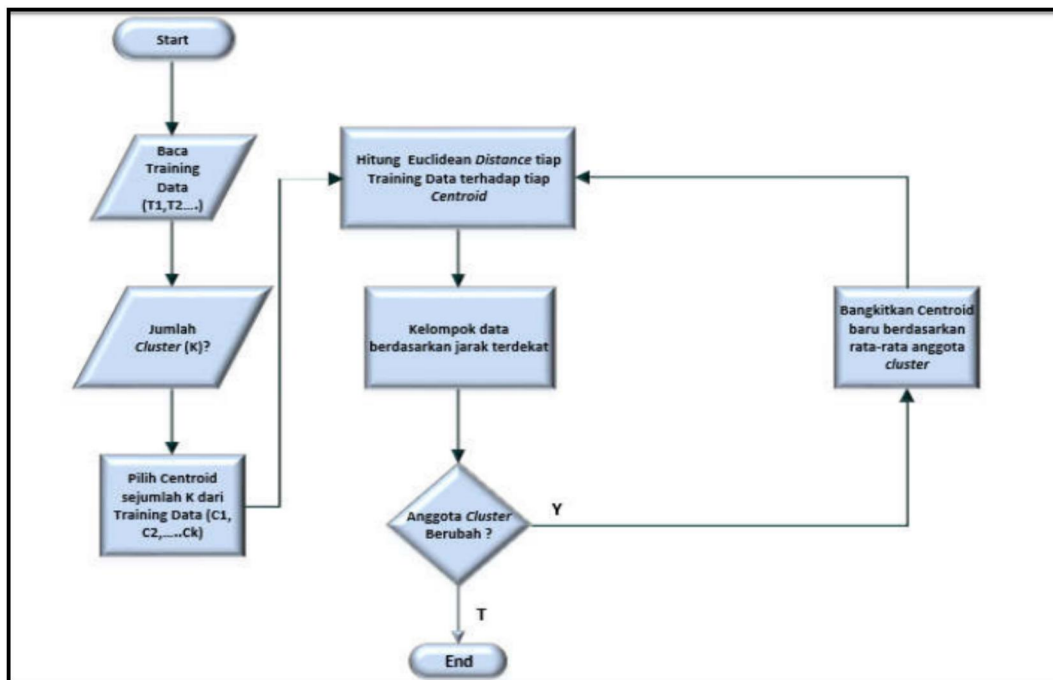


membutuhkan waktu yang cukup singkat daripada pengembangan suatu sistem informasi yang normal karena prosesnya dikerjakan dengan cepat.

2.2.2 Pengertian Metode *K-Means*

Menurut Melpa Metisen dan Latipa Sari (2015:113), *K-Means* merupakan metode clustering berbasis jarak yang membagi data ke dalam sejumlah *cluster* dan algoritma ini hanya bekerja pada atribut *numeric*. Algoritma *K-Means* termasuk *partitioning clustering* yang memisahkan data ke *k* daerah bagian yang terpisah. Algoritma *K-Means* sangat terkenal karena kemudahan dan kemampuannya untuk meng-*cluster* data yang besar dan data *outlier* dengan sangat cepat. Dalam algoritma *K-Means*, setiap data harus termasuk ke *cluster* tertentu dan bisa dimungkinkan bagi setiap data yang termasuk *cluster* tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke *cluster* lainnya.

Metode *K-Means* akan memilih pola *k* sebagai titik awal *centroid* secara acak. Jumlah iterasi untuk mencapai *cluster centroid* akan dipengaruhi oleh kandidat *cluster centroid* awal yang ditentukan secara *random* di mana jika posisi *centroid* baru tidak berubah. Nilai *K* yang dipilih menjadi *centroid* awal, akan dihitung dengan menggunakan rumus perhitungan jarak seperti *Euclidean Distance*, *Manhattan*, *Cosine Similarity* dan lain-lain. Metode perhitungan jarak adalah metode mencari jarak terdekat antara titik *centroid* dengan data. Data yang memiliki jarak terdekat dengan *centroid* akan membentuk sebuah *cluster*. Berikut ini adalah *flowchart* metode *K-Means*:



Gambar 2.1 Flowchart K-Means

Algoritma *K-Means* yang digambarkan dalam Gambar 2.1 dijelaskan sebagai berikut:

1. Tentukan k sebagai jumlah *cluster* yang akan dibentuk.
2. Tentukan k *centroid* awal secara *random*.
3. Hitung jarak setiap objek ke masing-masing *centroid* dari masing-masing *cluster* dengan menggunakan metode *Euclidian Distance*, seperti pada persamaan:

$$dik = \sqrt{\sum_j^m (C_{ij} - C_{kj})^2}$$

Dimana, dik adalah jarak antara data ke *centroid* dengan *centroid* ke- k ; m adalah jumlah atribut; C_{ij} adalah data ke- i ; ck adalah data pusat *cluster* ke- k .

4. Alokasikan masing-masing objek ke dalam *centroid* yang paling dekat.
5. Lakukan iterasi, kemudian tentukan posisi *centroid* baru dengan menggunakan persamaan:

$$C = \frac{\sum m}{n}$$

Anggota data yang termasuk ke dalam *centroid* tertentu n : jumlah data yang menjadi anggota *centroid* tertentu.

6. Ulangi langkah 3 jika posisi *centroid* baru tidak sama.

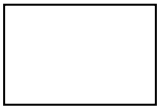
2.2.3 Pengertian *Data Flow Diagram* (DFD)

Menurut Adi Kurniawan (2020:11), *Data Flow Diagram* (DFD) adalah diagram arus data yang menggunakan simbol-simbol untuk menyajikan entitas, proses, arus data dan penyimpanan yang berkaitan dengan suatu sistem dengan tujuan untuk memberikan indikasi mengenai bagaimana data ditransformasi pada saat data bergerak melalui sistem dan menggambarkan fungsi-fungsi (dan sub fungsi) yang mentransformasikan aliran data.

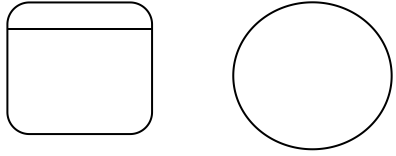

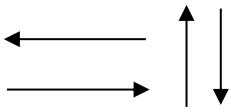
Menurut Supardi dalam Fajrian Noor, Dwi Pambudi dan Widiyanto (2018: 22) “DFD sering juga disebut DAD (Diagram Aliran Data) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).”

Dari kedua definisi di atas penulis menyimpulkan bahwa *Data Flow Diagram* (DFD) adalah gambaran sistem secara logika yang berupa aliran informasi dan transformasi informasi yang diaplikasikan sebagai data mulai dari masukan (*input*) sampai keluaran (*output*).

Tabel 2.1 Simbol-Simbol *Data Flow Diagram* (DFD)

No	Gambar	Keterangan
1		Kesatuan Luar (<i>Eksternal Entity</i>) Merupakan kesatuan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di luar lingkungan luarnya yang akan memberikan input atau menerima output sistem.

Lanjutan Tabel 2.1 Simbol-Simbol *Data Flow Diagram* (DFD)



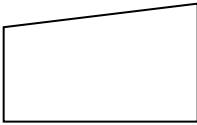
No	Gambar	Keterangan
2		Proses. Simbol ini digunakan untuk melakukan proses pengolahan data, yang menunjukkan suatu kegiatan yang mengubah aliran data yang masuk menjadi keluaran.
3		Penyimpanan Data/ <i>Data Store</i> merupakan tempat penyimpanan dokumen-dokumen atau <i>file-file</i> yang dibutuhkan.
4		Aliran Data. Menunjukkan arus data dalam proses.

(Sumber: Zefriyenni dan Santoso , 2017:22)


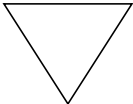
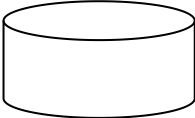

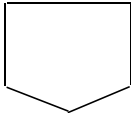
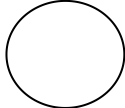

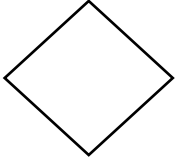

2.2.4 Pengertian *Blockchart*

Menurut Kristanto dalam Fajrian Noor, Dwi Pambudi dan Widiyanto (2018:22) *blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tetentu.

Tabel 2.2 Simbol-Simbol *Blockchart*

No	Simbol	Keterangan
1		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/berkas atau cetakan.
2		Multi Dokumen.
3		Menunjukkan <i>input</i> yang menggunakan keyboard.

Lanjutan Tabel 2.2 Simbol-Simbol *Blockchart*

No	Simbol	Keterangan
4		Proses yang dilakukan oleh komputer.
5		Menandakan dokumen yang diarsipkan (arsip manual).
6		Data penyimpanan (<i>data storage</i>).
7		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain..
9		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10		Terminasi yang menandakan awal dan akhir dari suatu aliran.
11		Pengambilan keputusan (<i>decision</i>).
12		Layar peraga monitor.

(Sumber: Kristanto, 2018:75)

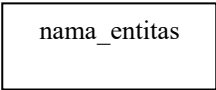
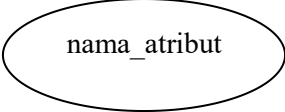
2.2.5 Pengertian *Entity Relationship Diagram (ERD)*

Menurut Sukrianto (2017:22), model ERD berisi komponen-komponen entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau sehingga dapat diketahui hubungan antara entitas-entitas yang ada dengan atribut-atributnya. Selain itu juga bisa menggambarkan hubungan yang ada dalam pengolahan data, seperti hubungan *many to many*, *one to many*, dan *one to one*.

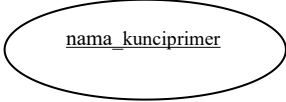
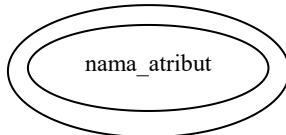
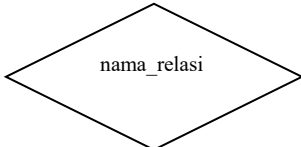
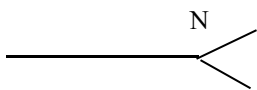
Sedangkan menurut Fatta dalam Noor dan Masykurinnisa (2017:95), ERD (*Entity Relationship Diagram*) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan dan digunakan dalam sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data. Pada akhirnya ERD bisa juga digunakan untuk menunjukkan aturan-aturan yang ada pada sistem informasi yang akan dibangun.

Dari kedua definisi di atas penulis menyimpulkan bahwa ERD (*Entity Relationship Diagram*) adalah gambar atau diagram dengan entitas yang di dalamnya terdapat atribut dan entitas yang saling berhubungan dalam sebuah pengolahan data menjadi sebuah informasi penting.

Tabel 2.3 Simbol-Simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
1.	Entitas/ <i>entity</i> 	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 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.

Lanjutan Tabel 2.3 Simbol-Simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> 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 multivalai/ <i>multivalue</i> 	<i>Field</i> atau kolom data butuh disimpan dalam satu entitas yang dapat dimiliki nilai lebih dari satu.
5	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi/ <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

(Sumber: Sukrianto, 2017:22)





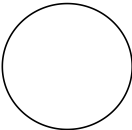

2.2.6 Pengertian *Flowchart*

Menurut Jogiyanto dalam Wahyudi, Anardani dan Dwi Hapsari (2019:24) “bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir atau arus (*flow*) di dalam program atau prosedur sistem secara logika.”

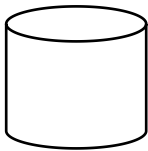
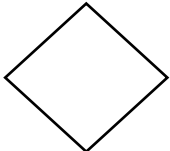



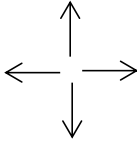
Sedangkan menurut Supardi dalam Noor, Pambudi dan Widiyanto (2018:23) “*Flowchart* merupakan Diagram Alur yang sering digunakan sistem analis dalam membuat atau menggambarkan logika program.”

Dari kedua definisi di atas penulis menyimpulkan bahwa “*Flowchart* adalah sistem analis yang menggambarkan alir data sebuah program dan proses untuk mengolah data menjadi sebuah informasi penting”.

Tabel 2.4 Simbol-Simbol *Flowchart*

No	Simbol <i>Flow Chart</i>	Fungsi <i>Flow Chart</i>
1	<i>Terminal point symbol</i> 	Simbol titik terminal digunakan untuk awal dan akhir dari suatu proses.
2	<i>Input / Output symbol</i> 	Simbol <i>input/output</i> yang digunakan untuk mewakili data input/output.
3	<i>Process symbol</i> 	Simbol proses digunakan untuk menunjukkan pengeluaran yang dilakukan oleh komputer.
4	<i>Predefined process symbol</i> 	Simbol proses terdefinisi yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
5	<i>Connector symbol</i> 	Simbol penghubung digunakan untuk menunjukkan sambungan dari bagan air yang terputus di halaman yang masih sama.
6	<i>Offline connector symbol</i> 	<i>Offline connector</i> merupakan simbol untuk masuk dan keluarnya suatu prosedur pada lembar kerja yang lain.

Lanjutan Tabel 2.4 Simbol *Flowchart*

No	Simbol <i>Flow Chart</i>	Fungsi <i>Flow Chart</i>
7	<i>Magnetic disk symbol</i> 	Simbol untuk <i>database</i> yang digunakan dalam program.
8	<i>Decision symbol</i> 	Simbol keputusan yang digunakan untuk suatu penyeleksian kondisi di dalam program.
9	<i>Display symbol</i> 	Simbol <i>display</i> digunakan untuk menunjukkan output yang ditampilkan di layar komputer.
10	<i>Document symbol</i> 	Simbol dokumen merupakan simbol yang digunakan untuk data yang berbentuk kertas.
11	<i>Display symbol</i> 	Simbol yang digunakan untuk <i>output</i> yang menunjukkan ke suatu <i>device</i> seperti <i>printer</i> .
12	<i>Line Connector</i> 	Arus dari prosedur yang dapat dilakukan dari atas ke bawah, dari bawah ke atas, dari kiri ke kanan, dan sebaliknya.

(Sumber: Jogiyanto, 2019:24)

2.2.7 Pengertian Kamus Data (*Data Dictionary*)

Menurut Kendall dan Kendall dalam Saeful Bachri (2015:26), kamus data merupakan hasil referensi data mengenai data, suatu data yang disusun oleh penganalisis sistem untuk pedoman selama melakukan analisis dan desain.

Sedangkan menurut Agustin, Suryatiningsih dan Siswanto (2016:106), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan dan keluaran dapat dipahami secara umum.

Dari kedua definisi di atas penulis menyimpulkan bahwa kamus data adalah suatu data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output, dan komponen data store.

Tabel 2.5 Simbol-Simbol Kamus Data

Simbol	Uraian
=	Disusun atau terdiri dari
+	Dan
[]	Baik... atau...
{ ⁿ }	N dikali diulang/bernilai banyak
()	Data Optional
...	Batas komentar

(Sumber: Supardi, 2015:7)

2.3 Teori Program

2.3.1 Pengertian *Website*

Menurut Hidayat dalam Syukron dan Hasan (2015:51), *website* adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar, diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Jenis-jenis *web* berdasarkan sifat atau *style*-nya adalah sebagai berikut:

1. *Website* Dinamis, merupakan sebuah website yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, .NET dan pemanfaatan *database* MySQL atau MS SQL.
2. *Website* Statis, merupakan *website* yang kontennya jarang diubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan *database*.

Sedangkan menurut Krisnayani, Resika Arthana dan Mahendra Darmawiguna (2016:2), *website* adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait di mana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Berdasarkan pendapat para ahli di atas, dapat disimpulkan bahwa *website* adalah sekumpulan halaman yang berisi informasi dan disediakan dalam jalur koneksi internet.

2.3.2 Pengertian PHP (*Hypertext Preprocessor*)

Madcoms dalam Ayu dan Permatasari (2018:20) menyatakan PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis. PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi PHP *license*. Untuk membuat program PHP kita diharuskan untuk menginstal *web server* terlebih dahulu.

Sedangkan menurut Abdulloh dalam Rose Handayani, Wijianto dan Anggoro (2018:79) PHPMyAdmin merupakan aplikasi berbasis *web* yang digunakan untuk membuat database MySQL sebagai tempat untuk menyimpan data-data website

Dari kedua definisi di atas penulis menyimpulkan bahwa PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang di proses untuk membuat program situs web dinamis.

2.3.3 Pengertian XAMPP

Madcoms dalam Ayu dan Permatasari (2018:19) menyatakan bahwa XAMPP adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PHPMyAdmin, PHP, Perl, Filezilla, dan lain. XAMPP berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan *web* memerlukan PHP, Apache, MySQL dan PHPMyAdmin.

Sedangkan menurut Wardana dalam Rose Handayani, Wijianto dan Anggoro (2018:78) dalam bukunya menerangkan bahwa XAMPP adalah paket *software*



yang di dalamnya sudah terkandung Web Server Apache, *database* MySQL, dan PHP *interpreter*.

Dari kedua definisi di atas penulis menyimpulkan bahwa XAMPP adalah paket kumpulan *software* lengkap berfungsi untuk memudahkan instalasi PHP bagi para pengembang *web* yang di dalamnya terkandung Web Server Apache, *database* MySQL, dan PHP *interpreter*.

2.3.4 Pengertian HTML (*Hyper Text Markup Language*)

Menurut Hidayatullah dan Kawistara dalam Ayu dan Permatasari (2018:20), HTML adalah bahasa standar yang digunakan untuk menampilkan halaman web. Yang bisa dilakukan dengan HTML yaitu:

- a. Mengatur tampilan dari halaman *web* dan isinya.
- b. Membuat tabel dalam halaman *web*.
- c. Mempublikasikan halaman *web* secara *online*.
- d. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.

Sedangkan menurut Abdulloh dalam Rose Handayani, Wijianto dan Anggoro (2018:77), HTML adalah singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur *website*.

Berdasarkan pendapat para ahli di atas dapat disimpulkan bahwa HTML merupakan bahasa pemrograman yang dapat menampilkan dan mengatur struktur *website*.

2.3.5 Pengertian CSS (*Cascading Style Sheet*)

Abdulloh dalam Rose Handayani, Wijianto dan Anggoro (2018:78) menjelaskan tentang bahwa CSS adalah singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*.

Menurut Hidayatullah dan Kawistara dalam Ayu dan Permatasari (2018:21), ada empat (4) cara memasang kode CSS ke dalam kode HTML/ halaman web yaitu:

1. *Inline style sheet* (memasukkan kode CSS langsung pada tag HTML).



2. *Internal style sheet* (*Embed* atau memasang kode CSS ke dalam bagian `<head>`).

3. Me-*link* ke *external* CSS.

Berdasarkan pendapat para ahli di atas dapat disimpulkan bahwa CSS merupakan *script* yang digunakan untuk mengubah tampilan desain halaman suatu *website*.

2.3.6 Pengertian *Google Maps* API

Menurut Kindarto dalam Ariyanti, Khairil dan Kanedi (2015:121), Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur Google Maps dalam *web* yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan Google Maps API. Google Maps API adalah suatu *library* yang berbentuk JavaScript.

2.3.7 Pengertian Basis Data (*Database*)

Menurut Sutabri dalam Ayu dan Permatasari (2018:17) menyatakan “basis data (*database*) adalah suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (*controlled redundancy*)”.

Sedangkan menurut Fathansyah dalam Sukrianto (2017:20) menyatakan “basis data adalah pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus. Perangkat inilah yang disebut DBMS (*Database Management System*) yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali.

Dari kedua definisi di atas penulis menyimpulkan bahwa basis data (*database*) adalah tempat untuk mengumpulkan atau menyimpan data-data yang nantinya akan diolah oleh program komputer guna untuk memperoleh informasi yang baru dan berguna.



2.3.8 Pengertian MySQL

Menurut Madcoms dalam Ayu dan Permatasari (2018:21) menyatakan “MySQL adalah sistem manajemen *Database* SQL yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur seperti *multi threaded*, *multi user* dan *SQL Database Management System* (DBMS)”.

Sedangkan menurut Faizal dan Irnawati dalam Rose Handayani, Wijianto dan Anggoro (2018:79) mendefinisikan MySQL merupakan perangkat lunak sistem manajemen berbasis data SQL (*Database Management System*) atau DBMS yang *multi thread* dan *multi user*, dengan sekitar 6 (enam) juta instalasi di seluruh dunia.

Dari kedua definisi di atas penulis menyimpulkan bahwa MySQL adalah sistem manajemen *Database* SQL yang bersifat *Open Source* yang dapat menampung data dengan jumlah yang besar, dapat di akses banyak *user* (*multi-user*).

2.3.9 Pengertian Notepad++

Menurut Madcoms dalam Ayu dan Permatasari (2018:21), Notepad++ adalah sebuah *text editor* yang sangat berguna dalam membuat program. Notepad++ menggunakan komponen *Scintilla* untuk menampilkan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan di atas sistem operasi *Microsoft Windows*.

Sedangkan menurut Reza Palevil dan Krisnawati dalam Fauzi Siregar, Handika Siregar dan Melani (2018:115), Notepad++ adalah sebuah aplikasi *text editor* yang bersifat gratis. Notepad++ menitik beratkan kegunaan aplikasi untuk *editing text* dalam waktu yang cepat dan praktis. Notepad++ mendukung banyak format bahasa pemrograman seperti PHP, HTML, JavaScript dan CSS. Aplikasi ini dapat diunduh secara bebas di alamat *notepad-plus-plus.org*.

Dari pengertian di atas dapat disimpulkan bahwa Notepad++ merupakan suatu *software* yang digunakan untuk mengedit *source* program dengan fitur yang kompatibel.



2.3.10 Pengertian JavaScript

Menurut Abdulloh dalam Rose Handayani, Wijianto dan Anggoro (2018:78), peran JavaScript dalam membuat *website* adalah memberikan efek animasi yang menarik dan interaktif dalam penanganan *event* yang dilakukan oleh pengguna *website*.

Menurut Hidayatullah dan Kawistara dalam Ayu dan Permatasari (2018:21), JavaScript (js) adalah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu web.

Dari pengertian di atas dapat disimpulkan bahwa JavaScript merupakan suatu bahasa pemrograman yang berperan dalam membuat tampilan *website* menjadi lebih menarik.