



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Krisbiantoro (2018:1), “Komputer merupakan alat yang dipakai untuk mengolah dan memproses data menurut perintah yang telah dirumuskan”.

Menurut Rianto (2021:3), “Komputer merupakan suatu perangkat elektronika yang memiliki kemampuan untuk menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis berdasarkan perangkat aturan tertentu.”

Berdasarkan pengertian diatas penulis menyimpulkan bahwa komputer adalah alat untuk mengelola dan memproses data serta membantu melaksanakan berbagai pekerjaan yang dilakukan oleh manusia, serta dapat bekerja secara otomatis.

2.1.2 Pengertian Perangkat Lunak

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

Menurut Rianto (2021:5), “Perangkat Lunak merupakan suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa perangkat lunak dapat digunakan dalam melaksanakan tugas berdasarkan perintah dari *user* sesuai dengan kriteria perangkat lunak tersebut.

2.1.3 Pengertian Internet

Menurut Sidik (2019:1), “Internet merupakan jaringan yang menghubungkan seluruh komputer di dunia melalui banyak media komunikasi, ada yang terhubung ke dalam jaringan komputer dunia melalui kabel, saluran telepon kabel, telepon seluler, radio, dan satelit”.



Menurut Aldo, et.al (2020:166), “Internet adalah suatu jaringan komunikasi yang menghubungkan satu media elektronik dengan media yang lainnya”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa internet dapat menghubungkan satu komputer kekomputer lain melalui saluran sinyal berupa kabel dan tanpa kabel (*wireless*) untuk berkomunikasi dan berbagai (tukar menukar) informasi.

2.1.4 Pengertian Website

Menurut Nurhidayah, et.al (2020:1), “*Website* merupakan kumpulan dari halaman web yang saling berhubungan yang berada pada sebuah domain di internet yang disediakan oleh perorangan, kelompok maupun organisasi dengan tujuan untuk memberi informasi kepada banyak orang.”

Menurut Elgamar (2020:3), “*Website* merupakan sebuah media yang memiliki banyak halaman yang saling terhubung (*hyperlink*), dimana website memiliki fungsi dalam memberikan informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa *website* adalah kumpulan halaman *web* yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Menurut Pane, et.al (2020:53), “Aplikasi adalah suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu”.

Menurut Wahyudi, et.al (2021:26), “Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju”.



Berdasarkan pengertian diatas penulis menyimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang dibuat untuk melayani suatu kebutuhan pada program komputer.

2.2.2 Pengertian *Monitoring*

Menurut Peraturan Pemerintah Nomor 39 Tahun 2006 dalam Mulyono dan Yumari (2017:9), “*Monitoring* merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu, dengan tujuan agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat menjadi landasan dalam mengambil keputusan tindakan selanjutnya yang diperlukan”.

Menurut Saputra dan Fadila (2020:12), “*Monitoring* merupakan proses analisa dan pengumpulan informasi berdasarkan indikator yang ditetapkan secara sistematis atas kegiatan program sehingga dapat dilakukan tindakan perbaikan untuk penyempurnaan program tersebut.”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa *monitoring* merupakan suatu kegiatan mengamati analisa dan pengumpulan informasi berdasarkan indikator, yang diperoleh dari hasil kegiatan program sehingga dapat dilakukan indakan perbaikan untuk penyempurnaan program tersebut.

2.2.3 Pengertian Alat Tulis Kantor

Menurut Julianti, et.al (2019:93), “Alat tulis kantor adalah benda-benda yang dipakai habis dalam pelaksanaan dalam pekerjaan sehari-hari dari pegawai-pegawai tata usaha”.

Menurut The Liang Gie dalam Halowo Fau (2020:28), “Alat tulis kantor adalah benda-benda yang dipakai habis dalam pelaksanaan pekerjaan sehari-hari dari pegawai pegawai tata usaha yang tergolong *Office Supplier* diantaranya pensil, pulpen, tinta,pita mesin tik, kertas blanko formulir karton berkas, jepitan kertas, lem, stabilo, penghapus, kertas HVS, kertas kuarto, kertas karton, dan lain sebagainya”.



Bedasarkan pengertian diatas penulis menyimpulkan bahwa alat tulis kantor adalah benda-benda perlengkapan kantor yang sangat penting, dalam pekerjaan manual sehari-hari pegawai di suatu instansi.

2.2.4 Metode *Double Exponential Smoothing*

Menurut Anjani et.al (2020:94-95), “*Double exponential smoothing* atau pemulusan eksponensial ganda digunakan ketika data menunjukkan adanya tren yang mana merupakan estimasi yang dihaluskan sebanyak dua kali dari pertumbuhan dan penurunan rata-rata pada akhir masing-masing periode. Kelebihan dari metode ini yaitu dapat memodelkan *trend* dan tingkat dari suatu deret waktu lebih efisien dibandingkan metode lain, karena memerlukan data yang lebih sedikit, dan menggunakan satu parameter terbaik sehingga menjadi lebih sederhana”.

Persamaan yang dipakai dalam implementasi pemulusan eksponensial linear satu-parameter dari Brown adalah sebagai berikut :

1. Menentukan Smoothing pertama ($S't$):

$$S'_t = \alpha X_t + (1 - \alpha)S'_{t-1}$$

2. Menentukan Smoothing kedua ($S''t$):

$$S''_t = \alpha S'_t + (1 - \alpha)S''_{t-1}$$

3. Menentukan besarnya konstanta (a_t):

$$a_t = 2S'_t - S''_t$$

4. Menentukan besarnya slope (b_t):

$$b_t = \alpha (S'_t - S''_t)$$

5. Menentukan besarnya forecast (f_{t+m}):

$$f_{t+m} = a_t + b_t \cdot m$$

6. Menentukan persentase eror:

$$PE = \left(\frac{X_t - F_t}{X_t} \right) \cdot 100$$

7. Menentukan nilai rata-rata diferensiasi absolut antara nilai yang diramal dan aktual:



$$MAPE = \frac{\sum |PE|}{n}$$

Keterangan :

S'_t : Nilai *smoothing exponential* tunggal.

X_t : Nilai aktual periode ke-t.

α : Parameter *smoothing* ($0 < \alpha < 1$).

S'_{t-1} : Nilai hasil *smoothing* tunggal sebelumnya (untuk nilai awal adalah 0).

S''_{t-1} : Nilai hasil *smoothing* ganda sebelumnya (untuk nilai awal adalah 0).

S''_t : Nilai *smoothing exponential* ganda.

a_t : Besarnya konstanta periode t.

b_t : *Slope* atau nilai tren dari data sesuai.

f_{t+m} : Besarnya *forecast* atau prediksi.

m : Jumlah periode yang akan diramalkan.

n : Banyaknya hasil data yang diramalkan.

PE : Persentase *error*

$MAPE$: Nilai rata-rata diferensiasi absolut

Dalam menentukan Nilai *Mean Absolute Percentage Error* (MAPE) menggunakan berdasarkan parameter α yang telah ditentukan, sebagai berikut.

| Parameter α |
|--------------------|
| 0,1 |
| 0,2 |
| 0,3 |
| 0,4 |
| 0,5 |
| 0,6 |
| 0,7 |
| 0,8 |
| 0,9 |

Tabel 2.1 Parameter α



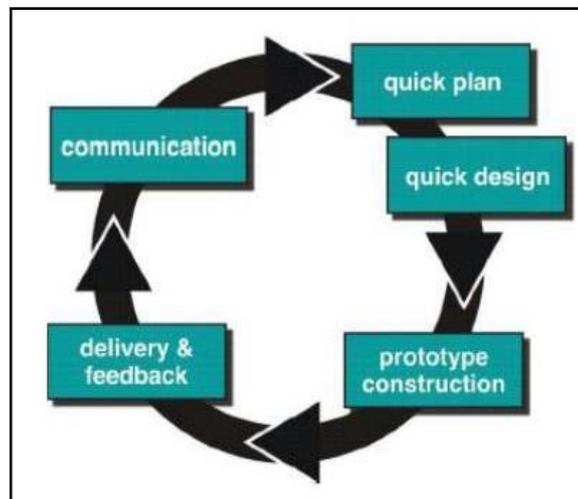
2.2 Teori Khusus

2.3.1 Model *Prototype*

Menurut Syarifudin & Ani (2019) (dikutip, Prabowo, 2020:52), “*Prototype* adalah sebuah versi awal dari perangkat lunak yang digunakan untuk mendemonstrasikan konsep, mencoba berbagai pilihan desain, dan menggali lebih banyak permasalahan serta solusinya.”

Berikut ini gambar tahapan-tahapan dari model *prototyping*:

Contoh gambar tahapan-tahapan dari model *prototyping* dapat dilihat pada gambar dibawah ini.



Gambar 2.1 Model *Prototype*

a. *Communication*

Proses pembuatan *prototype* ini disebut *prototyping*. Dasar pemikirannya adalah membuat *prototype* secepat mungkin, bahkan dalam waktu semalam, lalu memperoleh umpan balik dari pengguna yang akan memungkinkan *prototype* tersebut diperbaiki kembali dengan sangat cepat. Tahap ini merupakan tahap awal sebelum melakukan pekerjaan yang bersifat teknis. Tahap ini penting developer untuk berkomunikasi dan berkolaborasi dengan pelanggan atau perusahaan. Pada modul API dan Plugin, tahap *communication* dilakukan dengan menanyakan kepada tim yang mengerjakan proyek akhir yang sama dengan modul yang berbeda. Hal yang ditanyakan adalah *service*



atau layanan apa saja yang dibutuhkan untuk mendukung aplikasi yang dibuat setiap modulnya.

b. *Quick Plan*

Tahap ini merupakan tahap perencanaan yang dilakukan terhadap aplikasi yang akan dibuat. Perencanaan ini dilakukan dengan mencari garis besar dari aplikasi. Sehingga proses pada tahap ini bisa dibilang sangat cepat. Perencanaan cepat ini akan berfokus pada penyajian dari aspek-aspek *software* yang akan terlihat oleh *client*.

c. *Modelling Quick Design*

Tahapan selanjutnya dari metode *prototype* adalah modeling *quick plan*, dari tahapan-tahapan sebelumnya menjelaskan permasalahan dari peralatan yang harus dibutuhkan, tahapan ini menjelaskan rancangan tentang perangkat lunak yang akan dibangun. Tahap ini bisa disebut dengan tahap pembuatan sketsa. Dimana semua diawali dengan pembuatan yang belum terlihat jelas atau spesifik seperti sketsa. Jika telah sesuai dengan karakteristik lainnya, sketsa mulai dibuat lebih detail. Seperti itu rekayas perangkat lunak. Pertama harus membuat suatu model agar dapat memahami kebutuhan perangkat lunak tersebut. Kemudian desain yang dibuat harus sesuai agar mencapai kebutuhan yang diminta.

d. *Construction*

Pada tahapan ini dilakukakn penkodean berdasarkan rancangan-rancangan yang telah dibuat. Pada tahap ini *developer* membuat *coding* (pembuat *code*) baik manual atau otomatis. Jika telah selesai, maka pengujian harus langsung dilakukan untuk meminimalisir kesalahan-kesalahan dalam *coding*.

e. *Deployment, delivery & feedback*

Pada tahapan ini, program yang telah dibuat dilakukan pengujian untuk menguji fungsionalitas dari sistem yang dibuat. *Software* atau aplikasi sudah dapat dikirimkan kepada pengguna. Selanjutnya, pengguna akan memberikan umpan balik atau *feedback* kepada aplikasi dalam melakukan evaluasi jika diperlukan.

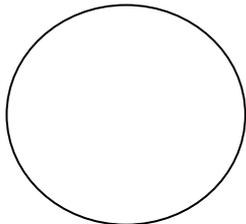


2.3.2 Pengertian DFD (*Data Flow Diagram*)

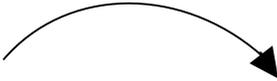
Menurut Rosa dan Shalahuddin (2018:70), “*Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

Adapun notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

Tabel 2.2 Simbol-simbol *Data Flow Diagram*

| No | Simbol | Keterangan |
|----|---|---|
| 1 |  | Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja. |
| 2 |  | File atau basis data penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada bisnis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)). Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda. |

Lanjutan Tabel 2.2 Simbol-simbol *Data Flow Diagram*

| No | Simbol | Keterangan |
|----|--|---|
| 3 |  | Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem dimodelkan. Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda. |
| 4 |  | Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>). Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”. |

Sumber: Rosa dan Shalahuddin (2018:71-72).

Rosa dan Shalahuddin (2018:72) menjelaskan, berikut ini adalah tahap-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga Context Diagram

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan, DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di *breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat



kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3, 4, 5, dan seterusnya merupakan breakdown dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

2.3.3 Pengertian *Block Chart*

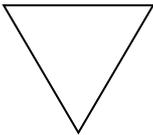
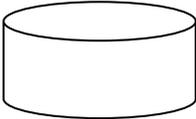
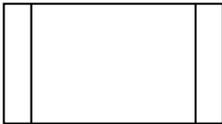
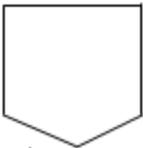
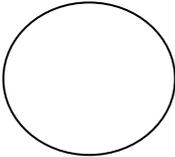
Menurut Kristanto (2018:68), “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Menurut Kristanto (2018:65) menjelaskan, “Simbol-simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini”:

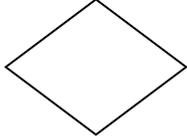
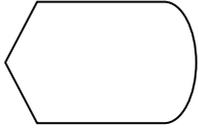
Tabel 2.3 Simbol-simbol *Block Chart*

| No | Simbol | Keterangan |
|----|---|--|
| 1 |  | Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/berkas atau cetakan. |
| 2 |  | Multi Dokumen. |
| 3 |  | Proses Manual. |

Lanjutan Tabel 2.3 Simbol-simbol *Block Chart*

| 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. |

Lanjutan Tabel 2.3 Simbol-simbol *Block Chart*

| No | Simbol | Keterangan |
|----|---|--|
| 11 |  | Pengambilan keputusan (<i>decision</i>). |
| 12 |  | Layar peraga (<i>monitor</i>). |
| 13 |  | Pemasukan data secara manual. |

Sumber: Kristanto (2018:65).

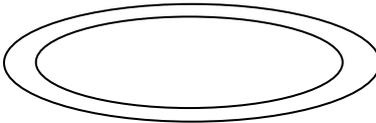
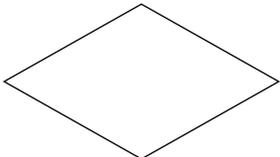
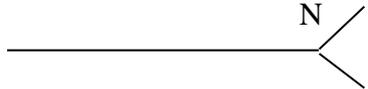
2.3.4 Pengertian *Entity Relationship Diagram* (ERD)

Menurut Rosa dan Shalahuddin (2018:50), “*Entity Relationship Diagram* (ERD) digunakan untuk pemodelan basis data relasional”. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain”. Adapun simbol-simbol dari *Entity Relationship Diagram* (ERD) adalah sebagai berikut :

Tabel 2.4 Simbol-simbol *Entity Relationship Diagram*

| No. | Simbol | Keterangan |
|-----|---|--|
| 1 | Entitas/ <i>entity</i>  | Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda memiliki data harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |

Tabel 2.4 Lanjutan Simbol-simbol *Entity Relationship Diagram*

| No. | Simbol | Keterangan |
|-----|--|---|
| 2 | Atribut  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas. |
| 3 | Atribut kunci primer  | <i>Field</i> atau kolom data yang butuh disimpan 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/multivalue  | <i>Field</i> atau kolom data yang dibutuhkan disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu. |
| 5 |  | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja. |
| 6 | Asosiasi/Association  | Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity 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: Rosa dan Shalahuddin (2018:50-51).

2.3.5 Pengertian *Flow Chart*

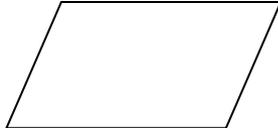
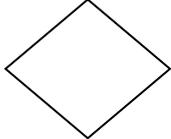
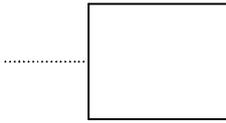
Menurut Rosa (2018:843), "*Flow Chart* adalah sebagai sebuah perangkat (*tool*) untuk membantu membuat rancangan algoritma, karena merupakan



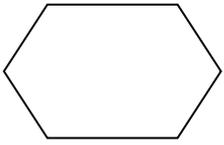
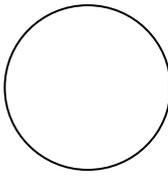
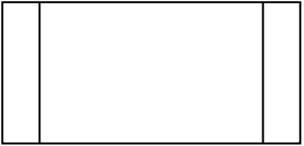
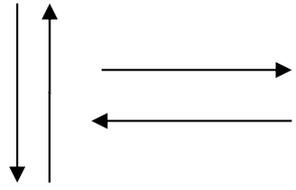
perangkat maka yang menggunakannya adalah yang merasa sesuai dengan perangkat ini, tidak harus digunakan untuk semua karena ada banyak perangkat untuk membantu cara berpikir manusia”.

Flowchart digambarkan dengan aturan simbol tertentu. Simbol-simbol dasar dari *flowchart* yang biasa digunakan untuk pemrograman dasar adalah sebagai berikut:

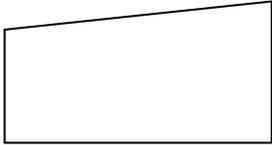
Tabel 2.5 Simbol-simbol *Flow Chart*

| No. | Simbol | Nama | Keterangan |
|-----|---|----------------------------------|---|
| 1. |  | Proses | Proses yang dilakukan secara internal di dalam komputer atau memori. |
| 2. |  | Data | Digunakan untuk beberapa operasi masukan/keluaran (<i>input/output</i> (1/0)) dengan berbagai tipe data dimasukkan bahwa komputer memperoleh masukan atau menghasilkan keluaran. |
| 3. |  | Keputusan (<i>decision</i>) | Digunakan untuk pemilihan dalam bentuk dua jawaban seperti iya/tidak (<i>yes/no, true/false</i>). |
| 4. |  | Komentar | Digunakan untuk menuliskan komentar pada diagram <i>flowchart</i> . |

Lanjutan Tabel 2.5 Simbol-simbol *Flow Chart*

| No. | Simbol | Nama | Keterangan |
|-----|---|---|--|
| 5 |  | Inisialisasi (<i>preparation</i>) | Digunakan untuk menggambarkan proses inisialisasi untuk blok pengulangan (<i>for</i>). |
| 6 |  | Konektor/penghubung (<i>connector</i>) | Mengijinkan <i>flowchart</i> digambar tanpa irisan garis atau tanpa aliran balik, atau bisa juga untuk menyambungkan dua buah garis. |
| 7 |  | Proses yang telah didefinisikan sebelumnya (<i>predefined process</i>) | Digunakan untuk memanggil sebuah rutin program atau bagian dari rutin program (<i>subroutines</i>) (fungsi atau prosedur selain yang sedang dirancang), proses atau program yang menginterupsi (program lain dari yang dirancang). |
| 8 |  | Pemberhentian (<i>terminal</i>) | Digunakan untuk memulai atau mengakhiri sebuah program, proses, atau program yang menginterupsi. |
| 9 |  | Garis aliran (<i>flow lines</i>) | Digunakan untuk menunjukkan arah aliran. |
| 10 |  | Menampilkan sesuatu ke layar (<i>display</i>) | Digunakan jika ada yang ditampilkan ke layar. |

Lanjutan Tabel 2.5 Simbol-simbol *Flow Chart*

| No. | Simbol | Nama | Keterangan |
|-----|---|---|---|
| 11 |  | Masukan manual (<i>manual input</i>) | Digunakan jika ada masukan manual dari <i>user</i> . |
| 12 |  | Operasi manual (<i>operasi manual</i>) | Biasanya digunakan jika ada blok pengulangan yang dihentikan secara manual dari masukan <i>user</i> . |

Sumber: Rosa (2018:844-845-846).

2.3.6 Pengertian Kamus Data (*Data Dictionary*)

Menurut Rosa dan Shalahudin, (2018:73), “Kamus data (*data dictionary*) adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum”.

Rosa dan Shalahudin, (2018:74), menjelaskan simbol-simbol yang di gunakan dalam kamus data, yaitu:

Tabel 2.6 Simbol-simbol Dalam Kamus Data

| No | Simbol | Keterangan |
|----|------------------|--------------------------------|
| 1 | = | Disusun atau terdiri dari |
| 2 | + | Dan |
| 3 | [] | Baik...atau... |
| 4 | { } ⁿ | N kali diulang/bernilai banyak |
| 5 | () | Data opsional |
| 6 | *...* | Batas komentar |

Sumber: Rosa dan Salahuddin (2018:74).



2.4 Teori Program

2.4.1 Pengertian MySQL

Menurut Enterprise (2018:2), “MySQL merupakan server yang melayani database. Untuk membuat dan database, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL”.

Menurut Manu (2021:25), “MySQL adalah database ternama yang dijalankan pada banyak jenis platform seperti Linux dan Windows karena sifat MySQL yaitu *open source*”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian MySQL adalah sebuah server *open source* yang dapat membuat dan mengolah database yang berupa sql dan banyak digunakan oleh para pemrogram di seluruh dunia.

2.4.2 Pengertian PHP

Menurut Enterprise (2018:1), “PHP merupakan bahasa pemrograman yang digunakan untuk membuat website dinamis dan interaktif”.

Menurut Oetomo dan Mahargiono (2020:1), “PHP adalah singkatan dari *Hypertext Preprocessor*, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan, pembuatan, dan pengembangan situs web biasanya digunakan bersamaan dengan HTML”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian PHP adalah bahasa pemrograman yang digunakan untuk pembangunan dan pengembangan sebuah aplikasi web yang berupa *script server side* dan dapat di hubungkan dengan server database sql dan nosql yang *open source*.

2.4.3 Pengertian XAMPP

Menurut Enterprise (2018:3), “XAMPP merupakan server yang paling banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi programmer pemula”.

Menurut Haqi (2019:8), “XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak operasi, yang merupakan kompilasi dari beberapa program”.



Berdasarkan pengertian diatas penulis menyimpulkan bahwa Xampp merupakan sebuah server *open source* yang banyak digunakan untuk menjalankan perintah database sql dengan integrasi program pada bahasa pemrograman PHP.

2.4.4 Pengertian Sublime Text

Menurut Rahman, et.al (2020:2), “Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi pyhton API”.

Menurut Mulyani (2020:106), “Sublime Text adalah text editor yang sangat populer karena fiturnya sangat lengkap. Text editor ini merupakan salah satu text editor yang canggih dan powerfull”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa Sublime *text* adalah aplikasi editor berupa teks untuk membantu suatu programmer dalam membangun dan mengembangkan suatu aplikasi program berbasis web. Sublime *text* juga memiliki fitur yang sangat lengkap.

2.4.5 Pengertian HTML

Menurut Sulistiono (2018:2), “HTML (*HyperText Markup Language*) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi seperti gambar, teks, videodan suara pada penjelajah web internet, yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi”.

Menurut Pane, et.al (2020:100), “HTML adalah sebuah bahasa pemrograman yang diciptakan untuk membuat program tampilan halaman web dasar, pada HTML juga terdiri dari beberapa kode yang biasa diawali dengan kurung siku buka dan diakhiri dengan kurung siku tutup”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa HTML (*Hytpertext Markup Language*) adalah bahasa standar pemrograman untuk sebuah website serta dapat menampilkan informasi seperti gambar, teks, video dan suara pada penjelajah web internet.



2.4.6 Pengertian Bootstrap

Menurut Subagia (2018:53), “*Bootstrap* adalah paket aplikasi siap pakai untuk membuat halaman *front-end* maupun *back-end* dari sebuah *website*”.

Menurut Purnama dan Watrianthos (2018:64), “*Bootstrap* adalah sebuah framework css yang dapat digunakan untuk mempermudah membangun tampilan web”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa *Bootstrap* adalah sebuah paket aplikasi berbentuk framework css yang digunakan dalam menambahkan fitur-fitur dalam membangun tampilan web.

2.4.7 Pengertian CSS

Menurut Rerung (2018:133), “CSS adalah *Cascading Style Sheets*. CSS biasanya selalu dikaitkan dengan HTML, karena keduanya memang saling melengkapi dimana HTML ditujukan untuk membuat struktur, atau konten dari halaman web”.

Menurut Azis, et.al (2019:36), “CSS (*Cascading Style Sheet*) adalah sebuah dokumen yang terdiri dari kode program yang digunakan untuk membuat elegan tampilan dari tampilan halaman *website* yang dibuat”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa CSS merupakan sebuah aturan yang terdiri dari kode program untuk mengendalikan beberapa tampilan suatu web dan dapat memberikan tampilan web lebih menarik.

2.4.8 Pengertian Black Box Testing (Pengujian Kontak Hitam)

Menurut Rosa dan Shalahudin, (2018:275-276), *Black Box Testing* Yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian



kotak hitam harus dengan kasus benar dan kasus yang salah, misalkan untuk kasus proses login maka kasus uji yang dibutuhkan adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.