



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Sujatmiko (2012:156), “komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program”.

Puspitosari (2013:1), “komputer adalah sebuah alat yang digunakan untuk mengolah data menurut perintah yang telah dirumuskan”.

Berdasarkan beberapa pendapat diatas dapat disimpulkan bahwa komputer adalah alat untuk mengolah data dengan mengikuti perintah yang telah dirumuskan.

2.1.2 Pengertian Data

Ladjamuddin (2013:9), “data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata”.

Fathansyah (2018:2), “data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang diwujudkan dalam angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya”.

Berdasarkan beberapa pengertian diatas dapat disimpulkan data adalah representasi kenyataan yang diwujudkan dalam angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

2.1.3 Pengertian Basis Data

Hidayatullah dan Kawistara (2015:147), “basis data adalah himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Sukamto dan Shalahuddin (2018:43), “basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.



Berdasarkan penjelasan diatas dapat disimpulkan bahwa basis data adalah media penyimpanan data agar dapat dimanfaatkan dan diakses dengan mudah dan cepat.

2.1.4 Pengertian Sistem

Kristanto (2012:1), “sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.

Anggraeni dan Irvani (2017:11) , “sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa sistem adalah unsur-unsur yang saling berhubungan untuk mencapai atau menyelesaikan sasaran tertentu.

2.1.5 Pengertian Informasi

Sutabri (2012:29), “informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan”.

Hutahaean (2015:9), “informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Anggraeni dan Irvani (2017:12), “informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa informasi adalah data yang sudah diolah kedalam bentuk yang lebih berguna sehingga dapat digunakan untuk pengambilan keputusan bagi penerimanya.



2.1.6 Pengertian Sistem Informasi

Kristanto (2012:12), “sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut”.

Anggraeni dan Irvani (2017:12) menjelaskan mengenai pengertian sistem informasi yaitu Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan sistem yang berfungsi untuk mengolah data untuk dapat menyediakan informasi kepada pihak luar untuk pengambilan keputusan.

2.2 Teori Khusus

2.2.1 Metode Pengembangan Sistem

Sukanto dan Shalahuddin (2018:28) menjelaskan mengenai metode pengembangan sistem Dalam pengembangan perangkat lunak atau sering disebut *Software Development Life Cycle* (SDLC). SDLC memiliki beberapa model dalam penerapan dan tahapan prosesnya, salah satunya adalah model *Waterfall*. Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*).

Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut penjelasan dari model *Waterfall*:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti yang dibutuhkan *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.



2. 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 proses pengodean.

3. Pembuatan Kode Program

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

4. Pengujian

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

5. Pendukung (*Support*) atau Pemeliharaan (*Maintenance*)

Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2.2 DFD (*Data Flow Diagram*)

Kristanto (2012:61), “*data flow diagram (DFD)* adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Ada 2 teknik dasar DFD yang umum dipakai yaitu Gane *and* Sarson dan Yourdon *and* De Marco”.

Sukamto dan Shalahuddin (2018:69) menjelaskan *Data flow diagram (DFD)* awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology (SSADM)* yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada

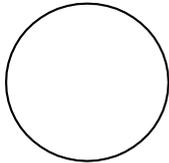


dekomposisi fungsional dari sebuah sistem. Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD menyediakan mekanisme untuk



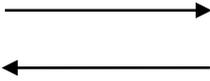
pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

Tabel 2.1 Simbol-simbol *Data Flow Diagram (DFD)* menurut Edward Yourdon dan Tom DeMarco

No	Notasi	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 atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram (ERD)</i> , <i>Conceptual Data Model (CDM)</i> , <i>Physical Data Model (PDM)</i>). catatan : nama yang diberikan pada sebuah penyimpanan biasanya kata benda



Lanjutan Tabel 2.1 Simbol-simbol *Data Flow Diagram (DFD)* menurut Edward Yourdon dan Tom DeMarco

3.		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>catatan : nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) berupa kata benda.</p>
4.		<p>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>).</p> <p>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”.</p>

(Sumber :Sukamto dan M. Shalahuddin, 2018 71-72)

Tabel 2.2 Simbol-simbol *Data Flow Diagram (DFD)* menurut Chris Gane dan Trish Sarson

No.	Notasi	Keterangan
1.		<p>Entity Luar, merupakan sumber atau tujuan dari aliran data dari atau ke sistem. Entity luar merupakan lingkungan luar sistem, jadi tidak tahu menahu mengenai apa yang terjadi di entity luar.</p>



Lanjutan Tabel 2.2 Simbol-simbol *Data Flow Diagram* (DFD) menurut Chris Gane dan Trish Sarson

2.		Aliran data, menggambarkan aliran data dari suatu proses ke proses lainnya.
3.		Proses, proses atau fungsi yang mentransformasikan data secara umum.
4.		Tempat penyimpanan, merupakan komponen yang berfungsi untuk menyimpan data atau file.

(Sumber: Kristanto, 2012:62-63)

2.2.2.1 Tahapan-tahapan Perancangan DFD

Menurut Sukanto dan M. Shalahuddin (2018: 72-73), berikut ini adalah 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 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



diatasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

2.2.2.2 Peraturan Penting Dalam DFD

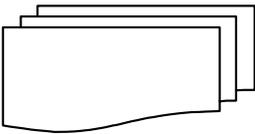
Menurut Kristanto (2012: 66-67), dalam penggambaran simbol DFD ada beberapa aturan yang harus diperhatikan sehingga dalam penggambarannya tidak terjadi kesalahan, yaitu:

1. Antara entiti luar tidak diijinkan terjadi hubungan atau relasi.
2. Tidak boleh ada aliran data antara entiti luar dengan *data store*.
3. Untuk alasan kerapian, entiti luar atau *data store* boleh digambar beberapa kali dengan tanda khusus, misalnya diberi nomor.
4. Satu aliran data boleh menggambarkan beberapa struktur data.
5. Bentuk anak panah aliran data boleh bervariasi.
6. Semua objek harus mempunyai nama.
7. Aliran data selalu diawali dan diakhiri dengan proses.
8. Semua aliran data harus mempunyai tanda arah.

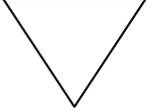
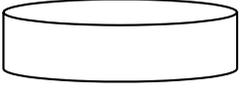
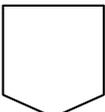
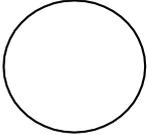
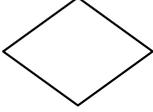
2.2.3 Block Chart Diagram

Kristanto (2012:75), "*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 maupun transaksi".

Tabel 2.3 Simbol-Simbol *Blockchart*

No.	Notasi	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku atau bendel atau berkas atau cetakan
2.		Multi dokumen

Lanjutan Tabel 2.3 Simbol-Simbol *Blockchart*

3.		Proses manual
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>)

Lanjutan Tabel 2.3 Simbol-Simbol *Blockchart*

12.		Layar peraga (<i>monitor</i>)
13.		Pemasukan data secara manual

(Sumber: Kristanto, 2012:75-77)

2.2.4 Flowchart Diagram

Ladjamuddin (2013:211), “*flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah”.

Sitorus (2015:14), “*flowchart* merupakan sebuah alat bantu yang berbentuk diagram alir di gunakan untuk menggambarkan sebuah algoritma yang terstruktur yang mudah dipahami oleh orang lain”.

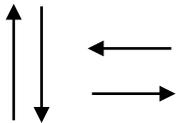
Jadi menurut beberapa pengertian diatas dapat disimpulkan bahwa *flowchart* merupakan sebuah alat bantu berbentuk diagram alir yang digunakan untuk menggambarkan urutan prosedur dalam sebuah program.

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses didalam program. Simbol-simbol yang digunakan dapat dibagi menjadi 3 kelompok, yakni:

1. *Flow Direction Symbols* (Simbol Penghubung atau Alur)

Simbol yang digunakan untuk menghubungkan antara simbol satu dengan yang lain. Simbol ini disebut juga *connecting line*.

Tabel 2.4 *Flow Direction Symbols*

No	Simbol	Keterangan
1.		Simbol Arus / Flow Untuk menyatakan jalannya arus suatu proses

Lanjutan Tabel 2.4 *Flow Direction Symbols*

2.		Simbol <i>Communication Link</i> Untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
3.		Simbol <i>Connector</i> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
4.		Simbol <i>Offline Connector</i> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.

(Sumber: Ladjamuddin, 2013:266)

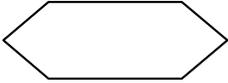
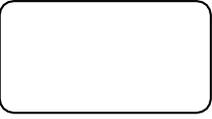
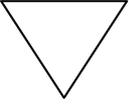
2. *Processing Symbols* (Simbol Proses)

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses atau prosedur.

Tabel 2.5 *Processing Symbols*

No	Simbol	Keterangan
1.		Simbol <i>Offline Connector</i> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
2.		Simbol <i>Manual</i> Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).
3.		Simbol <i>Decision</i> atau <i>Logika</i> Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya atau tidak.

Lanjutan Tabel 2.5 *Processing Symbols*

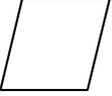
4.		Simbol <i>Predefined Process</i> Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk member harga awal.
5.		Simbol Terminal Untuk menyatakan permulaan atau akhir suatu program.
6.		Simbol <i>Keying Operation</i> Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang disebut <i>keyboard</i> .
7.		Simbol <i>off-line storage</i> Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
8.		Simbol <i>Manual Input</i> Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

(Sumber: Ladjamuddin, 2013:267)

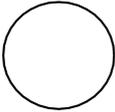
3. *Input – Output Symbols* (Simbol Input - Output)

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media input atau output.

Tabel 2.6 *Input – Output Symbols*

No	Simbol	Keterangan
1.		Simbol <i>Input – Output</i> Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.

Lanjutan Tabel 2.6 *Input – Output Symbols*

2.		Simbol <i>Punched Card</i> Untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
3.		Simbol <i>Magnetic-tape Unit</i> Untuk menyatakan <i>input</i> berasal dari pita <i>magnetic</i> atau <i>output</i> disimpan ke pita <i>magnetic</i> .
4.		Simbol <i>Disk Storage</i> Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
5.		Simbol <i>Document</i> Untuk mencetak laporan ke printer.
6.		Simbol <i>Display</i> Untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer).

(Sumber: Ladjamuddin, 2013:268)

2.2.5 ERD (*Entity Relationship Diagram*)

Ladjamuddin (2013:142), “ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”.

Sulianta (2017:158), “*entity relationship diagram* (ERD) merupakan diagram untuk merancang tabel-tabel yang nantinya akan diimplementasikan pada basis data”.

Berdasarkan pengertian diatas dapat disimpulkan ERD adalah diagram atau model jaringan yang menyusun data menjadi table yang nantinya diimplementasikan pada basis data.

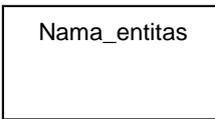
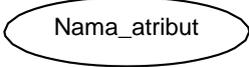
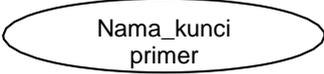
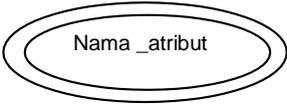
Sukanto dan Shalahuddin (2018:73-74) menjelaskan:

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan matematika. ERD digunakan untuk pemodelan basis data relasional. ERD memiliki



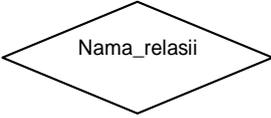
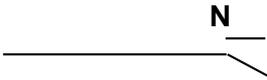
beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang paling banyak digunakan adalah notasi dari Chen.

Tabel 2.7 Simbol-simbol *Entity Relationship Diagram*

No.	Nama	Notasi	Keterangan
1.	Entitas atau <i>entity</i>		Entitas merupakan data inti yang akan disimpan; bakal table 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 table.
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 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 atau <i>multivalue</i>		<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.



Lanjutan Tabel 2.7 Simbol-simbol *Entity Relationship Diagram*

5.	Relasi		Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi atau <i>association</i>		<p>Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.</p> <p>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.</p>

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

2.2.6 Kamus Data (*Data Dictionary*)

Kristanto (2012:72), “kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap field atau file didalam sistem”.

Sukamto dan Shalahuddin (2018:73-74), “kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur”.

Kamus data biasanya berisi:

1. nama – nama dari data
2. digunakan pada – merupakan proses-proses yang terkait data
3. deskripsi – merupakan deskripsi data



4. informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

Tabel 2.8 Simbol-Simbol Kamus Data

No.	Notasi	Keterangan
1.	=	Artinya adalah terdiri atas
2.	+	Artinya adalah dan
3.	()	Artinya adalah oprasional
4.	[]	Artinya adalah memilih salah satu alternative
5.	**	Artinya adalah komentar
6.	@	Artinya adalah identifikasi atribut kunci
7.		Artinya adalah pemisah alternative symbol []

(Sumber: Sukamto dan Shalahuddin, 2018:74)

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Asropudin (2013:6), “aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms- Word, Ms-Excel”.

Sujatmiko (2012:23), “aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu”.

Berdasarkan pengertian diatas dapat disimpulkan aplikasi adalah *software* yang dibuat oleh perusahaan komputer untuk mengerjakan tugas tertent



2.3.2 Pengertian Penjualan

Alvonco (2014:235), “penjualan adalah suatu kegiatan yang ditujukan untuk mencari pembeli, memengaruhi, dan memberikan petunjuk agar pembeli dapat menyesuaikan kebutuhannya dengan produk yang ditawarkan serta mengadakan perjanjian mengenai harga yang menguntungkan bagi kedua belah pihak”.

Zimmerer (2014:235), “penjualan adalah sumber utama aliran kas yang masuk kedalam perusahaan”.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa penjualan adalah kegiatan yang ditujukan untuk mencari pembeli dengan usaha pemasaran produk.

2.3.3 Pengertian Persediaan

Margaretha (2005:145), Persediaan merupakan sejumlah bahan/barang yang disediakan oleh perusahaan, baik berupa barang jadi, bahan mentah, maupun barang dalam proses yang disediakan untuk melancarkan operasi perusahaan guna memenuhi permintaan konsumen setiap waktu.

Sigit (2005:161), Persediaan merupakan bahan atau barang yang disimpan untuk tujuan tertentu, antara lain untuk proses produksi, jika beruoa bahan mentah maka akan diproses lebih lanjut, jika berupa komponen (*spare part*) maka akn dijual kembali menjadi barang dagangan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa persediaan adalah barang/bahan yang telah disiapkan namun disimpan oleh perusahaan.

2.3.4 Pengertian Barang

Ramli (2013:61), “barang adalah setiap benda, baik berwujud maupun tidak berwujud, bergerak maupun tidak bergerak, yang dapat diperdagangkan, dipakai, dipergunakan, atau dimanfaatkan oleh pengguna barang”.

Nasution (2015:8), “barang bukan hanya sesuatu yang berwujud, tetapi juga yang tidak berwujud”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa barang adalah segala sesuatu yang berwujud atau tidak berwujud yang dapat digunakan oleh pengguna.



2.3.5 Pengertian Website

Yuhefizar, Mooduto, Hidayat (2009:2), Website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah dominan yang mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang berhubungan.

Hidayat (2010:2), Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang dapat digunakan untuk menampilkan 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.

Berdasarkan pengertian diatas dapat disimpulkan bahwa website adalah halaman yang diisi oleh informasi, gambar, atau animasi.

2.4 Teori Program

2.4.1 Pengertian XAMPP

Murya (2017:1), “XAMPP adalah perangkat lunak gratis yang bebas digunakan, XAMPP berfungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas Apache HTTP Server, *MySQL* database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl”.

Hidayatullah dan Kawistara (2015:127) menjelaskan bahwa kata XAMPP berasal dari:



1. X yang berarti *cross platform* karena XAMPP bisa dijalankan di Windows, Linux, Mac, dan sebagainya.
2. A yang berarti Apache sebagai *web sever*-nya.
3. M yang berarti *MySQL* sebagai *Database Management System* (DBMS)-nya.
4. PP yang berarti PHP dan Perl sebagai bahasa yang didukungnya.

2.4.2 Pengertian HTML (*Hypertext Markup Language*)

Asropudin (2013:44), “*Hypertext Markup Language* adalah bahasa komputer yang digunakan untuk membuat sebuah halaman *webpage*”.

Hidayatullah dan Kawistara (2015:13), “*Hypertext Markup Language* (HTML) adalah bahasa *standard* yang digunakan untuk menampilkan halaman *web*”.

Berdasarkan beberapa pengertian diatas dapat disimpulkan bahwa HTML adalah salah satu bahasa yang digunakan untuk pembuatan aplikasi yang di tampilkan pada halaman *web*.

2.4.3 Pengertian PHP (*Hypertext Preprocessor*)

Raharjo (2016:38), “PHP adalah suatu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*”.

Hidayatullah dan Kawistara (2015:231) mengatakan definisi PHP:

PHP adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP juga dapat diintegrasikan dengan HTML, JavaScript, JQuery, Ajax. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan *file* bertipe HTML.

Berdasarkan penjelasan diatas dapat disimpulkan PHP adalah suatu bahasa pemrograman yang berisikan *script* untuk membangun aplikasi *web* yang dijalankan dengan menggunakan *web server*.

2.4.3.1 Sintaks Dasar PHP

243.1.1 PHP *Opening* dan *Closing Tag*

Hidayatullah dan Kawistara (2015:241), “dalam HTML diharuskan menulis *opening* dan *closing tag* seperti `<html> </html>`, dan dalam PHP pun harus



melakukan *opening* dan *closing tag* sebagai awal dan akhir program PHP”.

Berikut contoh program dibawah ini:

```
<?php /*Opening Tag PHP */  
/* isi program PHP */  
/* Closing Tag PHP */ ?>
```

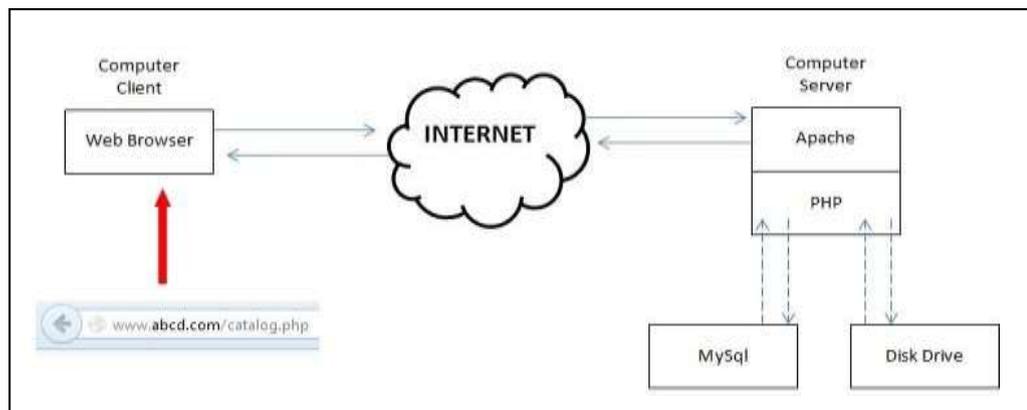
24312 Komentor dalam PHP

Hidayatullah dan Kawistara (2015:241), “PHP mendukung komentar seperti pada bahasa C, C++ dan Unix *Shell-style* (*Perl style*)”. Perhatikan contoh dibawah ini:

```
<?php  
echo ‘Ini adalah contoh’; //contoh gaya komentar satu baris  
  
/* Ini adalah contoh komentar  
lebih dari satu baris */  
echo ‘Contoh lagi’; #contoh gaya komentar satu baris pada shell  
?>
```

2.4.3.2 Cara Kerja PHP

Raharjo (2016:39) menjelaskan, adapun cara kerja aplikasi *web* yang ditulis dengan PHP dapat diilustrasikan dengan gambar dibawah ini:



Gambar 2.1 Cara Kerja PHP

(Sumber: <https://curhatphp.blogspot.co.id>)

Berikut ini adalah keterangan dari gambar diatas:

1. User menulis *www.abcd.com/catalog.php* kedalam *address bar* dari *web browser* (IE, Mozilla Firefox, Opera, dll).
2. *Web browser* mengirimkan pesan diatas ke computer server (*www.abcd.com*) melalui internet, meminta halaman *catalog.php*.
3. *Web server* (misalnya Apache), program yang berjalan di computer *server*, akan menangkap pesan tersebut, lalu meminta *interpreter* PHP (program lain yang juga berjalan di komputer *server*) untuk mencari file *catalog.php* dalam *disk drive*.
4. *Interpreter* PHP membaca file *catalog.php* dari *disk drive*.
5. *Interpreter* PHP akan menjalankan perintah-perintah atau kode PHP yang ada dalam file *catalog.php*. Jika kode dalam file *catalog.php* melibatkan akses terhadap *database* (misalnya MySQL) maka *interpreter* PHP juga akan berhubungan dengan MySQL untuk melaksanakan perintah-perintah yang berkaitan dengan *database*.
6. *Interpreter* PHP megirimkan halaman dalam bentuk HTML ke Apache.
7. Melalui internet, Apache mengirimkan halaman yang diperoleh dari *interpreter* PHP ke komputer *user* sebagai respon permintaan yang diberikan.
8. *Web browser* dalam komputer akan menampilkan halaman yang dikirim oleh Apache.

2.4.4 Pengertian MySQL (*My Structured Query Language*)

Hidayatullah dan Kawistara (2015, “MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi *web*.”



DBMS (*Database Management System*) adalah aplikasi yang dipakai untuk mengelola basis data”.

Dalam penggunaannya *MySQL* memiliki beberapa jenis tipe data. Diantaranya:

2.4.4.1 Tipe Data Numerik

MySQL menggunakan seluruh tipe data numeric standar ANSI. Berikut ini adalah tipe data numeric yang biasanya digunakan beserta penjelasannya.

Tabel 2.9 Tipe-Tipe Data Numerik

No.	Tipe Data	Deskripsi
1.	INT	Nilai integer yang bisa bertanda atau tidak. Jika bertanda, maka rentang yang diperbolehkan adalah -2147482648 sampai 2147483647, sedangkan jika tidak bertanda maka rentangnya dari 0 sampai 4294967295.
2.	TINYINT	Nilai integer yang sangat kecil. Rentangnya -128 sampai 127 untuk yang bertanda dan 0 sampai 255 untuk yang tidak bertanda.
3.	SMALLINT	Nilai integer yang sangat kecil dengan rentang -31768 sampai 32767 untuk yang bertanda sedangkan untuk yang tidak bertanda dari 0 sampai 65535
4.	MEDIUMINT	Integer dengan ukuran sedang dengan rentan -8388608 sampai 8388607 atau 0 sampai 16777215
5.	BIGINT	Integer dengan ukuran besar dengan rentang -9223372036854775808 sampai 9223372036854775807 atau 0 sampai 18446744073709551615
6.	FLOAT (M,D)	Bilangan pecahan dengan panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimalnya bisa sampai 24 digit. Defaultnya float (10,2). Bilangan float selalu bisa bertanda.



Lanjutan Tabel 2.9 Tipe-Tipe Data Numerik

7.	DOUBLE (M,D)	Bilangan pecahan dengan presisi dua kali lipat. Panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimalnya bisa sampai 53 digit. Defaultnya double (16,4). Bilangan double selalu bisa bertanda. Sinonim double adalah real.
8.	DECIMAL(M,D)	Adalah bilangan pecahan dan harus didefinisikan M dan D-nya. Setiap desimal membutuhkan 1byte. Sinonim dari desimal adalah numeric.

(Sumber: Hidayatullah dan Kawistara, 2015:181-182)

2.4.4.2 Tipe Data Tanggal dan Waktu

Berikut ini adalah tipe data tanggal dan waktu didalam *MySQL*:

Tabel 2.10 Tipe-Tipe Data Tanggal dan Waktu

No.	Tipe Data	Deskripsi
1.	DATE	Adalah tipe data tanggal dengan format YYYY-MM-DD, antara 1000-01-01 dan 9999-12-31. Contoh: 17 Agustus 1945 akan disimpan sebagai 1945-08-17.
2.	DATETIME	Adalah kombinasi tanggal dan waktu dengan format YYYY-MM-DD HH:MM:SS dan rentang data antara 1000-01-01 00:00:00 sampai dengan 9999-12-31 23:59:59. Contoh pukul 10:00 pagi pada tanggal 17 Agustus 1945 akan disimpan sebagai 1945-08-17 10:00:00.
3.	TIMESTAMP	Sebuah penanda waktu antara 1 Januari 1970 tengah malam sampai tahun 2037. Formatnya mirip datetime tetapi tanpa pembatas diantara angkanya. Contoh: pukul 10:00 pagi pada tanggal 17 Agustus 1945 akan disimpan sebagai 19450817100000.
4.	TIME	Menyimpan waktu dalam format HH:MM:SS. Contoh: pukul 10.00 pagi akan disimpan menjadi 10:00:00.

**Lanjutan Tabel 2.10** Tipe-Tipe Data Tanggal dan Waktu

5.	YEAR (M)	Menyimpan data tahun dalam format 2 atau 4 digit. Jika M diisi dengan nilai 2, maka rentang tahunnya dari 1970-2069 sedangkan jika M diisi dengan nilai 4 maka YEAR bisa bernilai 1901 sampai dengan 2155. Default nilai M adalah 4.
----	----------	--

Sumber: *Hidayatullah dan Kawistara, 2015:182-183*)

2.4.4.3 Tipe Data String

Berikut ini adalah tipe data string yang paling umum didalam *MySQL*:

Tabel 2.11 Tipe-Tipe Data String

No.	Tipe Data	Deskripsi
1.	CHAR (M)	String dengan ukuran tetap. Ukurannya antara 1-255 karakter. Ukuran ditentukan dengan nilai M. Contoh: Char(6)
2.	VARCHAR (M)	String dengan ukuran bervariasi antara 1 sampai dengan 255 karakter. Contoh: VARCHAR(25).
3.	TEXT	String dengan ukuran maksimum 65535 karakter. String yang tersimpan didalam text dianggap tidak <i>case sensitive</i> . Untuk kapasitas yang lebih kecil bisa menggunakan TINYTEXT dengan kapasitas maksimal 255 karakter. Sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMTEXT (maksimal 16777215 karakter) dan LONGTEXT (maksimal 4294967295 karakter).
4.	BLOB	<i>Binary Large Objects</i> adalah tipe data untuk menyimpan data binary dalam jumlah besar. Biasa digunakan untuk menyimpan citra. Untuk penyimpanan data yang lebih kecil bisa menggunakan TINYBLOB (maksimal 255 karakter) sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMBLOB (maksimal 16777215 karakter) dan LONGBLOB (maksimal 4294967295 karakter).



Lanjutan Tabel 2.11 Tipe-Tipe Data *String*

5.	ENUM	Enumerasi atau sebuah <i>list</i> (daftar). Jadi misalnya Anda ingin bahwa sebuah nilai terbatas hanya boleh dengan nilai tertentu saja maka anda bisa membuat sebuah daftar. Misalnya saja nilai itu hanya bisa terdiri dari A-E, maka Anda bisa membuatnya menjadi ENUM('A','B','C','D','E').
----	------	---

(Sumber:Hidayatullah dan Kawistara, 2015:182-183)