

BAB II

LANDASAN TEORI

2.1 Sistem

(Tata Sutabri, 2012) mendefinisikan Sistem adalah sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

Menurut Tata Sutabri sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu :

1. **Komponen sistem (*Component*)**
Susatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan.
2. **Batasan sistem (*Boundary*)**
Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.
3. **Lingkungan luar sistem (*Environment*)**
Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang memepengaruhi oprasi sistem tersebut disebut lingkungan luar sistem. lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.
4. **Penghubung sistem (*Interface*)**
Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain.
5. **Masukan sistem (*Input*)**
Energi yang dimasukkan kedalam sistem tersebut masukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).
6. **Keluaran sistem (*Output*)**
Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolahan sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan yang telah direncanakan.

2.2 Informasi

Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari tak berguna menjadi berguna bagi penerimanya. Nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan (Tata Sutabri, 2012).

2.3 Sistem Informasi

Sistem informasi adalah suatu sistem di 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 laporan-laporan yang diperlukan oleh pihak luar tertentu” Tata Sutabri (2012).

Dari pengertian di atas dapat ditarik kesimpulan bahwa sistem informasi adalah suatu sistem yang terdapat pada sebuah organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan merupakan kegiatan strategi dari suatu organisasi serta tersedianya pihak luar yang tertentu dengan laporan-laporan yang diperlukan.

2.4 Data

Menurut Rusdiana & Moch.Irfan (2014: 71) data adalah fakta atau bagian dari fakta yang mengandung arti sehubungan dengan kenyataan, simbol-simbol, gambar-gambar, angka-angka, huruf atau simbol yang menunjukkan suatu ide, objek, kondisi, atau situasi dan lainnya, yang didapatkan melalui suatu observasi atau secara data diartikan sebagai keterangan tentang sesuatu.

2.5 Konsep Dasar Informasi

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian, yaitu :

a. Informasi Strategis

Informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.

b. Informasi Taktis

Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.

c. Informasi Teknis

Informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stock, retur penjualan, dan laporan kas harian.

2.6 Metodologi Pengembangan Sistem

Menurut Tenia Wahyuningrum dan Dwi Januarita (2014) Rapid Application Development (RAD) adalah sebuah proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan dalam waktu yang singkat. RAD menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana working model (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (requirement) pengguna dan selanjutnya disingkirkan. Dalam pengembangan sistem informasi normal, memerlukan waktu minimal 180 hari, namun dengan menggunakan metode RAD, sistem dapat diselesaikan dalam waktu 30-90 hari. Model RAD memiliki 3 tahapan sebagai berikut:



1. Rencana Kebutuhan (Requirement Planning)

User dan analyst melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

2. Proses Desain Sistem (Design System)

Pada tahap ini keaktifan user yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara user dan analyst. Seorang user dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan user yang dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi software yang meliputi organisasi sistem secara umum, struktur data dan yang lain.

3. Implementasi (Implementation)

Tahapan ini adalah tahapan programmer yang mengembangkan desain suatu program yang telah disetujui oleh user dan analyst. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini user biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut.

2.7 Alat Bantu Perancangan Sistem

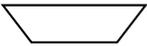
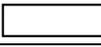
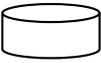
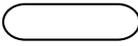
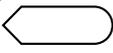
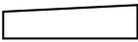
Alat Bantu perancangan sistem yang dapat digunakan adalah :

a) Bagan Alir Dokumen

Bagan alir dokumen (*Document flowchart*) merupakan bagan alir yang menunjukkan arus data dari laporan dan formulir termasuk tembusan-tembusannya. Simbol yang digunakan dalam bagan alir dokumen dapat dilihat pada tabel dibawah ini.

Tabel 2.1 Simbol-simbol Bagan Alir Dokumen

| Simbol | Keterangan |
|--------|------------|
|--------|------------|

| | |
|---|--|
| Dokumen  | Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan |
| Kegiatan manual  | Menunjukkan pekerjaan yang dilakukan tidak dengan program komputer. |
| Simpanan / arsip  | Menunjukkan dokumen yang diarsipkan (arsip manual) |
| Proses  | Menunjukkan kegiatan proses yang dilakukan dengan program komputer |
| Hard Disk  | Menunjukkan <i>input / output</i> menggunakan <i>hard disk</i> |
| Terminasi  | Menunjukkan awal atau akhir dari suatu proses. |
| Keputusan  | Menunjukkan suatu penyeleksian kondisi di dalam program. |
| Display  | Menunjukkan <i>output</i> yang ditampilkan di <i>monitor</i> |
| Pemasukan  | Menunjukkan <i>input data</i> secara manual melalui <i>keyboard</i> |
| Garis alir  | Menunjukkan arus dari data |
| Penjelasan  | Menunjukkan penjelasan dari suatu proses |
| Penghubung  | Menunjukkan penghubung kehalaman yang masih sama atau ke halaman lain. |
|  | Menunjukkan Barang, bias dalam bentuk baju, batik, dan lain-lain |
|  | Menunjukkan komunikasi menggunakan telepon |

b) UML (Unified Modeling Language)

UML (Unified Model Language) merupakan alat industri standar yang memungkinkan kita untuk mengkomunikasikan dengan jelas tentang kebutuhan, arsitektur dan desain . UML

merupakan salah satu alat yang paling berguna dan paling populer di dalam dunia pengembangan sistem. Hal ini disebabkan karena UML adalah sebuah metode pemodelan visual yang memungkinkan kepada pengembang untuk membuat sebuah blueprints yang dapat di mengerti, sehingga hasil blueprint ini dapat dimengerti antara satu dengan yang lainnya (Muhammad Taufik, 2015).

UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Sukanto dan Shalahuddin, 2013)

UML (Unified Modelling Language) menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori object-oriented dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar team programmer maupun dengan pengguna. Widodo dan Herlawati (2011), “UML diaplikasikan untuk maksud tertentu”, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Tabel 2.2 Tipe Diagram UML (*Unified Modelling Language*)

| No. | Diagram | Tujuan |
|-----|----------------|---|
| 1 | <i>Class</i> | Memperlihatkan himpunan kelas-kelas, antarmukaantarmuka, kolaborasi-kolaborasi, serta relasi-relasi |
| 2 | <i>Package</i> | Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen |

| | | |
|---|----------------------|--|
| 3 | <i>Use Case</i> | Diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas) |
| 4 | <i>Sequence</i> | Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu |
| 5 | <i>Communication</i> | Sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan |
| 6 | <i>Statechart</i> | Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas |
| 7 | <i>Activity</i> | Tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem |
| 8 | <i>Component</i> | Memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya |
| 9 | <i>Deployment</i> | Memperlihatkan konfigurasi saat aplikasi dijalankan (run-time) |

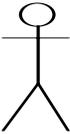
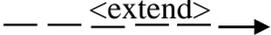
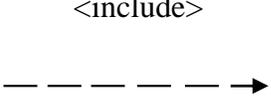
Jenis-jenis Diagram UML (*Unified Modelling Language*)

- Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. (Sukamto dan Shalahuddin, 2013).

Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.3 Simbol *Use Case* Diagram

| Simbol | Deskripsi |
|--|---|
| Use Case  | Fungsional yang disediakan sistem sebagai unit-unit yang paling saling bertukar pesan antar unit atau aktor: biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case |
| Aktor / actor  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang tapi aktor belum merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor |
| Asosiasi / association  | Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor |
| Ekstensi / extend  | Case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan |
| Generalisasi/generalization  | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi satu adalah fungsi yang lebih umum dari lainnya |
| Menggunakan/include/uses  | Fungsinya atau sebagai syarat dijalankan use case ini. |

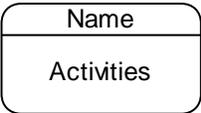
- Activity Diagram

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak". Yang perlu diperhatikan

disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Sukamto dan Shalahuddin,2013).

Adapun simbol-simbol yang digunakan dalam activity diagram adalah sebagai berikut:

Tabel 2.4 Simbol *Activity* Diagram

| Keterangan | Simbol | Deskripsi |
|-----------------|---|--|
| Status awal |  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas |  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan |  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan |  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| <i>Swimlane</i> |  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
| Status akhir |  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

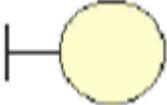
- Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek". Oleh karena

itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlihat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Sequence diagram menunjukkan urutan event kejadian dalam suatu waktu. Komponen sequence diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. (Sukanto dan Shalahuddin, 2013).

Berikut simbol-simbol yang digunakan dalam sequence diagram adalah:

Tabel 2.5 Simbol *Sequence* Diagram

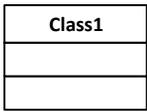
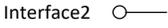
| No. | Symbol | Keterangan |
|-----|---|---|
| 1. |  | Menggambarkan orang yang berinteraksi dengan sistem |
| 2. |  | Menggambarkan hubungan kegiatan yang akan dilakukan |
| 3. |  | Menggambarkan sebuah penggambaran dari form |
| 4. |  | Menggambarkan penghubung antara boundary dengan tabel |

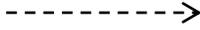
| | | |
|-----|--|--|
| 5. | Lifeline  | Menggambarkan tempat mulai dan berakhirnya sebuah pesan |
| 6. | Line message  | Menggambarkan pengiriman pesan |
| 7. | Object  | Menyatakan objek yang berinteraksi pesan |
| 8. | Garis hidup  | Menyatakan kehidupan suatu objek |
| 9. | Waktu aktif  | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya actor tidak memiliki waktu aktif |
| 10. | Stimulus  | Menyatakan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek lain |

- Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.(Sukamto dan Shalahuddin, 2013). Berikut simbol-simbol yang digunakan dalam class diagram adalah sebagai berikut:

Tabel 2.6 Simbol *Class* Diagram

| Simbol | Deskripsi |
|---|--|
| <p>Kelas</p>  | Kelas pada struktur sistem. |
| <p>Natarmuka/<i>interface</i></p>  | Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek. |
| <p>Asosiasi</p>  | Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| <p>Asosiasi berarah</p>  | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| <p>Generalisasi</p>  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus). |
| <p>Kebergantungan</p> | Relasi antar kelas dengan makna kebergantungan antar kelas. |

| | |
|---|---|
|  | |
| Agregasi  | Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>). |

2.8 Sistem Basis Data

Sistem basis data adalah kumpulan subsistem yang terdiri atas basis data dengan para pemakai yang menggunakan basis data secara bersama–sama, personal–personal yang merancang dan mengolah basis data, teknik–teknik untuk merancang dan mengelola basis data, serta sistem komputer untuk mendukungnya. Dari pengertian tersebut dapat disimpulkan bahwa sistem basis data mempunyai beberapa elemen penting yaitu :

- a) Basis data sebagai inti dari sistem basis data
- b) Perangkat lunak (*software*) untuk perancangan dan pengelolaan basis data
- c) Perangkat keras (*Hardware*) sebagai pendukung operasi pengolahan data
- d) Manusia (*Brainware*) yang mempunyai peran penting dalam sistem tersebut yaitu sebagai pemakai atau para *spesialis* informasi yang mempunyai fungsi sebagai perancangan atau pengelola

Istilah Dalam *Database*

Beberapa istilah dalam *database* yang sering dipakai antara lain :

- a) *Entity*
Entity adalah konsep informasi yang direkam, meliputi orang, kejadian dan tempat.
- b) *Atribut* atau *Field*
Atribut atau *Field* adalah sesuatu yang mewakili *entity*.
- c) *Data Value*
Data Value adalah informasi yang tersimpan dalam setiap *atribut*.
- d) *Record*

Record adalah kumpulan *atribut* yang saling berkaitan satu dengan yang lain dan menginformasikan suatu *entity* secara lengkap.

e) *File*

File adalah kumpulan *record* yang mempunyai panjang atribut yang sama tetapi berbeda data *valuanya*.

f) Basis data atau *Database*

Database adalah kumpulan *file* satu dengan *file* yang lainnya yang membentuk satu informasi sistem secara keseluruhan.

Field kunci memegang peranan yang sangat penting dalam pembuatan tabel yang berisi *entity* dan *relasinya*. *Field* kunci merupakan satu *field* atau satu *set field* yang terdapat dalam satu *file* yang merupakan kunci dan mewakili *record*. Kunci disini akan sangat penting apabila dalam program nanti terdapat fasilitas pencarian, karena *field* yang merupakan kunci akan menjadi penentu dalam pencarian program. *Field* kunci dibagi menjadi beberapa bagian yaitu :

a) Kunci Kandidat

Kunci Kandidat adalah satu *atribut* atau *field* yang mengidentifikasi secara unik dari suatu kejadian yang sifatnya khusus dari suatu *entity*.

b) Kunci *Primer*

Kunci *Primer* adalah kunci kandidat yang dipilih untuk mewakili setiap kejadian dari suatu *entity*.

c) Kunci Alternatif

Kunci Alternatif adalah kunci kandidat yang tidak dipakai sebagai kunci *primer*.

d) Kunci Tamu

Kunci Tamu adalah kunci *primer* yang ditempatkan pada *file* lain dan biasanya menunjukkan dan melengkapi suatu hubungan antara *file* satu dengan *file* yang lainnya.

Manfaat dari *database* sebagai berikut :

- a) Media permanen penyimpanan pengolahan data.
- b) Petunjuk dan penjelasan bagaimana hasil pengolahan data disimpan.
- c) Data yang tersimpan dapat diubah dan dihapus.

d) Rujukan pembuatan laporan

Sedangkan tujuan dari konsep *database* sebagai berikut :

a) Meminimumkan terjadinya pengulangan data yang sama (*redudancy data*).

b) Mencapai *interpendensi data*.

2.9 Internet

Menurut Sibero (2013:10) menjelaskan internet (*Interconnection Network*) adalah jaringan komputer yang menghubungkan antara jaringan secara global, *internet* dapat juga disebut jaringan dalam satu jaringan yang luas”. Sedangkan menurut Hidayatullah dan kawistara (2015:1) bahwa “*Internet* adalah jaringan global yang menghubungkan komputer-komputer di seluruh dunia. Dengan *internet* sebuah komputer bisa mengakses data yang terdapat pada komputer lain dibenua yang berbeda.

2.10 Perangkat Lunak Penunjang

Bahasa pemrograman yang menunjang dalam pembuatan *website* antara lain sebagai berikut:

a) Bahasa Pemrograman HTML (*hypertext Markup Language*)

HTML atau yang merupakan singkatan dari Hypertext Markup Language adalah bahasa standar pemrograman untuk membuat suatu website yang bisa diakses dengan internet. Dengan kata lain halaman website yang kita lihat dan kita baca disusun dengan menggunakan bahasa ini dan kemudian diterjemahkan oleh computer agar dapat dipahami oleh penggunanya.

HTML merupakan standar pembuatan website secara luas agar laman website dapat ditampilkan pada layar komputer. HTML disusun dengan kode dan simbol tertentu yang dimasukkan ke dalam sebuah file atau dokumen. Jadi setiap Anda membuka website apapun dengan menggunakan browser maka web tersebut dibuat dengan menggunakan HTML.

Dari singkatan Hypertext Markup Language itu sendiri kita bisa mengetahui makna dari HTML. Hypertext adalah suatu metode yang digunakan untuk berpindah laman web ke laman yang lain dengan mengklik suatu tulisan atau simbol pada laman website. Simbol atau tag HTML tersebut ditulis pada laman HTML yang sudah disediakan pada dashboard

website pada umumnya. Sementara bahasa dalam html adalah bahasa pemrograman atau script yang disusun dari tag-tag tertentu yang nantinya akan diterjemahkan kedalam teks atau visual yang dapat dilihat pada website. Muhammad Rizwan (2018).

Sumber: <https://www.nesabamedia.com/pengertian-dan-fungsi-html/>

b) *PHP*

Menurut Hartoto (2016:95) PHP adalah bahasa scripting yang menyatu dengan HTML dan berada di server (server side HTML embedded scripting). Perintah - perintah yang kita masukkan akan sepenuhnya dijalankan dan dikerjakan di server.

c) *MySql*

Menurut Hendry (2015:25) MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis di bawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

d) *Database*

Menurut Ario Suryo Kusumo (2016:55) Database adalah kumpulan data yang disimpan dalam media elektronik, saling berhubungan, diorganisasikan dengan baik agar tidak terjadi redundansi/ pengulangan dan inkonsistensi, serta nantinya dapat dimanfaatkan kembali. Untuk mengelola database digunakan DBMS (Data Management System), yaitu perangkat lunak untuk melakukan manajemen basis data dan berinteraksi dengan aplikasi pengguna. Dalam aplikasi, DBMS merupakan backend (database yang mendukung aplikasi).

e) *Xampp*

Menurut Sarwandi (2016:19) XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kumpulan yang tersusun secara teratur dari beberapa program. Fungsinya sebagai server yang berdiri sendiri (localhost) yang terdiri atas program Apache HTTP Server, MYSQL database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Program ini tersedia dalam GNU (General Public License) dan bebas, merupakan web server yang mudah digunakan yang dapat melakukan tampilan halaman web yang dinamis.

f) Adobe Dreamweaver

Menurut Wahana Komputer (2012:2) Adobe Dreamweaver merupakan program aplikasi yang digunakan untuk melakukan desain halaman website secara visual. Dalam mendesain halaman website aplikasi ini menyediakan fasilitas-fasilitas yang memberikan kemudahan bagi para pengembang web.