



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Pangera *et.al* (2010:3), “Komputer adalah perangkat elektronik, beroperasi dibawah perintah pengendali yang disimpan pada memori komputer, dapat menerima data, memproses data berdasarkan aturan tertentu, mencetak hasilnya, dan menyimpan data untuk penggunaan dimasa depan”.

Dhanta (2009:11), “Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima input, memproses input sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dan hasil pengolahannya, serta menyediakan output dalam bentuk informasi”.

2.1.2. Pengertian Sistem

Sutabri (2012:6), “Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama – sama untuk mencapai tujuan tertentu”.

Mulyanto (2009:1), “Sistem adalah kumpulan dari elemen – elemen yang berinteraksi untuk mencapai suatu tujuan tertentu sebagai satu kesatuan”.

2.1.3. Karakteristik Sistem

Sutabri (2012:13), “model umum sebuah sistem terdiri dari *input*, *proses*, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem”.



Adapun karakteristik yang dimaksud adalah sebagai berikut:

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerjasama membentuk satu kesatuan. Komponen – komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat – sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan *Supra* sistem.

b. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah – pisahkan.

c. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

d. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain tersebut dengan penghubung sistem atau I. Penghubung ini memungkinkan sumber – sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu *integrasi* sistem yang membentuk satu kesatuan.



e. Masukkan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, "program" adalah *maintenance input* yang digunakan untuk mengoperasikan *computer*. Sementara "data" adalah *signal input* yang akan diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, di mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal – hal lain yang merupakan input bagi subsistem lainnya.

g. Pengolah Sistem (*Process*)

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

h. Sasaran Sistem (*Objective*)

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

2.1.4. Tahap – Tahap Pengembangan Sistem

Sutabri (2012:59), "fase pengembangan sistem informasi disebut juga sebagai siklus hidup pengembangan sistem informasi yang garis besarnya terdiri dari lima langkah. Tahapan-tahapan pekerjaan dalam pelaksanaan tidak kaku namun dapat disesuaikan dengan kebutuhan, seperti cara interaksi".



Tahapan utama dalam proses pengembangan sistem informasi adalah sebagai berikut:

1. Investigasi Sistem

Manfaat dari *fase* penyelidikan ini adalah untuk menentukan *problem-problem* atau kebutuhan yang timbul. Hal itu memerlukan pengembangan sistem secara menyeluruh atau ada usaha lain yang dapat dilakukan untuk memecahkannya. Dengan kata lain sistem yang ada tetap berjalan tanpa perlu perubahan maupun pembangunan sistem yang baru. Hal itu dapat terjadi karena kebutuhan itu tidak dapat diimplementasikan atau ditangguhkan pelaksanaannya untuk suatu kurun waktu tertentu. *Alternatif* lainnya mungkin hanya diperlukan perbaikan-perbaikan pada sistem tanpa harus menggantinya.

2. Analisis Sistem

Tahap analisis bertitik tolak pada kegiatan – kegiatan dan tugas – tugas dimana sistem yang berjalan dipelajari lebih mendalam, konsepsi dan usulan dibuat untuk menjadi landasan bagi sistem yang akan dibangun. Salah satu tujuan terpenting dari tahap ini adalah untuk mendefinisikan sistem yang berjalan. Pemakai sistem dan analisis sistem bekerja sama untuk menjabarkan kebutuhan dan kemampuan dari sistem yang baru yang akan diusulkan.

3. Desain Sistem

Pada tahap ini sebagian besar kegiatan yang berorientasi ke komputer dilaksanakan. *Spesifikasi perangkat keras* dan *perangkat lunak* yang telah disusun pada tahap sebelumnya ditinjau kembali dan disempurnakan. Rencana pembuatan program dilaksanakan dan juga *testing* programnya. Latihan bagi para pemakai sistem dimulai. Pada akhirnya dengan berpartisipasi penuh dari pemakai sistem, dilakukan test sistem secara menyeluruh. Apabila pemakai sistem telah puas melihat hasil *testing* yang dilakukan maka dilanjutkan ke tahap selanjutnya.

4. Implementasi Sistem

Tahap ini adalah prosedur yang dilakukan untuk menyelesaikan desain sistem yang disetujui dan menguji, meng-*install*, dan memulai penggunaan sistem baru atau sistem yang diperbaiki. Tujuan tahap implementasi ini adalah untuk



menyelesaikan desain sistem yang sudah disetujui, menguji serta mendokumentasikan program – program dan prosedur sistem yang diperlukan, memastikan bahwa personil yang terlibat dapat mengoperasikan sistem baru dan memastikan bahwa konversi sistem lama ke sistem yang baru dapat berjalan secara baik dan benar.

5. Pemeliharaan Sistem

Disarankan adanya dua tahap *review* yang harus dilaksanakan. Pertama kali tidak terlalu lama setelah penerapan sistem, dimana tim proyek masih ada dan masing – masing anggota masih memiliki ingatan yang segar atas sistem yang mereka buat. *Review* berikutnya dapat dilakukan kira – kira setelah enam bulan berjalan. Tujuannya adalah untuk menyakinkan apakah sistem tersebut berjalan sesuai dengan tujuan semula dan apakah masih ada perbaikan atau penyempurnaan yang harus dilakukan. Selain itu, tahap ini juga merupakan bentuk evaluasi untuk memantau supaya sistem informasi yang dioperasikan dapat berjalan secara optimal dan sesuai dengan harapan pemakai maupun organisasi yang menggunakan sistem tersebut. Selanjutnya setiap tahun, organisasi tersebut menggunakan 10% – 25% dari biaya sistem awal untuk memelihara sistem tersebut. Tujuan dari proses pemeliharaan sistem ini adalah untuk melakukan evaluasi sistem secara cepat dan efisien, menyempurnakan proses pemeliharaan sistem dengan selalu menganalisis kebutuhan informasi yang dihasilkan sistem tersebut dan meminimalkan gangguan kontrol dan gangguan operasi yang disebabkan oleh proses pemeliharaan sistem.

2.2. Teori Khusus

2.2.1. Pengertian Database

Shalahudin *et.al* (2013:43), “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Mulyanto (2009:194), “*Database* adalah sebuah *file* yang mengkoordinasi *file – file* data yang saling berhubungan dan memiliki kepentingan yang sama sehingga akan mempermudah pengolahan data”.



Sutarman (2009:15), “Database adalah sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan record – record yang menyimpan data dan hubungan di antaranya”.

2.2.2. Kamus Data (*Data Dictionary*)

Shalahudin *et.al* (2013:73), “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 adalah elemen – elemen atau simbol – simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau *file* didalam system. 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 2.1 Simbol – Simbol pada Kamus Data

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

Sumber:Shalahuddin *et.al* (2013:73)

2.2.3. Diagram Konteks

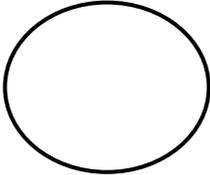
Kristanto (2008:70), “Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entiti luar, masukan, dan keluaran sistem”.



“Diagram konteks adalah suatu diagram alir yang tingkat tinggi yang menggambarkan seluruh *jarangan*, masukan, dan keluaran”. Untuk menggambarkan sistem yang sedang berjalan. Diagram ini merupakan gambaran umum sistem yang nantinya akan kita buat, secara uraian dapat dikatakan bahwa diagram konteks itu berisi siapa saja yang memberikan data (input) ke sistem serta kepada siapa data informasi yang harus dihasilkan sistem.

Simbol-simbol yang terdapat dalam diagram konteks, sebagai berikut:

Tabel 2.2. Simbol – Simbol pada Diagram Konteks

No.	Simbol	Keterangan
1.		Entitas adalah Pihak – pihak yang berada diluar sistem, tetapi secara langsung berhubungan dengan sistem dalam hal memberi data atau menerima informasi.
2.		Proses merupakan isi mengenai sistem yang akan dibuat.
3.		Berisi data atau informasi yang mengalir dari satu pihak ke sistem dan sebaliknya.

2.2.4. Data Flow Diagram (DFD)

Shalahudin *et.al* (2013:70), “Data Flow Diagram (*DFD*) adalah representasi grafik yang menggambarkan *aliran* informasi dan *transformasi* informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

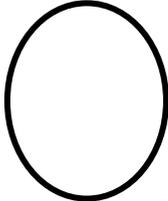


Beberapa petunjuk yang dapat dipakai dalam proses pembuatan simbol *DFD*. Petunjuk-petunjuk itu adalah sebagai berikut:

1. Penamaan yang jelas
 - a. Sebaiknya menggunakan nama yang mengacu pada fungsi, yaitu gabungan antara kata kerja yang *spesifik*.
 - b. Jangan menggunakan nama terlalu umum.
 - c. Gunakan nama yang mudah dimengerti oleh pemakai.
2. Memberi nomor pada proses

Nomor yang diberikan pada proses tidak harus menjadi nomor urut. Penomoran dimaksudkan sebagai identifikasi proses dan memudahkan penurunan (level yang lebih rendah) ke proses berikutnya.
3. Penggambaran kembali
 - a. Ukuran dan bentuk lingkaran tetap sama.
 - b. Panah yang melengkung dan lurus tidak menjadi masalah.
4. Hindarilah proses yang mempunyai masukan tetapi tidak mempunyai keluaran begitu pula sebaliknya, hindarilah proses yang mempunyai keluaran tetapi tidak mempunyai masukan.
5. Hati – hati dengan aliran data dan proses yang tidak dinamai. Aliran proses yang tidak diberi nama dapat mengakibatkan elemen data yang saling tidak berhubungan akan menjadi satu.

Tabel 2.3. Simbol – Simbol pada Data Flow Diagram

No	Simbol	Keterangan
1.		Proses atau fungsi prosedur, pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur didalam kode program.

Lanjutan **Tabel 2.3.** Simbol – Simbol pada Data Flow Diagram

2.		<p><i>Field</i> atau basisdata atau penyimpanan pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi yang harus dibuat menjadi tabel basisdata yang dibutuhkan, tabel harus sesuai dengan perancangan tabel pada basisdata <i>Entity Relationship Diagram</i>, <i>Conceptual Data Model</i>, <i>Physical Data Model</i>.</p>
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>
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>

Sumber: Shalahudin *et.al* (2013:70)

Aturan – aturan dalam pembuatan Data Flow Diagram adalah sebagai berikut:

- a. Di dalam data flow diagram tidak boleh menghubungkan antara *entity* luar (*external entity*) dengan *entity* luar (*external entity*) lainnya secara langsung.
- b. Di dalam data flow diagram tidak boleh menghubungkan *data store* yang satu dengan *data store* yang lainnya secara langsung.
- c. Di dalam data flow diagram tidak boleh menghubungkan *data store* dengan *entity* luar secara langsung.
- d. Setiap proses harus ada memiliki data flow yang masuk dan ada juga data flow yang keluar.



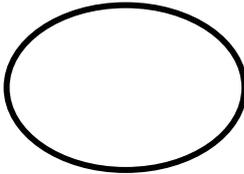
2.2.5. Entity Relationship Diagram (ERD)

Shalahuddin *et.al* (2013:50), “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”.

Entity Relationship Diagram (*ERD*) adalah suatu model jaringan berisi komponen – komponen himpunan *entitas* dan himpunan *relasi* yang masing – masing dilengkapi dengan *atribut – atribut* yang mempresentasikan seluruh fakta dari “Dunia nyata” (*Real World*) yang kita tinjau, dapat digambarkan secara sistematis dengan menggunakan diagram Entity-Relationship (*diagram E-R*).

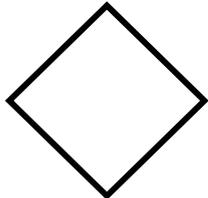
Notasi-notasi simbolik didalam diagram *E–R* yang dapat digunakan, adalah sebagai berikut:

Tabel 2.4. Simbol – Simbol pada Entity Relationship Diagram

No	Simbol	Keterangan
1.		<i>Entitas</i> adalah sebutan lain dari <i>record</i> dan <i>entities</i> atau <i>set entity</i> adalah sebutan lain dari <i>file</i> . <i>Entitas</i> adalah <i>objek</i> yang harus ada disuatu unit usaha yang akan dibuat komputersasinya, atau <i>entitas</i> adalah sebuah <i>objek</i> yang unik yang bisa dibedakan antara satu <i>objek</i> dengan <i>objek</i> lainnya.
2.		<i>Atribut</i> adalah suatu karakteristik yang biasa untuk menggambarkan seluruh atau sebagian dari <i>record</i> .



Lanjutan Tabel 2.4. Simbol – Simbol pada Entity Relationship Diagram

3.		<i>Relationship</i> keterhubungan atau keterkaitan antara satu <i>entitas</i> dengan satu atau lebih <i>entitas</i> lain.
4.		Baris sebagai penghubung antara himpunan, <i>relationship</i> dan himpunan <i>entitas</i> dengan atributnya.

Sumber:Shalahuddin *et.al* (2013:51)

2.2.6. Flowchart

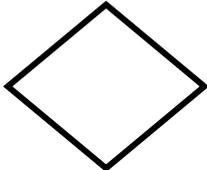
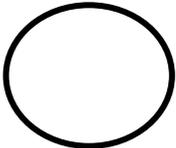
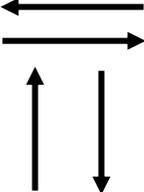
Siallagan (2009:6), *Flowchart* adalah suatu diagram alir yang mempergunakan simbol atau tanda untuk menyelesaikan masalah.

eWolf Community (2012:16), *Flowchart* adalah simbol – simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. *Flowchart* digunakan untuk mempermudah penyusunan program. Adapun simbol-simbol dalam *flowchart* adalah sebagai berikut:

Tabel 2.5. Simbol – Simbol pada *Flowchart*

No	Simbol	Keterangan
1.		Simbol <i>Start</i> atau <i>End</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.

Lanjutan Tabel 2.5. Simbol – Simbol pada *Flowchart*

3.		Simbol Input/Output yang mendefinisikan masukan dan keluaran proses.
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.
5.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama.
6.		Keputusan artinya menunjukkan suatu perbandingan yang harus dibuat bila hasilnya “ya”, maka arah alir akan menunjukkan ke suatu tempat, bila “tidak” akan menunjukkan ketempat lain.
7.		Penghubung artinya simbol untuk keluar atau masuk proses dalam lembar atau halaman yang sama.
8.		<i>Off – Line Connector</i> artinya simbol untuk keluar atau masuk proses dalam lembar halaman yang sama.

Lanjutan Tabel 2.5. Simbol – Simbol pada *Flowchart*

9.		Simbol yang mendefinisikan proses yang dilakukan secara manual.
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen.
11.		Simbol yang menyatakan bagian dari program (<i>sub-program</i>).
12.		Simbol masukan atau keluaran dari atau ke sebuah <i>pita magnetic</i> .
13.		Simbol <i>database</i> atau basis data.

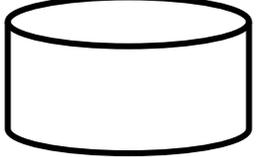
Sumber: Community (2012:17)

2.2.7. Blockchart

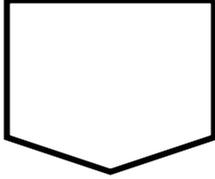
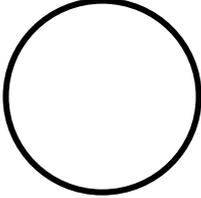
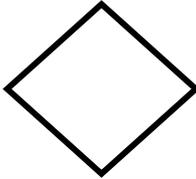
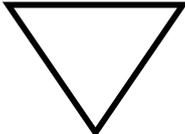
Kristanto (2008:75), “*Blockchart* adalah suatu model logika data yang berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”.

Adapun simbol-simbol yang sering digunakan dalam *blockchart*, yaitu:

Tabel 2.6. Simbol – Simbol pada *Blockchart*

No.	Simbol	Keterangan
1.		Suatu simbol yang digunakan dalam menandakan suatu dokumen, bisa dalam bentuk surat, formulir, buku/benda/berkas atau cetakan.
2.		Suatu simbol yang digunakan dalam menandakan Multi Dokumen, bisa dalam bentuk surat, formulir, buku/benda/berkas atau cetakan.
3.		Suatu simbol yang digunakan dalam bentuk proses yang dilakukan secara manual.
4.		Suatu simbol yang digunakan dalam proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
5.		Suatu simbol yang digunakan dalam data penyimpanan (<i>data storage</i>).
6.		Suatu proses yang dilakukan dengan menggunakan komputer.

Lanjutan **Tabel 2.6.** Simbol – Simbol pada *Blockchart*

7.		Suatu simbol yang digunakan dalam terminasi yang mewakili simbol-simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
8.		Suatu simbol yang digunakan dalam terminasi yang mewakili simbol-simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
9.		Suatu simbol yang digunakan dalam pengambilan keputusan.
10.		Suatu simbol yang digunakan dalam pemasukan data secara manual.
11.		Suatu simbol yang menandakan dokumen yang diarsipkan (arsip manual).
12.		Suatu simbol terminasi yang menandakan awal dan akhir dari suatu aliran.
13.		Suatu simbol yang diartikan sebagai layar peraga (<i>monitor</i>).

Sumber : Kristanto (2008:75)



2.3. Pengertian – Pengertian Judul

2.3.1. Pengertian Aplikasi

Hendrayudi (2009:143), “Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus)”.

Sutabri (2012:147), “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya”.

2.3.2. Pengertian Data

Sutarman (2009:3), “Data adalah fakta dari sesuatu pernyataan yang berasal dari kenyataan, dimana pernyataan tersebut merupakan hasil pengukuran atau pengamatan”.

2.3.3. Pengertian Pengolahan Data

Sutarman (2009:4), “Pengolahan Data adalah proses perhitungan/transformasi data input menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan.

2.3.4. Pengertian Pasien

Kamus Besar Bahasa Indonesia Kontemporer, “Pasien adalah orang sakit yang dirawat oleh dokter di rumah sakit, klinik, dan sebagainya”.

2.3.5. Pengertian Rehab

Kamus Besar Bahasa Indonesia Kontemporer, “Rehab (Rehabilitasi) adalah pemulihan kepada keadaan atau keadaan yang semula”.

Kamus Besar Bahasa Indonesia Kontemporer, “Rehab (Rehabilitasi) adalah perbaikan kesehatan fisik, mental, dan moral penjahat, penyandang cacat, dan sebagainya melalui pengobatan dan pendidikan”.



2.3.6. Pengertian Medik

Kamus Besar Bahasa Indonesia Kontemporer, “Medik adalah dokter, mahasiswa ke dokteran, anggota korps kesehatan militer ia mengabdikan diri pada negaranya sebagai seorang medik di medan perang”.

Kamus Besar Bahasa Indonesia Kontemporer, “Medis (Medik) adalah yang berkenaan dengan pengobatan dalam kedokteran. Ia bekerja sebagai tenaga medis di sebuah rumah sakit”.

2.3.7. Pengertian Yayasan Pembinaan Anak Cacat Palembang

Yayasan Pembinaan Anak Cacat Palembang adalah yayasan bercirikan kerelawanan yang bersifat mengobati, mendidik, dan membimbing anak-anak penyandang disabilitas atau anak – anak berkebutuhan khusus agar dapat kembali ke masyarakat dengan penuh percaya diri dengan bekal keterampilan yang dimilikinya.

2.3.8. Pengertian Aplikasi Pengolahan Data Pasien Rehab Medik pada Yayasan Pembinaan Anak Cacat Palembang

Aplikasi pengolahan data pasien rehab medik pada Yayasan Pembinaan Anak Cacat Palembang adalah suatu perangkat lunak komputer yang dibangun untuk mengolah data pasien rehab medik yang dilakukan oleh Yayasan Pembinaan Anak Cacat Palembang.

2.4. Teori Program

2.4.1. Pengenalan PHP

2.4.1.1. Pengertian PHP

Prasetio (2014:122), *PHP* (Hypertext Preprocessor) adalah bahasa *script* yang ditanam diisisi *server*.

Proses *PHP* dijalankan di *server* (*Windows/Linux*). Sebuah halaman dibuka dan mengandung kode *PHP*, *prosesor* itu akan menerjemahkan dan mengeksekusi semua perintah dalam halaman tersebut, dan kemudian menampilkan hasilnya ke browser sebagai halaman *HTML* biasa. Karena



penerjemahan ini terjadi di *server*, sebuah halaman ditulis dengan *PHP* dapat dilihat dengan menggunakan semua jenis *browser*, di sistem operasi apapun.

Seperti sebagian besar bahasa *script* lainnya, *PHP* dapat ditanamkan langsung kedalam *HTML*. Kode *PHP* dipisahkan dari *HTML* dengan menggunakan tanda *Start* dan *End*. Ketika sebuah dokumen di baca, *prosesor PHP* hanya menerjemahkan area yang ditandai saja, dan menampilkan hasilnya pada tempat yang sama.

PHP juga memiliki kemampuan untuk hampir sepenuhnya memisahkan kode dari *HTML*. Untuk proyek bersama yang lebih besar, metode ini sangat *ideal* karena memungkinkan *desainer* untuk mengerjakan *layout* halaman tanpa mempengaruhi kode *PHP*nya.

Prasetio (2014:129), Kode-kode *PHP* disisipkan di antara kode-kode *HTML*. Sebagai akibatnya, *PHP* dan *HTML* akan sama-sama di tulis dalam teks biasa. Kode *PHP* (misalnya dalam contoh di bawah ini adalah halaman yang menampilkan kata-kata "Anda berada di situs *Prothelon!*") akan berada disela-sela kode-kode dalam sebuah *file HTML* yang *berekstensi.php*, bukan *.htm* atau *.html* seperti biasanya.

Contoh kode *script PHP* yang dapat diletakkan pada kode *HTML* dengan menggunakan tag *<?php>*

2.4.1.2. Cara Kerja *PHP*

Prasetio (2014:127), Cara menggunakan *PHP* sangat mudah. Pada prinsipnya Anda hanya perlu menyisipkan kode *PHP* ke dalam tag-tag *HTML* yang sudah ada di situs Anda. Cara bekerjanya secara singkat adalah seperti ini. Ketika ada yang mengakses *web* Anda di halaman yang berisi kode *PHP* (tentunya dengan *file* *berekstensi.php*), *server* Anda akan mengeksekusinya dan kemudian mengirimkan hasil eksekusinya ke *web server* untuk selanjutnya ditampilkan menggunakan kode *HTML*.



2.4.1.3. Cara Penulisan Kode *PHP*

Prasetio (2014:131), Aturan-aturan dasar penulisan sintaks *PHP*:

1. Penamaan *File PHP* *File PHP* Anda harus disimpan dengan ekstensi.php.
2. Komentar *PHP*

Komentar adalah bagian penting dalam kode *PHP* yang Anda buat. Prasetio (2014:132) menyebutkan bahwa komentar dapat dimanfaatkan untuk membantu menelusuri *error*. Dengan mengubah sebagian blok kode menjadi komentar, kita dapat melakukan proseriterasi untuk melakukan isolasi dan menemukan masalah pada *script PHP* yang mengalami *error*.

3. Permulaan Kode *PHP*

Blok kode *PHP* diawali dengan “<?php” (atau cukup disingkat “<?” saja bila server anda mengizinkan dan biasanya bisa).

4. Akhir kode *PHP*

Blok kode *PHP* ditutup dengan menambahkan “?>” diakhir blok kodenya.

5. Akhir Baris Program *PHP*

Setiap baris instruksi program diakhiri dengan tanda titik koma “;”. Artinya walaupun Anda menuliskannya lebih dari 1 baris tetap akan dianggap satu baris instruksi program jika belum ada tanda titik koma.

6. Tanda Kurung

Tanda Kurung akan banyak anda gunakan dalam kode *PHP*. Salah satu penggunaan yang sering dilakukan adalah dalam memanggil fungsi. “print” adalah nama fungsi dan informasi lain yang perlu ditambahkan pada fungsi dan dituliskan di dalam kurung.



```
<html>
<head>
<title> Contoh Halaman PHP </title>
</head>
<body>
<font color="blue">PHP kode saya akan membuat halaman ini
menampilkan:</font>
</p>
<?php
Print ("Anda berada di situs Prothelon!");
?>
</body>
</html>
```

Gambar 2.1. Contoh kode *script PHP*

2.4.2. Pengenalan XAMPP

2.4.2.1. Pengertian XAMPP

Nugroho (2013:1), *XAMPP* adalah paket program berbasis *web*, di dalamnya berisi: *Software Apache*, *PHP* dan *database MySQL*. *XAMPP* adalah paket *web programming*, kita dapat menggunakan *database MySQL Server* untuk belajar Pemrograman Visual, juga disana telah tersedia *tools phpMyAdmin* yang hanya berjalan di sisi *server web* untuk mengakses *database MySQL*.

Prasetio (2014:13), *XAMPP* merupakan singkatan dari *X* (empat *system operasi* apapun), *Apache*, *MySQL*, *PHP*, *Perl*. *XAMPP* merupakan *tools* yang menyediakan paket perangkat lunak ke dalam satu buah paket.

2.4.2.2. Memahami Folder XAMPP

Dibawah folder utama *xampp*, terdapat beberapa folder penting yang perlu diketahui. Penjelasan fungsinya sebagai berikut:

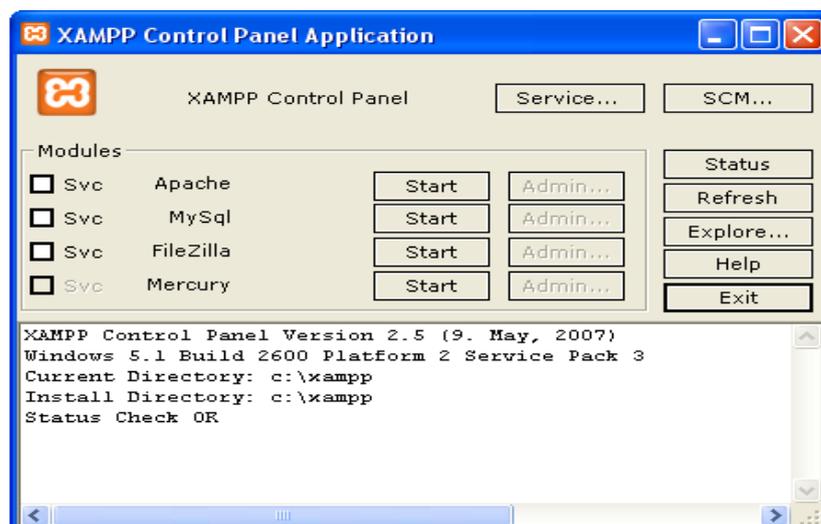
- a. *Apache*: Folder utama dari *Apache Webserver*.



- b. *Htdocs*: Folder utama untuk menyimpan data-data latihan *web*, baik *PHP* maupun *HTML* biasa. Pada folder ini, anda dapat membuat subfolder sendiri untuk mengelompokkan file latihannya. Semua folder dan file program di *htdocs* bisa diakses dengan mengetikkan alamat `http://localhost/` di *browser*.
- c. *Manual*: Berisi subfolder yang di dalam terdapat manual program dan *database*, termasuk manual *PHP* dan *MYSQL*.
- d. *Mysql*: Folder utama untuk *database MYSQL* server. Di dalamnya terdapat subfolder data (lengkapannya: `C:\xampp\mysql\data`) untuk merekam semua nama *database*, serta subfolder *bin* yang berisi *tools* klien dan server *MYSQL*.
- e. *php*: Folder utama untuk program *PHP*.

2.4.2.3. Pengoperasian XAMPP

Nugroho (2013:17), Klik *start/stop* untuk mengaktifkan /deaktif sebuah *server*. Klik Start disebelah *MySQL* untuk mengaktifkan *server database MySQL*. Anda bisa menutup *XAMPP control panel* tersebut tanpa mematikan aplikasi *server*nya dengan cara mengklik tanda silang (X) di sudut kanan. Untuk mematikan seluruh *server* klik *Exit*. Anda bisa mengetahui aplikasi *server* masih hidup dari tanda *XAMPP* berwarna oranye pada *Taskbar*.



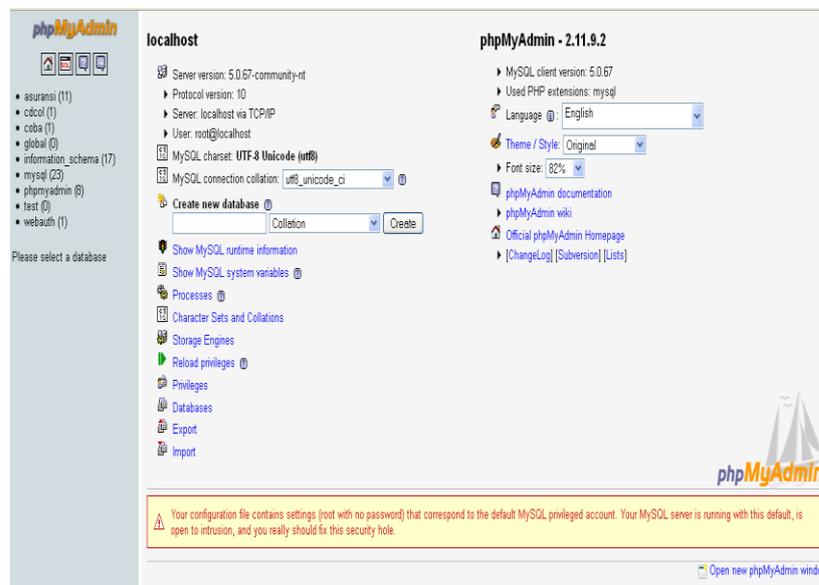
Gambar 2.2. XAMPP control panel application



2.4.3. PHPMYAdmin

Prasetio (2014:48), *PHPMYAdmin* adalah *tools* yang paling umum digunakan nantinya pada saat akan melakukan pembuatan dan perubahan *database* di *server* yang sebenarnya.

Prasetio (2014:48), *PHPMYAdmin* adalah *tools* berbasis *web* yang berguna untuk mengelola *database MySQL*. *XAMPP* memiliki *tools PHPMYAdmin* yang ada dalam paket *XAMPP* untuk mengelola *database* yang ada di *server*. Tujuan dibuatnya program ini adalah untuk membantu dalam pembuatan dan pengelolaan *database MySQL*.



Gambar 2.3. *PHP MyAdmin*