



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Metode Pengembangan Sistem

Sukanto dan Shalahuddin (2016:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak ini perlu didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur sistem, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan sistem dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak ini yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan kode program

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

d. Pengujian

Pengujian fokus pada perangkat lunak dari segi logika 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.

e. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung dan pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk perangkat lunak yang baru.

2.1.1.1 Analisis PIECES

Menurut Hanif (2007:51), “Analisis PIECES (*Performance, Information, Economy, control, Efficiency, dan Service*) merupakan teknik untuk mengidentifikasi dan memecahkan permasalahan yang terjadi pada sistem informasi”. Dari analisis ini akan menghasilkan identifikasi masalah utama dari suatu sistem serta memberikan solusi dari permasalahan tersebut. Analisis PIECES terdiri dari:

1. Analisis Kinerja (*Performance*)

Adalah kemampuan menyelesaikan tugas pelayanan dengan cepat sehingga sasaran atau tujuan segera tercapai. Kinerja diukur dengan jumlah produksi (*Troughput*) dan waktu tanggap (*Respon Time*) dari suatu sistem.

Jumlah Produksi adalah jumlah pekerjaan yang bias diselesaikan selama jangka waktu tertentu. Sedangkan waktu tanggap adalah waktu transaksi yang terjadi dalam proses kinerja.

2. Analisis Informasi (*Information*)

Adalah evaluasi kemampuan sistem informasi dalam menghasilkan nilai atau produk yang bermanfaat untuk menyikapi peluang dalam menangani masalah yang muncul. Situasi dalam analisis informasi ini meliputi:

- a. Akurasi, informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan.



- b. Relevan, informasi tersebut memiliki manfaat bagi pihak pemakai maupun pihak pengelola. Dimana relevansi setiap orang berbeda satu dengan yang lainnya.
3. Analisis Ekonomi (*Economy*)
 Adalah Penilaian sistem atas biaya dan keuntungan yang akan didapatkan dari sistem yang diterapkan. Sistem ini akan memberikan penghematan operasional dan keuntungan bagi instansi atau perusahaan. Hal yang diperlukan dalam analisis ini meliputi biaya dan keuntungan.
4. Analisis Keamanan (*Controlling*)
 Adalah Sistem keamanan yang digunakan harus dapat mengamankan data dari kerusakan, misalnya dengan *memback up data*. Selain itu sistem keamanan juga harus dapat mengamankan data dari akses yang tidak diizinkan. Analisis ini meliputi pengawasan dan pengendalian.
5. Analisis Efisiensi (*Efficiency*)
 Adalah sumber daya yang ada guna meminimalkan pemborosan. Efisiensi dari sistem yang dikembangkan adalah pemakaian secara maksimal terhadap sumberdaya infrastruktur, dan sumberdaya manusia. Serta efisiensi juga menganalisis keterlambatan pengolahan data yang terjadi.
6. Layanan (*Service*)
 Adalah mengkoordinasikan aktifitas dalam pelayanan yang ingin dicapai sehingga tujuan dan sasaran pelayanan dapat capai.

2.2 Teori Khusus

2.2.1 Pengertian DFD (*Data Flow Diagram*)

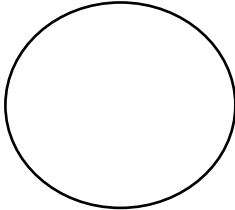
Kristanto (2008:61), “DFD (*Data Flow Diagram*) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.” Sedangkan menurut Rosa dan Shalahuddin (2016:71) menjelaskan, “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 (ouput)”.

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

Tabel 2.1 Simbol-simbol *Data Flow Diagram*

Notasi	Keterangan
	Proses atau fungsi atau prosedur, pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman 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
	File basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harusnya sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)) Catatan : Nama yang diberikan pada sebuah penyimpanan biasanya kata benda

Lanjutan Tabel 2.1 Simbol-simbol *Data Flow Diagram*

Notasi	Keterangan
	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 Catatan : Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda
	Aliran data: merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses masukan (<i>input</i>) atau keluaran (<i>output</i>) Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”

(Sumber : Sukamto dan Shalahuddin, 2016:71-72)

Sukamto dan Shalahuddin (2016:72-73) menjelaskan bahwa berikut ini adalah tahapan-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 kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

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

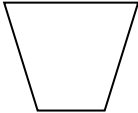
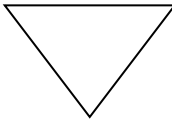
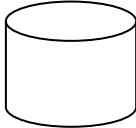
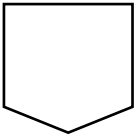
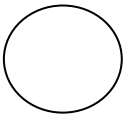
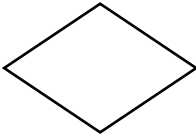
2.2.2 Blockchart

Kristanto (2008:75) menjelaskan, ”*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”. Adapun simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

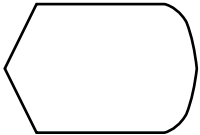
Tabel 2.2. Simbol-Simbol dalam *Blockchart*

Simbol	Keterangan
	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/ bendel/ berkas atau cetakan.
	Multi dokumen

Lanjutan Tabel 2.2. Simbol-Simbol dalam *Blockchart*

Simbol	Keterangan
	Proses Manual
	Proses yang dilakukan oleh komputer.
	Menandakan dokumen yang diarsipkan (arsip manual)
	Data penyimpanan (<i>Storage</i>)
	Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
	Terminasi yang menandakan awal dan akhir dari suatu aliran
	Pengambilan keputusan (<i>decision</i>)

**Lanjutan Tabel 2.2.** Simbol-Simbol dalam *Blockchart*

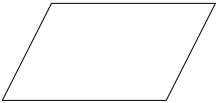
Simbol	Keterangan
	Layar peraga (<i>monitor</i>).
	Pemasukkan data secara manual.

(Sumber : Kristanto, 2008:75-77)

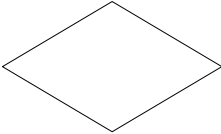
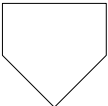
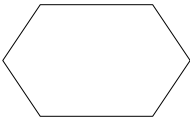
2.2.3 Bagan Alir (*Flowchart*)

Sitorus (2015:14) menjelaskan, “*Flowchart* adalah langkah-langkah menyelesaikan masalah yang dituliskan dalam simbol-simbol tertentu.” Menurut Sitorus (2015:14-16), Gambaran simbol-simbol *Flowchart* sebagai berikut:

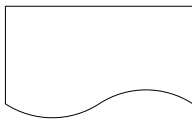
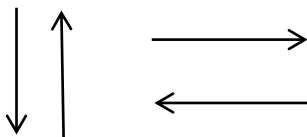
Tabel 2.3. Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
	<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
	<i>Input/output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya.
	<i>Proccess</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.

Lanjutan Tabel 2.3. Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
	<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
	<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
	<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
	<i>Predefined process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	<i>Fanned Card</i>	Menyatakan input berasal dari kartu atau output data ke kartu.
	<i>Punch Tape</i>	

Lanjutan Tabel 2.3. Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
	<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui Printer)
	<i>Flow</i>	Menyatakan jalannya arus suatu proses

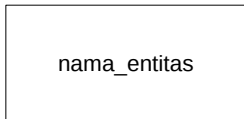
(Sumber: Sitorus, 2015:15-16)

2.2.4 Pengertian ERD (*Entity Relational Diagram*)

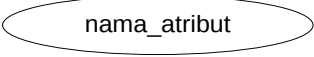
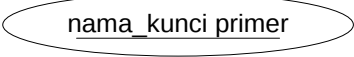
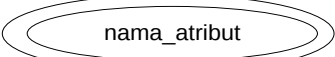
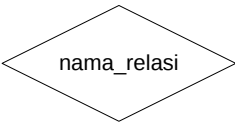
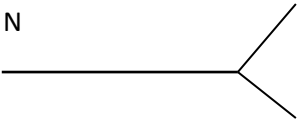
Rosa dan Shalahuddin (2016:50) menjelaskan, “*Entity Relational Diagram* (ERD) digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain.”

Adapun simbol-simbol yang digunakan pada ERD dengan notasi Chen adalah sebagai berikut:

Tabel 2.4. Simbol-Simbol *Entity Relationship Diagram* (ERD)

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimoan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel

**Lanjutan Tabel 2.4.** Simbol-Simbol *Entity Relationship Diagram* (ERD)

Simbol	Deskripsi
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
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)
Atribut multivalai / multivalue 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / association 	Penghubung antar relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering



	disebut dengan one to many menghubungkan entitas A dan entitas B maka
--	---

2.2.5 Pengertian Kamus Data

Sukamto dan Shalahuddin (2016:73) menjelaskan, “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (ouput) dapat dipahami secara umum (memiliki standar cara penulisan).” Kamus data memiliki beberapa simbol sebagai berikut :

Tabel 2.5. Simbol-simbol Kamus Data

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

(Sumber: Sukamto dan Shalahuddin, 2016:74)

2.2.6 Metode Pengujian Perangkat Lunak

2.2.6.1 Pengertian Metode Pengujian

Sukamto dan Shalahuddin (2016:272) menyatakan, “Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian”.



2.2.6.2 Metode Pengujian

Secara umum pola pengujian perangkat lunak adalah sebagai berikut:

1. Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai isi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (*independent test group* (ITG)).
4. Pengujian dan penirkutuan (*debugging*) merupakan aktivitas yang berbeda tetapi penirkutuan (*debugging*) harus diakomodasikan pada berbagai strategi pengujian.

2.2.6.3 Back-Box Testing (Pengujian Kotak Hitam)

Sukamto dan Shalahuddin (2016:275), “*Black-box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji coba yang dilakukan adalah:

- a. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.



2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Menurut Juansyah (2015:2), “Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju”. Hal senada juga diungkapkan oleh Riswaya et.al (2014:1) “Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai tujuan pembuatan aplikasi tersebut.”

Dari beberapa definisi di atas penulis menyimpulkan bahwa aplikasi adalah program siap pakai yang dapat digunakan untuk melaksanakan suatu fungsi dan perintah-perintah dari pengguna aplikasi agar dapat digunakan sesuai dengan tujuan.

2.3.2 Pengetian Web

Menurut Shelly, Gary B dan Misty E.Vermaat (2012:12), “Web adalah salah satu jasa yang paling populer di internet. Web berisi miliaran dokumen yang disebut laman-laman web. Laman web dapat berisi teks, gambar, audio, dan video”. Hal senada juga dikemukakan oleh Arief (2011:7), “Web adalah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protocol HTTP (*Hypertext Transfer Protokol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*.”

Dari beberapa definisi di atas penulis menyimpulkan bahwa Web adalah dokumen yang berisi teks, gambar, audio, dan video yang dapat diakses melalui aplikasi *browser*.

2.3.3 Pengertian Pengolahan Data

Kristanto (2008:8), pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan. Hal senada juga diungkapkan oleh Ladjamudin (2013:9), “Pengolahan



data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan data menjadi informasi yang memiliki kegunaan”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa pengolahan data adalah waktu yang diperlukan untuk menghasilkan perubahan bentuk data menjadi informasi yang berguna

2.3.4 Pengertian Monitoring

Menurut Sutabri (dikutip oleh Herliana dan Rasyid, 2016:43), “*Monitoring* didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan”. Sedangkan menurut Widiastuti dan Susanto (2012:196), “*Monitoring* adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis dan kontinu tentang kegiatan/program sehingga dapat dilakukan tindakan koreksi untuk penyempurnaan program/ kegiatan selanjutnya.

Dari beberapa definisi di atas penulis menyimpulkan bahwa *monitoring* adalah langkah untuk mengkaji apakah kegiatan atau program yang dilaksanakan secara sistematis dan kontinu telah sesuai dengan pencapaian tujuan.

2.3.5 Pengertian CSR (*Corporate Social Responsibility*)

Said (2015:4) menjelaskan, “*CSR (Corporate Social Responsibility)* adalah kewajiban sosial swasta maupun perusahaan kepada masyarakat dan pemerintah sebagai dampak dari ekspansi bisnisnya yang dimungkinkan telah mengganggu keseimbangan lingkungan sosial kemasyarakatan dimana mereka menjalankan aktivitasnya. Selain itu menurut Nurdizal, et al, “*CSR (Corporate Social Responsibility)* adalah tindakan atau konsep yang dilakukan oleh perusahaan (sesuai kemampuan perusahaan tersebut) sebagai bentuk tanggungjawab mereka terhadap sosial atau lingkungan sekitar perusahaan berada. Program CSR meliputi pengurangan kemiskinan, pelestarian lingkungan dan pembangunan ekonomi yang



berkelanjutan adalah bagian dari upaya pengembangan perusahaan secara berkelanjutan yang berguna untuk membantu memperbaiki *financial performance* dan akses modal, serta meningkatkan *corporate image* dalam penjualan atau layanan jasa.

Dari beberapa definisi di atas penulis menyimpulkan bahwa CSR (*Corporate Social Responsibility*) adalah kewajiban sosial swasta maupun perusahaan kepada masyarakat dan pemerintah sebagai bentuk tanggungjawab mereka terhadap sosial atau lingkungan sekitar perusahaan berada.

2.3.6 Pengertian Beasiswa

Menurut Sulistyono dan Winiarti (2015:1), “Beasiswa merupakan pemberian berupa bantuan keuangan maupun pendidikan yang diberikan perorangan, mahasiswa atau pelajar yang digunakan demi keberlangsungan pendidikan yang ditempuh bagi yang memiliki prestasi dibidang akademik, non akademik dan kemampuan ekonominya lemah, serta telah memenuhi syarat-syarat yang ditentukan pihak pemberi beasiswa”. Hal senada juga diungkapkan oleh Jumadi (2015:116), “Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan. Pemberian beasiswa dapat dikategorikan pada pemberian cuma-cuma ataupun pemberian dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan perorangan, mahasiswa atau pelajar yang memiliki prestasi dibidang akademik, non akademik dan kemampuan ekonominya lemah yang bertujuan untuk digunakan saat menempuh pendidikan dengan waktu yang telah ditetapkan dan diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan.



2.3.7 Pengertian Aplikasi Pengolahan Data dan Monitoring Mahasiswa

Penerima Biaya Pendidikan Sekitar Bukit Asam (Bidiksiba) pada Kantor CSR PT Bukit Asam Persero, Tbk Tanjung Enim

Aplikasi pengolahan data dan monitoring mahasiswa Penerima Biaya Pendidikan Sekitar Bukit Asam (Bidiksiba) pada Kantor CSR PT Bukit Asam Persero, Tbk Tanjung Enim adalah sebuah aplikasi yang digunakan untuk mengolah data dan monitoring prestasi akademik (IPK) mahasiswa Bidiksiba khususnya di Politeknik Negeri Sriwijaya oleh bidang pendidikan CSR PTBA.

2.4 Teori Program

2.4.1 Basis Data (*Database*)

Menurut Setyaningrum (2013:2), “Basis data merupakan kumpulan data yang terdiri dari atribut, *entity* dan *relationship* dari informasi suatu instansi atau perusahaan yang dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi”. Sedangkan menurut Puspitosari (2013:3), “*Database* adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat.

Dari beberapa definisi di atas penulis menyimpulkan bahwa basis data (*database*) adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat memenuhi kebutuhan informasi suatu organisasi.

Wahana Komputer (2010:143-144), mengemukakan bahwa dalam *database* yang struktur data disimpan dalam berbagai tingkatan/hierarki hingga membentuk suatu database. Tingkatan yang tertinggi adalah *database*, sedangkan tingkatan terendah adalah data value/nilai.

1. *Value*/nilai

Value/nilai adalah tingkatan data terendah yang menyusun suatu database.

Value/nilai adalah data yang disimpan di dalam setiap kolom/elemen.

2. *Field*/kolom/atribut

Field/kolom/atribut adalah tingkatan kedua yang menyusun suatu record yang menjelaskan kumpulan data yang disimpan. Suatu *record* yang menjelaskan kumpulan data yang disimpan.



3. *Record*/baris

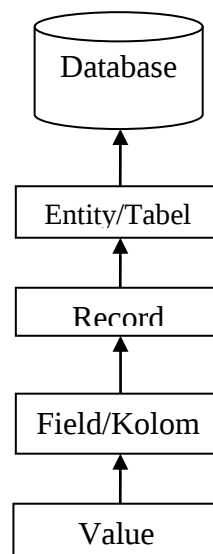
Record/baris adalah kumpulan dari *field*/kolom yang saling berhubungan yang membentuk suatu tabel. Satu *record* akan menyimpan satu informasi tentang data objek yang disimpan.

4. *Entity*/tabel

Entity/tabel adalah kumpulan dari *record* data yang menjelaskan tentang subjek data.

5. *Database*/file

Database/file adalah kumpulan tabel-tabel yang menjelaskan suatu subjek data.



Gambar 2.1 Tingkatan Data

2.4.1.1 Pengertian MySQL

Menurut Nugroho (2013:1), “MySQL adalah *Relational Database Management System* (RDMS), yaitu database relasi yang memiliki relasi standar yaitu SQL (*Structured Query Language*)”. Sedangkan menurut Arief (2011:151), “MySQL adalah salah satu jenis database server yang digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya.”

Dari beberapa definisi di atas penulis menyimpulkan bahwa, MySQL adalah salah satu jenis database server yang digunakan untuk mengelola data berupa



database relasi yang memiliki relasi standar yaitu SQL (*Structured Query Language*).

2.4.2 Web Editor

Menurut Arief (2011:19), “*Web editor* adalah program aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen *web* baik *Client side scripting* maupun *server side scripting*. Penulis menggunakan *Notepad++* sebagai web editor dalam membangun sebuah aplikasi pengolahan data dan monitoring Mahasiswa bidiksiba. dan juga fitur-fitur yang terdapat didalamnya mudah digunakan oleh penulis.

2.4.2.1 Notepad++

Menurut Pahlevi dan Krisnawati (2013:4), “*Notepad++* adalah sebuah aplikasi *text editor* yang bersifat gratis. *Notepad* menitikberatkan kegunaan aplikasi untuk editing *text* dalam waktu yang cepat dan praktis. *Notepad++* mendukung banyak format bahasa pemrograman seperti PHP, HTML, JavaScript dan CSS”. *Notepad++* adalah salah satu *text editor* yang paling banyak di gunakan untuk menuliskan sebuah program. Dari beberapa definisi di atas penulis menyimpulkan bahwa *notepad++* adalah salah satu *text editor* yang paling banyak di gunakan untuk menuliskan sebuah program yang mendukung banyak format bahasa pemrograman seperti PHP, HTML, *JavaScript* dan CSS.

2.4.3 Bahasa Pemrograman

2.4.3.1 HTML (*HyperText Markup Language*)

Menurut Setiawan (2015:1), “HTML (*HyperText Markup Language*) merupakan Bahasa pengkodeaan untuk menghasilkan dokumen-dokumen *hypertext* untuk digunakan di *World Wide Web* atau *web*.” Hal senada juga diungkapkan oleh Arief (2011:23), “HTML (*HyperText Markup Language*) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan pada *web browser*.



Dari beberapa definisi di atas penulis menyimpulkan bahwa HTML (*HyperText Markup Language*) merupakan Bahasa pengkodeaan untuk menghasilkan dokumen berupa *web page* yang disajikan pada *web browser*.

2.4.3.2. PHP (*Hypertext Preprocessor*)



Gambar 2.2 Tampilan Logo PHP

(Sumber: www.yudana.id)

Menurut Badiyanto (2013:43), “PHP (*Hypertext Preprocessor*) adalah Bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML/PHP banyak dipakai untuk membuat situs *web* dinamis”. Sedangkan menurut Shelly, Gary B dan Misty E. Vermaat (2012:680), “PHP (*Hypertext Preprocessor*), adalah bahasa yang serupa dengan C, java, dan perl yang biasanya digunakan pada *server web* yang berbasis linux untuk dapat menciptakan halaman *web* dinamis dengan memasukan *script* PHP kedalam dokumen HTML pada halaman *web*.

Dari beberapa definisi di atas penulis menyimpulkan bahwa PHP (*Hypertext Preprocessor*) bahasa *server-side scripting* yang dapat menciptakan halaman *web* dinamis dengan memasukan *script* PHP kedalam dokumen HTML pada halaman *web*.

2.4.3.2.1 Skrip PHP

Badiyanto (2013:32-33) mengatakan bahwa, “PHP yang merupakan sebuah bahasa scripting yang terpasang pada HTML, dan skrip HTML (*Hyper Text Markup Language*) adalah bahasa standar untuk membuat halaman *web*. Biasa file dituliskan dengan ekstensi *.htm* atau *.html*”.

Contoh:

File latihan1.html

```
<HTML>
```

```
<HEAD>
```



```

<TITLE> Latihan HTML </TITLE>
</HEAD>
<BODY>
    Mari Belajar Membuat Web
</BODY>
</HTML>

```

Contoh di atas bisa ditulis dengan menggunakan PHP sebagai berikut yang kodenya di simpan dengan latihan1.php.

```

<HTML>
<HEAD>
<TITLE> Latihan HTML </TITLE>
</HEAD>
<BODY>
<?php
    Printf ("Mari Belajar Membuat Web");
    // atau
    Echo "<br>";
    Echo "Mari Belajar Membuat Web";
?>
</BODY>
</HTML>

```

2.4.3.3 Pengertian Java Script

Menurut Shelly, Gary B dan Misty E.Vermaat (2012:680), “*JavaScript* adalah bahasa terjemahan yang memungkinkan pemrograman menambahkan muatan dinamis dan elemen-elemen interaktif pada suatu halaman *web*”. Sedangkan menurut Wahana (2014:2), “*JavaScript* adalah bahasa pemrograman *client side* untuk memberikan efek dinamis pada antarmuka *web*”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa *javascript* adalah bahasa pemrograman *client side* dengan menambahkan muatan dinamis dan elemen-elemen interaktif pada suatu halaman *web*.



2.4.3.3.1 Penulisan Java Script

Menurut Badiyanto (2013:27), “JavaScript biasanya dituliskan dalam bentuk fungsi yang ditaruh di tag <head> yang dibuka dengan tag <script type=”teks/javascript>”.

Contoh:

```
<script type=”teks/javacsript.....”>
    Alert(“Halo Apa Kabar”);
</script>
```

Kode JavaScript juga bisa diletakkan di *file* sendiri yang berekstensi .js (singkatan dari JavaScript).

Contoh :

```
<script type=”teks/javacsript” scr=alamat.js”>
</script>
```

2.4.3.4 Pengertian CSS (*Cascading Style Sheet*)

Menurut Wahana Komputer (2014:2), “CSS (*Cascading Style Sheet*) adalah bahasa yang digunakan untuk mengatur tampilan halaman web”. Sedangkan menurut Madcoms (2013:54), “CSS (*Cascading Style Sheet*) adalah kumpulan kode-kode untuk mengendalikan tampilan isi dari suatu halaman *web*”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa CSS (*Cascading Style Sheet*) adalah salah satu bahasa desain web yang digunakan untuk mengatur dan memisahkan konten utama dengan tampilan dokumen yang meliputi layout, warna dan huruf pada tampilan halaman *web*.

2.4.4 Webserver

Untuk bisa mengakses aplikasi web yang akan dibuat oleh penulis, maka diperlukan yang namanya *webserver*. Penulis menggunakan *XAMPP* sebagai *webserver* dalam membangun Aplikasi Pengolahan Data dan Monitoring Mahasiswa Bidiksiba di Politeknik Negeri Sriwijaya.



2.4.4.1 Pengertian XAMPP

Prasetyo dan Dahlan (2015:31) menjelaskan, “*XAMPP* adalah sebuah perangkat lunak yang di dalamnya sudah tersedia database server MySQL yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows”. Sedangkan menurut Nugroho (2013:1) bahwa, “*XAMPP* adalah paket program berbasis *web*, yang didalamnya berisi *software Apache, PHP* dan *database MySQL*”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa *XAMPP* adalah salah satu paket software *web server* yang terdiri dari *Apache, MySQL, PHP* dan *phpMyAdmin* yang dapat digunakan dalam pemograman *web*.