



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Perangkat Lunak

Sukamto dan Shalahuddin (2013:2), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumen perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).”

Sutarman (2009:88), “*Software* merupakan suatu program yang berisi barisan instruksi (perintah) yang ditulis dalam bahasa komputer yang dimengerti oleh *hardware* komputer.

Jadi, perangkat lunak atau *software* merupakan bagian penting dari komputer yang berupa program-program untuk mengoperasikan suatu pekerjaan dengan aturan tertentu.

2.1.2. Pengertian Program

Sutarman (2012:3), “Program adalah barisan perintah/instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sebagai barisan perhitungan numerik, dimana barisan perintah tersebut berhingga, berakhir, dan menghasilkan *output*.”

Sutarman (2009:3), “Program adalah barisan perintah atau instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sebagai barisan perhitungan numerik, di mana barisan perintah tersebut berhingga, berakhir dan menghasilkan *output*.”

Jadi, program merupakan kumpulan instruksi yang dapat dimengerti komputer dan memudahkan penggunaanya dalam mengoperasikan suatu pekerjaan hingga menghasilkan suatu keluaran yang berarti.

2.1.3. Pengertian Data

Asropuddin (2013:22), “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti”.

Sutabri (2012:3), “Data adalah hal, peristiwa, atau kenyataan lain ataupun yang mengandung sesuatu pengetahuan untuk dijadikan dasar guna penyusunan keterangan, pembuatan kesimpulan, atau penetapan keputusan.”

Jadi, data adalah kumpulan dari angka maupun karakter yang berasal dari kenyataan yang belum berarti dan harus diolah terlebih dahulu untuk menjadi suatu informasi yang lebih berarti bagi penggunaannya.

2.1.4. Pengertian Sistem

Sutabri (2012:10), “Sistem merupakan kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.”

Kristanto (2008:1), “Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.”

Jadi, sistem merupakan sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan.

2.1.5. Karakteristik Sistem

Sutabri (2012:13-14) menjelaskan bahwa sistem memiliki karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan.

2. 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-pisah.

3. 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.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki sasaran dan tujuan 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.

2.1.6. Metode Pengembangan Sistem

Sukanto dan Shalahudin (2013:28) 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 terurut mulai dari analisis, desain, pengkodean, pengujian, dan pemeliharaan.

a. Analisis

Tahap analisis dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami sistem seperti apa yang dibutuhkan oleh *user*.

b. Desain

Tahap desain 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.

c. Pengkodean

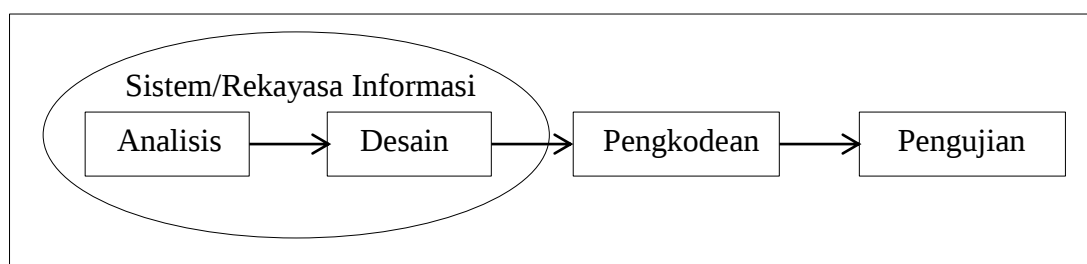
Pada tahap pengkodean, desain harus ditranslasikan ke dalam program sistem. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Tahap pengujian fokus pada sistem 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. Pemeliharaan

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



Gambar 2.1. Ilustrasi Model *Waterfall*

(Sumber : Sukamto dan Shalahudin, 2013 : 28)

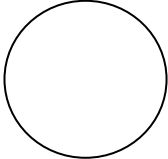
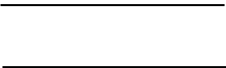
2.2. Teori Khusus

2.2.1. Pengertian DFD (*Data Flow Diagram*)

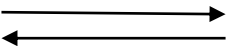
Sukamto dan Shalahuddin, (2013:70), “ *Data Flow Diagram* (DFD) atau dalam bahasa indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

Sukamto dan Shalahuddin, (2013:71), menjelaskan notasi pada DFD adalah sebagai berikut :

Tabel 2.1. Simbol-simbol *Data Flow Diagram*

No	Simbol	Keterangan
1.		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja.
2.		<i>File</i> 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</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM))

Lanjutan Tabel 2.1. Simbol-simbol *Data Flow Diagram*

No	Simbol	Keterangan
3.		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>) berupa kata benda.
4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>) catatan : nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”

(Sumber : Sukamto dan Shalahuddin, 2013:71)

Menurut Sukamto dan Shalahuddin (2013:72), berikut ini adalah tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* :

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 2.

2.2.2. Pengertian Blockchart

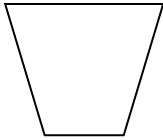
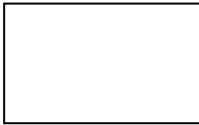
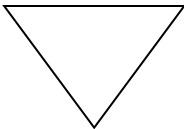
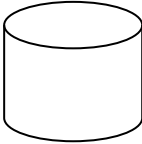
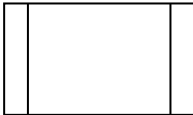
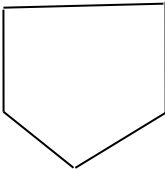
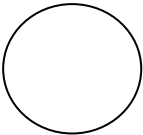
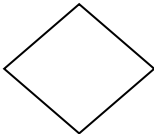
Kristanto (2008:75) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Kristanto (2008:75) menjelaskan, “Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

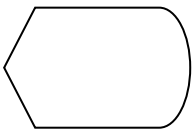
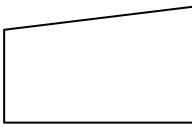
Tabel 2.2. Simbol-simbol dalam *Block Chart*

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

Lanjutan Tabel 2.2. Simbol-simbol dalam *Block Chart*

No.	Simbol	Keterangan
3.		Proses Manual
4.		Proses dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (<i>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.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).

Lanjutan Tabel 2.2. Simbol-simbol dalam *Block Chart*

No	Simbol	Keterangan
12.		Layar peraga (<i>monitor</i>).
12.		Pemasukkan data secara manual.

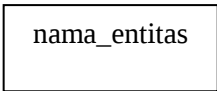
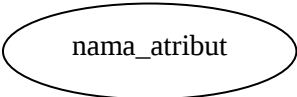
(Sumber : Kristanto, 2008 : 75)

2.2.3. Pengertian ERD (*Entity Relational Diagram*)

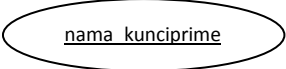
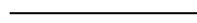
Sukamto dan Shalahuddin, (2013:289), “*Entity Relationship Diagram (ERD)* adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relational.”

Sukamto dan Shalahuddin, (2013:50), menjelaskan simbol-simbol yang digunakan dalam ERD, yaitu:

Tabel 2.3. Simbol-simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
1.	Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel 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 tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas

Tabel 2.3. Simbol-simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
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.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
5.	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B

(Sumber : Sukanto dan Shalahuddin, 2013:50)

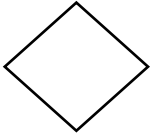
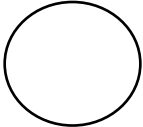
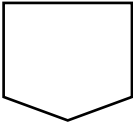
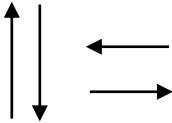
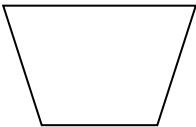
2.2.4. Pengertian *Flowchart*

Ewolf Community (2012:16) mengemukakan, “*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya”.

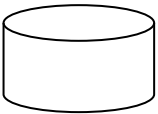


Ewolf Community (2012:16) juga menjelaskan tentang simbol-simbol yang sering digunakan dalam *flowchart* dapat dilihat pada tabel berikut ini:

Tabel 2.4. Simbol-simbol dalam *Flow Chart*

No.	Simbol	Arti
1.		Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i>
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja
2.		Simbol <i>Input/Output</i> 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.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
7.		Simbol untuk menghubungkan antar proses atau antar simbol
8.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, <i>printer</i> , dll
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual

Lanjutan Tabel 2.4. Simbol-simbol dalam *Flow Chart*

No.	Simbol	Arti
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen
11.		Simbol yang menyatakan bagian dari program (subprogram)
12.		Simbol masukan atau keluaran dari atau ke sebuah pita magnetic
12.		Simbol database atau basis data

(Sumber : Ewolf community, 2012:16)

2.2.5. Pengertian Kamus Data

Sukamto dan Shalahuddin, (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).”

Sukamto dan Shalahuddin, (2013:73), menjelaskan simbol-simbol yang di gunakan dalam kamus data, yaitu :

Tabel 2.5. Simbol-simbol dalam Kamus Data

No	Simbol	Arti
1	=	disusun atau terdiri atas
2	+	Dan
3	[]	baik ...atau...
4	{ } ⁿ	n kali diulang/ bernilai banyak
5	()	data operasional
6	*...*	batas komentar

(Sumber : Sukamto dan Shalahuddin, 2013:73)

2.2.6. Pengertian Daftar Kejadian (*Event List*)

Kristanto (2008:70) menjelaskan bahwa daftar kejadian (*event list*) digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggapan atau respon dari sistem. Adapun cara mendeskripsikan daftar kejadian adalah sebagai berikut :

1. Pelaku adalah entity luar, jadi bukan sistem.
2. Menguji setiap entity luar dan mencoba mengevaluasi setiap entity luar yang terjadi pada sistem.
3. Hati-hati dengan kejadian yang spesifik, yang tak sengaja menyatu dalam paket yang sama.
4. Harus diingat bahwa kejadian yang dimodelkan bukan hanya interaksi normal antara sistem dengan entiti luar, karena itu harus dievaluasi kebutuhan sistem untuk menanggapi kejadian yang gagal.
5. Setiap aliran keluaran sebaiknya merupakan respon dari kejadian.
6. Setiap kejadian yang tidak berorientasi pada waktu dalam daftar kejadian sebaiknya mempunyai masukan sehingga sistem dapat mendeteksi kejadian yang berlangsung.
7. Setiap kejadian sebaiknya menghasilkan keluaran langsung sebagai respon atau disimpan dalam berkas untuk bahan masukan.

2.3. Pengertian-Pengertian Judul

2.3.1. Pengertian Aplikasi

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

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

Jadi, dapat disimpulkan bahwa aplikasi adalah alat yang dibuat untuk mengerjakan tugas-tugas tertentu yang difungsikan secara terpadu.

2.3.2. Pengertian Pengolahan Data

Sutarman (2012:4), “Pengolahan data adalah proses perhitungan/transformasi data *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan.”

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

Jadi, pengolahan data merupakan proses pengolahan data yang diinputkan untuk diubah menjadi informasi yang berguna.

2.3.3. Pengertian Legalisir

Legalisir atau legalisasi dipakai untuk menjamin keabsahan suatu dokumen atau surat, seperti dikatakan oleh Imanta Sembiring (2012) sebagai berikut.

Legalisir dokumen atau surat yang dibuat di bawah tangan tangan tersebut ditanda-tangani di hadapan notaris, setelah dokumen atau surat tersebut dibacakan atau dijelaskan oleh Notaris yang bersangkutan. Sehingga tanggal dokumen atau surat yang bersangkutan adalah sama dengan tanggal legalisasi dari notaris. Dengan demikian, notaris menjamin keabsahan tanda-tangan dari para pihak yang dilegalisir tanda-tangannya, dan pihak (yang bertanda-tangan dalam dokumen) karena sudah dijelaskan oleh notaris tentang isi surat tersebut, tidak bisa menyangkal dan mengatakan bahwa yang bersangkutan tidak mengerti isi dari dokumen atau surat tersebut.

2.3.4. Pengertian Aplikasi Pengolahan Data Legalisir Ijazah pada Politeknik Negeri Sriwijaya Palembang

Aplikasi Pengolahan Data Legalisir Ijazah pada Politeknik Negeri Sriwijaya Palembang adalah suatu aplikasi yang dibuat dengan menggunakan bahasa pemrograman *PHP* dan *Database MySql* dengan tujuan untuk membantu proses pengolahan data legalisir ijazah bagi Bagian Akademik Kemahasiswaan (BAK), sehingga dapat mengoptimalkan kinerja pengolahan data legalisir yang terjadi di Politeknik Negeri Sriwijaya Palembang.

2.4. Teori Program

2.4.1. Basis Data (*Database*)

Priyadi (2014:2) menyatakan, “Secara umum, pengertian basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital.”

Wahana Komputer (2010:2) menyatakan bahwa, *database* adalah sebuah struktur yang umumnya terbagi dalam dua hal, yaitu database flat dan sebuah database relasional.

Priyadi (2014:3) juga menjelaskan beberapa elemen basis data, diantaranya adalah sebagai berikut:

1. Tabel

Pada suatu basis data, tabel direpresentasikan menjadi suatu bentuk segiempat berupa matriks, yang terdiri dari kolom dan baris.

2. *Field*

Pada tabel basis data, kolom merupakan representasi untuk sebuah nama *field* yang pembacaan datanya dilakukan secara vertikal.

3. *Record*

Pada tabel basis data, baris merupakan suatu representasi untuk sebuah *record* yang pembacaan datanya dilakukan secara horizontal. Satu baris pada sebuah tabel merupakan data yang dimiliki oleh satu *record*. Nilai-nilai yang dimiliki oleh sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabel tersebut.

4. Kardinalitas

Kardinalitas merupakan batasan dari banyaknya hubungan, yang dapat dilakukan oleh himpunan entitas dalam melakukan relasi dengan himpunan entitas lainnya. Variasi kemungkinan untuk melakukan relasi yang dimiliki oleh kardinalitas terdiri dari empat macam, yaitu:

- a. Satu ke satu $\longrightarrow$ (1:1).
- b. Satu ke banyak $\longrightarrow$ (1:N).
- c. Banyak ke satu $\longrightarrow$ (N:1).
- d. Banyak ke banyak $\longrightarrow$ (N:N).

2.4.2. Pengertian HTML

Winarno, dkk. (2014:1), “*Hyper Text Markup Language (HTML)* adalah sebuah bahasa untuk menampilkan konten di Web.

Risnandar (2013:12), “*HTML* merupakan sesuatu yang dapat diterjemahkan pada komputer-komputer yang bervariasi baik spesifikasi komputer, *Operating System* maupun *browser*-nya.”

2.4.3. Pengertian CSS

Risnandar (2013:39), “*CSS (Cascading Style Sheets)* adalah kumpulan aturan-aturan pemformatan yang mengontrol tampilan konten dalam sebuah halaman web.”

Saputra (2012:27), “*CSS* yang merupakan singkatan dari *Cascading Style Sheet* merupakan bahasa pemrograman web yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web lebih rapih, terstruktur, dan seragam.”

2.4.4. Pengertian MySQL

Risnandar (2013:92), “*MySQL* merupakan basis data yang bersifat *open source* sehingga banyak digunakan di dunia.”

Saputra (2013:14) menyatakan, “*MySQL* merupakan *database storage engine* yang paling banyak digunakan oleh *web developer* karena sifatnya *free*, alias gratis.”

Dari kedua pengertian di atas, dapat disimpulkan bahwa *MySQL* adalah jenis *software* pada *Relational Database Management System (RDBMS)* pada sebuah manajemen *database* sebagai basis data.

2.4.5. Sekilas Tentang PHP

2.4.5.1. Pengertian PHP (Hypertext Preprocessor)

Winarno dkk. (2014:49), “*PHP* adalah sebuah bahasa pemrograman web berbasis server (*server-side*) yang mampu memarsing kode PHP dari kode web dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client (browser)*.”

Risnandar (2013:57), “*PHP (Hypertext Preprocessing)* merupakan bahasa pemrograman yang biasa digunakan untuk membuat halaman *HTML*.”



Sidik (2012:4), “PHP merupakan secara umum dikenal sebagai bahasa pemrograman script-script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

2.4.5.2. Skrip *PHP*

Menurut Risnandar (2013:58), berikut adalah contoh yang umum digunakan untuk menjelaskan tentang PHP sebagai script yang disisipkan (*embedded script*) dalam dokumen HTML :

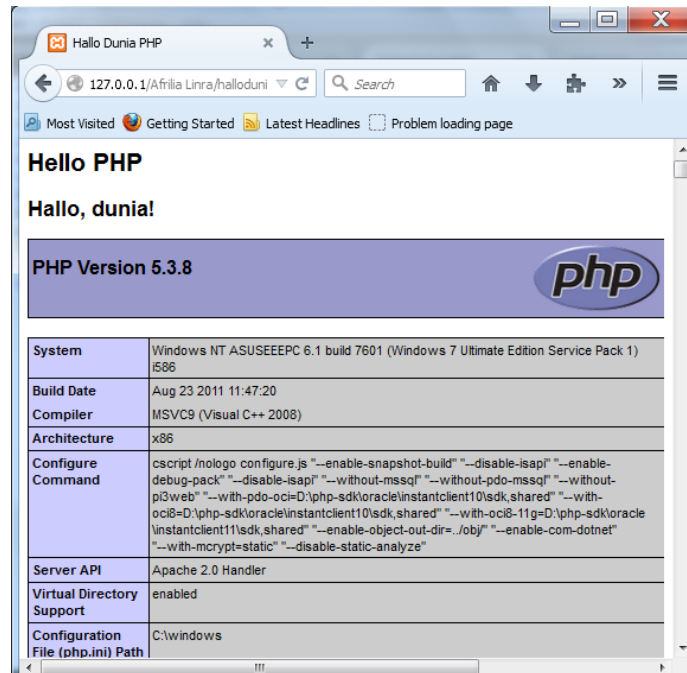
```
<html>
<head>
<title>Hallo Dunia PHP</title>
</head>
<body>
<h1>Hello PHP</h1>
<?php
echo '<h2>Hallo, dunia!</h2>';
phpInfo();
?>
</body>
</html>
```

(**Sumber** : Risnandar, 2012 : 58)

Perhatikan contoh diatas, setiap *statement* dalam *PHP* selalu diakhiri dengan tanda titik koma (;). Dalam contoh diatas terdapat dua buah *statement* yaitu :

- echo digunakan untuk menampilkan tulisan HTML <h2>Hallo, dunia!</h2> di file yang akan dibaca oleh browser.
- phpInfo(); → *statement* ini digunakan untuk memerintahkan browser untuk menampilkan versi *PHP* beserta detailnya.

Setelah disimpan dengan nama file `hallodunia.php` dan dijalankan di *browser* berikut ini adalah tampilannya :



Gambar 2.2. Tampilan Skrip *PHP* pada *browser*
(*sumber : Risnandar, 2013 : 59*)

2.4.6. Sekilas tentang XAMPP

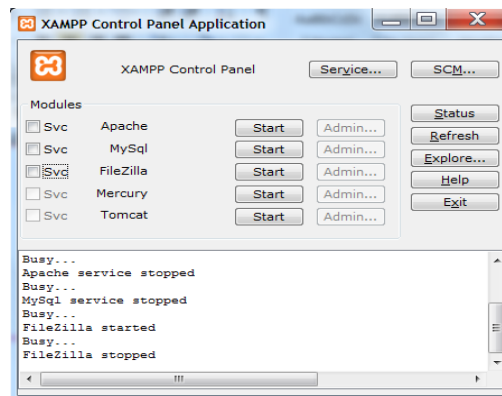
Sidik (2012:72), “XAMPP (X(/Window/Linux) Apache MySQL PHP dan Perl) merupakan paket *server web* PHP dan database MySQL yang paling populer di kalangan pengembangan *web* dengan menggunakan PHP dan MySQL sebagai databasenya. Paket XAMPP, sesuai dengan kepanjangannya, X yang berarti Windows atau Linux, pengguna bisa memilih paket yang diinginkan untuk Windows atau Linux. XAMPP dapat diperoleh dari <http://xampp.org> atau <http://apache-friends.org>.

Risnandar (2013:53), “XAMPP adalah suatu program yang digunakan sebagai server untuk mengeksekusi fungsi yang ada dalam halaman website yang kita buat sekaligus menampilkan halaman website tersebut agar bisa diakses oleh *user*. ”

Menurut Sidik (2012:73), langkah-langkah pemasangan XAMPP dari file zip yang telah di download adalah sebagai berikut :

1. Unzip file XAMPP yang telah didownload, ke dalam file directori `c:/xampp`

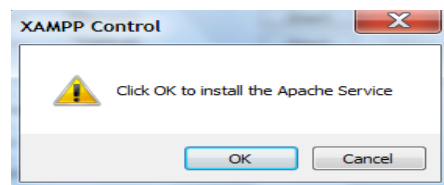
2. Masuk ke dalam directori c:/xampp
3. Pilih file xampp-control.exe, kemudian jalankan dengan menekan <enter> atau klik dua kali, tampilan berikut akan ditampilkan :



Gambar 2.3. Tampilan XAMPP Control Panel

(Sumber : Menurut Sidik 2012:73)

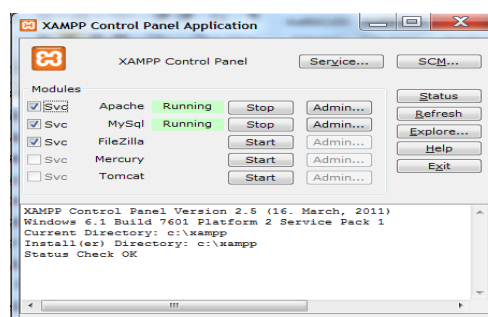
Dari tampilan di atas, beri tanda cek pada baris Svc Apache, dialog konfirmasi unjuk menjalankan Apache menjadi Service ditampilkan seperti berikut :



Gambar 2.4. Tampilan Dialog Konfirmasi Menjalankan Apache

(Sumber : Menurut Sidik 2012:73)

4. Klik tombol OK.
5. Setelah itu klik tombol start pada baris yang sama, menjalankan Apache. Maka tampilan menjadi seperti berikut :



Gambar 2.5. Tampilan Svc Apache Running

(Sumber : Menurut Sidik 2012:73)



2.4.7. Sekilas tentang *Sublime Text 2.0.2*

2.4.7.1. Pengertian *Sublime Text 2.0.2*

Sibarani (2013: 43), “*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform* sistem operasi dengan menggunakan teknologi *Phyton API*. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah *fleksibel* dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.”

Berikut beberapa fitur yang diunggulkan dari aplikasi *Sublime Text 2.0.2*:

a. *Goto Anything*

Fitur yang sangat membantu dalam membuka *file* ataupun menjelajahi isi dari *file* hanya dengan beberapa *keystrokes*.

b. *Multiple Selections*

Fitur ini memungkinkan *user* untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi *file* lebih cepat dari sebelumnya.

c. *Command Pallete*

Dengan hanya beberapa *keystorkes*, *user* dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

d. *Distraction Free Mode*

Bila *user* memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu *user* dengan memberikan tampilan layar penuh. *Split Editing* dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan *editing* perpecahan. Mengedit sisi *file* dengan sisi, atau mengedit dua lokasi di satu *file*. Anda dapat mengedit dengan banyak baris dan kolom yang *user* inginkan.

e. *Instant Project Switch*

Menangkap semua *file* yang dimasukkan kedalam *project* pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua *file* yang ada ataupun untuk beralih ke *file* dalam *project* lainnya dengan cepat.

f. *Plugin API*

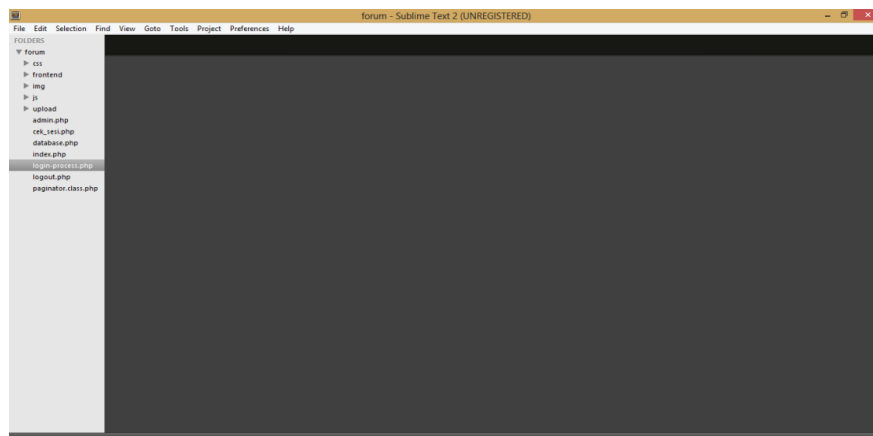
Dilengkapi dengan *plugin* API berbasis *Python* sehingga membuat aplikasi ini sangat tangguh.

g. *Customize Anything*

Aplikasi ini memberikan *user* fleksibilitas dalam hal pengaturan fungsional dalam aplikasi ini.

h. *Cross Platform*

Aplikasi ini dapat berjalan hampir disemua *operating system modern* seperti *Windows*, *OS X*, dan *Linux based operating system*.



Gambar 2.6. Halaman Utama *Sublime Text 2.0.2*

(Sumber : Sibarani, 2013: 44)

2.4.7.2. *Komponen Sublime Text*

a. *Minimap*

Minimap merupakan fitur menunjukkan kode pengguna di seluruh sudut kanan atas layar. Hal ini memungkinkan mereka untuk melihat garis besar keseluruhan kode mereka dan memiliki fungsi untuk melompat antara bagian yang berbeda dari *file*.

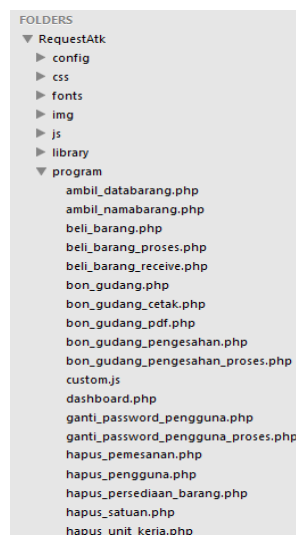


Gambar 2.7. Tampilan *Minimap* pada *Sublime Text 2.0.2*

(**Sumber :** Sibarani, 2013: 45)

b. Side Bar

Komponen dalam *sublime text* yang berfungsi menunjukan *file* yang sedang terbuka pada *sublime text*. Memungkinkan pengguna dalam memilih *file – file* yang terbuka secara cepat.

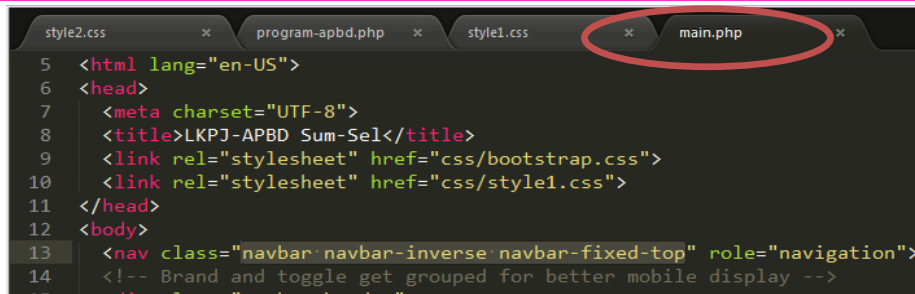


Gambar 2.8. Tampilan *Side Bar* pada *Sublime Text 2.0.2*

(**Sumber :** Sibarani, 2013: 45)

c. Tabs

Salah satu komponen pada *sublime text* yang berfungsi menampilkan nama *file* yang diubah dalam *sublime text*, dan memudahkan penggantian panel dengan mengklik tab yang diinginkan.

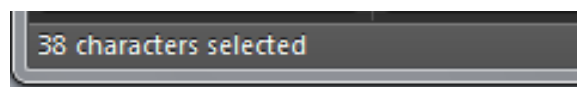


Gambar 2.9. Tampilan Tabs pada Sublime Text 2.0.2

(Sumber : Sibarani, 2013: 44)

d. Status Bar

Menunjukkan posisi baris dan kolom *syntax* yang sedang diubah dalam *sublime text*. Memudahkan pengguna mengetahui posisi *syntax* yang mengalami *error* setelah dijalankan.



Gambar 2.10. Tampilan Status Bar pada Sublime Text 2.0.2

(Sumber : Sibarani, 2013: 46)

e. Menu Bar

Berisi menu-menu yang digunakan untuk membantu dalam membuat dan mengubah *syntax*. Menu dalam *sublime text*, yaitu *file*, *edit*, *selection*, *find*, *view*, *goto*, *tools*, *project*, *preferences* dan *help*.



Gambar 2.11. Tampilan Menu Bar pada Sublime Text 2.0.2

(Sumber : Sibarani, 2013: 47)

f. Pane

Merupakan lembar kerja pada *sublime text*. Berisi kumpulan *syntax* dari *file* yang digunakan pada *sublime text*.



```

1 <?php
2 //koneksi
3 require('../config/connect.php');
4 //identifikasi variabel
5 $tgl_diterima = $_POST['tgl_diterima'];
6 $nama_penerima = $_POST['nama_penerima'];
7 $tgl_diserahkan = $_POST['tgl_diserahkan'];
8 $nama_penyeraah = $_POST['nama_penyeraah'];
9 $nama_penyetuju = $_POST['nama_penyetuju'];
10 $badge_penerima = $_POST['badge_penerima'];
11 $badge_penyeraah = $_POST['badge_penyeraah'];
12 $badge_penyetuju = $_POST['badge_penyetuju'];
13
14 //validasi cek data kosong atau tidak
15 if(empty($tgl_diterima) or empty($tgl_diserahkan) or empty($badge_penerima) or empty($
16     echo "<script>alert('Harap isi semua data.');"location.href='../main.php?p=pengesa
17     exit();
18 endif;
19
20 $sql_insert = "INSERT INTO bon_barang SET tgl_diterima = '$tgl_diterima',badge_peneris
21 $query_insert = mysql_query($sql_insert);
22 echo mysql_error();
23 if($query_insert){
24     echo "<script>alert('Data Pengesahan Bon berhasil ditambah.');"location.href='bon_f
25     exit();
26 }else{
27     echo "<script>alert('data Pengesahan Bon gagal ditambah. Harap hubungi admin.');"l
28     exit();
29 }
30 ?>

```

Gambar 2.12. Tampilan Pane pada Sublime Text 2.0.2

(Sumber : Sibarani, 2013: 44)

2.4.8. JavaScript

Winarno, dkk. (2014:129), “JavaScript adalah bahasa scripting client side yang sangat populer. Ini karena JavaScript bisa dipakai di HTML, web, untuk server, pc, laptop, tablet, ponsel dan lainnya.”

Winarno, dkk (2014:134) menjelaskan bahwa cara menggunakan JavaScript adalah dengan menggunakan tag <script> dan </script>. JavaScript biasanya dituliskan pada bagian <body> dalam HTML. Contohnya :

```

<!Doctype html>
<html>
<body>

<script>
document.write("<h1>Ini adalah Heading </h1>");
document.write("<p>Ini adalah pragraf.</p>");

</body>
</html>

```

(sumber : Winarno, dkk., 2014:134)