



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

Adapun penulis memasukkan teori umum yang berisi pengertian-pengertian yang terdapat dalam penyusunan Laporan Akhir ini. Berbagai pengertian-pengertian tersebut adalah sebagai berikut:

2.1.1. Pengertian Komputer

Hariyanto (2014:1), komputer adalah perangkat yang mampu melakukan komputasi dan membuat keputusan logis, dengan kecepatan cahaya milyaran kali lebih cepat dibanding manusia dalam mengolah data di bawah kendali sekumpulan instruksi yang disebut program atau aplikasi komputer.

Asropudin (2013:19), komputer adalah alat bantu secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.

Hartono (2013:27), komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang untuk secara otomatis melakukan serangkaian urutan penghitungan (*arithmetic*) atau proses-proses yang diurutkan secara logis.

Sujatmiko (2012:156), komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program. Alat serbaguna ini memegang peranan penting dalam teknologi komunikasi.

2.1.2. Pengertian Data

Hartono (2013:15), data adalah hasil pengukuran dan pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan atau kejadian.

Asropudin (2013:22), data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti.

Kadir (2014:44), data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi yang tidak mempunyai makna atau tidak terpengaruh secara langsung kepada pemakai.



2.1.3. Pengertian Informasi

Sutabri (2012:22), informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

Kadir (2013:45), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Dari pengertian diatas, dapat disimpulkan bahwa informasi adalah kumpulan dari data yang telah diolah kedalam bentuk yang lebih berarti bagi penggunanya.

2.1.4. Pengertian Aplikasi

Menurut Sutabri (2012:147), aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai dengan kemampuan yang dimilikinya.

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

Dari pengertian diatas, dapat disimpulkan bahwa aplikasi adalah suatu software atau perangkat lunak yang memiliki fungsi tertentu yang berguna bagi pemakainya.

2.1.5. Pengertian Virtual Account

Dari [website BCA](#) di dapatkan pengertian dari akun virtual, disebut *virtual account* karena setoran yang masuk ke rekening tersebut hanya sesaat. Begitu sampai ke proses batch (proses induk komputer melakukan transaksi akhir hari) maka saldo dari setoran di virtual akun akan dipindahkan.

Virtual Account (“VA”) adalah sistem yang dibuat untuk memfasilitasi pengumpulan dan rekonsiliasi dana. VA mempunyai kode yang unik untuk mengidentifikasi pihak yang melakukan pembayaran. Oleh karena itu, VA tidak dapat dianggap sebagai rekening operasional walaupun terdapat penyebutan “rekening”. VA tidak dapat digunakan untuk tujuan selain pengelompokan dan pemetaan dana masuk yang diterima oleh Nasabah untuk tujuan catatan pembayaran.



2.1.6. Pengertian Keuangan

Keuangan adalah segala sesuatu yang bertalian dengan uang, seluk beluk uang, urusan uang, dan keadaan uang.

2.1.7. Pengertian Perancangan Virtual Account Keuangan Mahasiswa (Khusus Uang KKL) Politeknik Negeri Sriwijaya Berbasis Android

Perancangan *Virtual Account* Keuangan Mahasiswa (Khusus Uang KKL) Politeknik Negeri Sriwijaya Berbasis *Android* adalah suatu sistem aplikasi berbasis *android* yang dibuat untuk memudahkan Sekretaris Jurusan dalam melakukan proses pencatatan pembayaran uang KKL mahasiswa, serta memudahkan mahasiswa untuk mengakses informasi pembayaran uang KKL berbasis *website* bagi mahasiswa yang akan mengikuti kegiatan KKL yang ada di Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya.

2.2. Teori Khusus

Teori ini akan membahas mengenai desain sistem yang merupakan suatu fase dimana diperlukan suatu keahlian perencanaan untuk elemen-elemen komputer yang akan menggunakan sistem baru.

2.2.1. Basis Data (*Database*)

Kadir (2014:218), basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi.

Sukanto dan Shalahuddin (2014:432), basis data adalah sistem yang terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia yang dibutuhkan.

Enterprise (2014:1), *database* adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap database mempunyai API tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data yang ada di dalamnya.

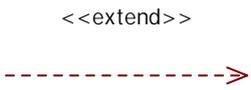
2.2.2. Use Case Diagram

Sukanto dan Shalahuddin (2014:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan



dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah symbol-simbol yang ada pada diagram *use case*:

Tabel 2.1 Simbol-Simbol *Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi / <i>association</i> 	Komunitas antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tamhanan itu; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.



Lanjutan Tabel 2. 1 Simbol-Simbol *Use Case Diagram*

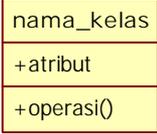
<p>Generalisasi / <i>Generalization</i></p> 	<p>Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p><i>Include</i></p> <p><<include>></p> 	<p>Relasi use case tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini; <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</p>

(Sumber: Sukamto dan Shalahuddin,2014:155)

2.2.3. *Class Diagram*

Sukamto dan Shalahuddin (2014:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system. Kelas memiliki apa yang disebut atribut dan moode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah symbol-simbol yang ada pada diagram kelas:

Tabel 2.2 Simbol-Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur sistem.</p>
<p>Antar muka/ <i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>



Lanjutan Tabel 2.2 Simbol-Simbol *Class Diagram*

Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (whole-part).

(Sumber: Sukamto dan Shalahuddin, 2014:146)

2.2.4. Activity Diagram

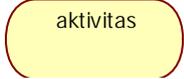
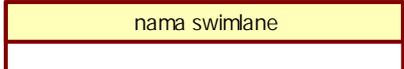
Sukamto dan Shalahuddin (2014:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu utama yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan pa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah symbol-simbol yang ada pada activity diagram:

Tabel 2.3 Simbol-simbol dalam *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.



Lanjutan Tabel 2.3 Simbol-simbol dalam *Activity Diagram*

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

(Sumber: Sukamto dan Shalahuddin,2014:162)

2.2.5. Sequence Diagram

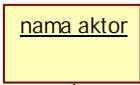
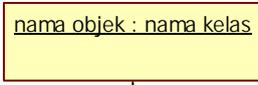
Sukamto dan Shalahuddin (2014:161), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan ,mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang



diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*.

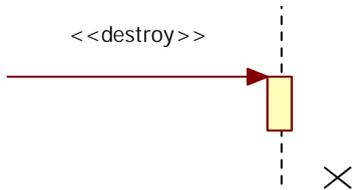
Banyak diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>nama aktor</p> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>menyatakan kehidupan suatu objek.</p>
<p>Objek</p>  <p>nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>



Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

<p>Pesan tipe create</p> <p style="text-align: center;"><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe call</p> <p style="text-align: center;">1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesan tipe send</p> <p style="text-align: center;">1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe return</p> <p style="text-align: center;">1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> <p style="text-align: center;"><<destroy>></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.</p>

(Sumber: Sukamto dan Shalahuddin,2014:165)

2.2.6. Pengertian Kamus Data

Kadir (2013:30), kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi yang berfungsi untuk mendefinisikan data yang mengalir pada sistem.



Sukamto dan Shalahuddin (2014:73), 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).

Tabel 2.5 Simbol-simbol dalam Kamus Data

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

(Sumber: Sukamto dan Shalahuddin, 2013:73)

2.3. Teori Program

2.3.1. Sekilas Tentang Android

2.3.1.1. Pengertian Android

Salbino (2014:7), *android* merupakan sistem operasi berbasis Linux yang bersifat terbuka (*Open Source*) dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet.

Tampilan android didasarkan pada manipulasi langsung, menggunakan masukan sentuhan yang serupa dengan tindakan didunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitn untuk memanipulasi objek dilayar.

2.3.1.2. Sejarah Android

Salbino (2014:8) menjelaskan sejarah *android*, pada bulan Oktober 2003 Android, Inc. didirikan di Palo Alto, California, oleh Andy Rubin (pendiri *Danger*), Rick Miner (pendiri *Wildfire Communication, Inc.*), Nick Sean(mantan VP T-Mobile), dan Chris White (kepala desain dan pengembangan antarmuka *WebTV*) untuk mengembangkan perangkat *smartphone* yang lebih sadar akan lokasi dan preferensinya. Awal tujuan pengembangan android yaitu untuk mengembangkan



sebuah sistem operasi canggih yang ditujukan untuk kamera digital namun pasar *smartphone* untuk menyaingi Symbian dan Windows Mobile (iPhone Apple belum dirilis saat itu). Android Inc. dioperasikan secara diam-diam, hanya para pengembang sedang menciptakan sebuah perangkat lunak untuk *smartphone*. Pada tahun yang sama, Rubin kehabisan uang. Steve Periman, seorang teman dekat Rubin, meminjaminya \$10.000 tunai dan menolak tawaran saham di perusahaan.

Google mengakuisisi Android Inc. pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki Google. Pendiri Android Inc. seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google. Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan platform *smartphone* menggunakan kernel linux. Google memasarkan platform tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bias diperbarui. Google telah menyeleksi beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerja sama ini terbuka bagi siapapun yang ingin berpartisipasi.

Pada tanggal 5 November 2007, Open Handset Alliance (OHA) didirikan, OHA bertujuan untuk mengembangkan standard terbuka bagi perangkat seluler. Saat itu, Android diresmikan sebagai produk pertamanya; sebuah platform perangkat seluler yang menggunakan kernel Linux versi 2.6. Telepon seluler komersial pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008.

Pada tahun 2010, Google merilis seri Nexus; perangkat *smartphone* dan tablet dengan OS Android yang diproduksi oleh HTC, LG, dan Samsung. HTC bekerjasama dengan Google dalam merilis produk *smartphone* Nexus pertama, yakni Nexus One. Seri ini telah diperbaharui dengan perangkat yang lebih baru, misalnya telepon pintar Nexus 4 dan tablet Nexus 10 yang diperbaharui oleh LG dan Samsung. Pada 15 Oktober 2014, Google mengumumkan Nexus 6 dan Nexus 10 yang diproduksi oleh Motorola dan HTC. Pada 13 Maret 2013, Larry Page memberitahukan dalam postingan blognya bahwa Andy Rubin telah pindah dari divisi android untuk mengerjakan proyek-proyek baru di Google. Ia digantikan oleh



Sundar Pichai, yang sebelumnya menjabat sebagai kepala divisi Google Chore, yang mengembangkan Chrome OS.

Sejak tahun 2008, Android terus melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi. Setiap versi utama meningkatkan kinerja sistem operasi. Setiap versi utama yang dirilis dinamakan secara alfaabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula; misalnya, versi 1.5 bernama Cupcake, yang kemudian diikuti oleh 1.6 Donut. Versi terbaru adalah 5.0 Lollipop, yang dirilis pada 15 Oktober 2014.

2.3.1.3 Versi Android

Android terus berkembang dan hal itu ditandai dengan versinya. Versi pertama hingga buku ini ditulis dapat dilihat pada tabel berikut.

Tabel 2.6 Versi Android

Versi	Nama	Tanggal Rilis	Level API
1.0	Tanpa nama	23 September 2009	1
1.1	Tanpa nama	9 Februari 2009	2
1.5	Cupcake	30 April 2009	3
1.6	Donut	15 September 2009	4
2.0	Eclair	26 Oktober 2009	5
2.1	Eclair	12 Januari 2010	7
2.2	Froyo	20 Mei 2010	8
2.3	Gingerbread	6 Desember 2010	10
3.0	Honeycomb	22 Februari 2011	11
3.1	Honeycomb	10 Mei 2011	12
3.2	Honeycomb	15 Juli 2011	13
4.0	Ice Cream Sandwich	19 Oktober 2011	14 (Versi 4.0.1-4.02) 15 (Versi 4.0.3-4.0.4)
4.1	Jelly Bean	9 Juli 2012	16
4.2	Jelly Bean	13 November 2012	17

(Sumber: Kadir, 2013:3)



Pembaruan versi terbaru *android*:

Versi	Nama	Tanggal Rilis	Level API
4.3	Jelly Bean	24 Juli 2013	18
4.4	KitKat	31 Oktober 2013	19
5.0	Lollipop	3 November 2014	20

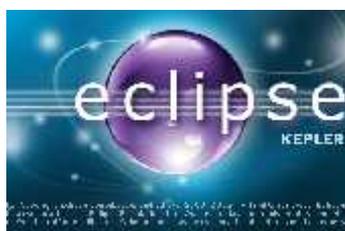
2.3.2. Sekilas Tentang Java

2.3.2.1. Pengertian Java

Asropudin (2013:52), java merupakan bahasa pemrograman untuk menciptakan isi yang aktif dalam halaman Web, juga dapat dijalankan dalam semua komputer.

2.3.2.2. Pengertian *Eclipse*

Kadir (2013:5), *eclipse* adalah perangkat aplikasi yang tergolong sebagai IDE (*integrated development environment*), karena menyediakan berbagai fasilitas untuk pembuatan aplikasi.



Gambar 2.1 *Eclipse*

2.3.2.3. Pengertian *Java Development Kit (JDK)*

Kadir (2013:4), *Java Development Kit (JDK)* adalah perangkat pengembangan aplikasi Java yang bisa diunduh secara gratis, perangkat ini mutlak diperlukan untuk membuat aplikasi Android.

2.3.2.4. Pengertian *Android SDK (Software Development Kit)*

Kadir (2013:5), *android SDK* adalah kumpulan *software* yang berisi mengenai pustaka, *debugger* (alat pencari kesalahan program), emulator (peniru perangkat bergerak), dokumentasi, kode contoh, dan panduan.



2.3.2.5. Pengertian *Android Development Tools*

Kadir (2013:6), *Android Development Tools* (ADT) adalah *plugin* untuk Eclipse IDE yang memungkinkan Eclipse digunakan untuk mengembangkan aplikasi Android.

2.3.2.6. Pengertian *Android Virtual Device* (AVD)

Kadir (2013:15), *Android Virtual Device* (AVD) atau dikenal dengan sebutan emulator perlu dibuat terlebih dahulu untuk kepentingan pengujian aplikasi Android.

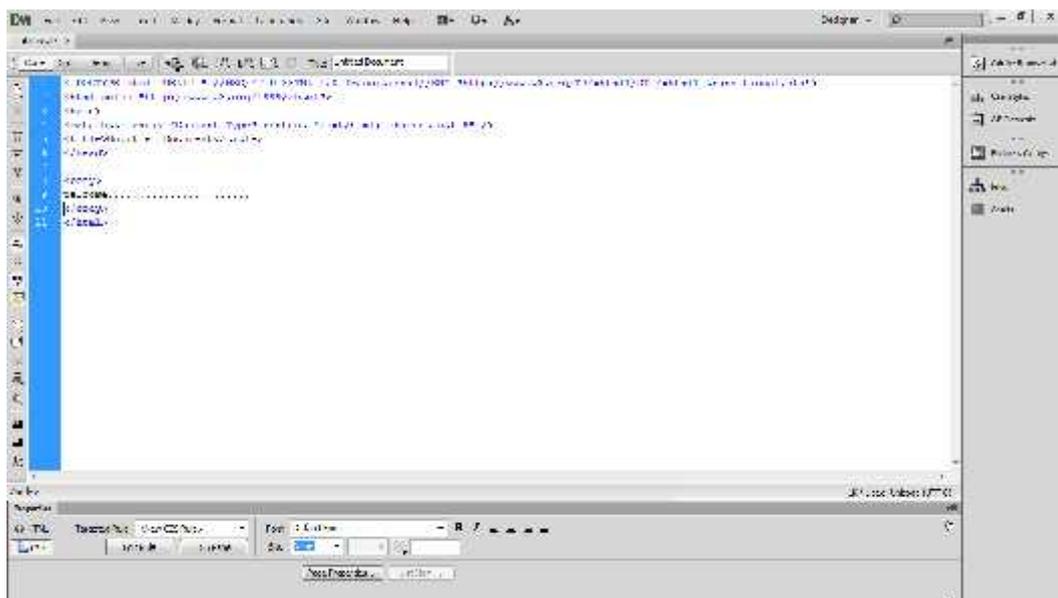
2.3.3. Sekilas Tentang *Adobe Dreamweaver CS5*

2.3.3.1. Pengertian *Adobe Dreamweaver CS5*

Madcoms (2011:2), *Adobe Dreamweaver CS5* adalah sebuah *HTML editor professional* untuk mendesain *web* secara visual dan mengelola situs atau halaman *web*. *Dreamweaver* merupakan *software* utama yang digunakan oleh *web* Desainer maupun *Web Programmer* dalam mengembangkan suatu situs *web*.

2.3.3.2. Area Kerja *Adobe Dreamweaver CS5*

Area kerja *Adobe Dreamweaver CS5* adalah lingkungan yang digunakan untuk merancang halaman *web*.



Gambar 2.2 Tampilan area kerja *dreamweaver CS5*



2.3.4. Sekilas Tentang PHP

2.3.4.1. Pengertian PHP (*Hypertext Preprocessor*)

Kadir (2013:47), PHP adalah bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*, karena pemrosesan program PHP dilakukan di lingkungan *web browser*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*). Oleh sebab itu, seperti yang dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “*view source*” pada *web browser* yang mereka gunakan.

Mundzir (2014:7), PHP berasal dari kata “*Hypertext Preprocessor*”, yaitu bahasa pemrograman *universal* untuk penanganan pembuatan dan pengembangan sebuah situs *web* dan bisa digunakan bersamaan dengan HTML.

Sianipar (2015:1), PHP merupakan bahasa skrip yang ditanam dalam HTML. Ini berarti bahwa Anda dapat menggabungkan kode PHP dan HTML dalam *file* yang sama.

2.3.4.2. Sintak Dasar PHP

Untuk menuliskan sintak pemrograman *PHP*, membutuhkan program editor text tambahan. Salah satu editor text yang paling umum dipakai adalah Notepad atau Notepad ++. Ketika menuiskan program *PHP* selalu diawali dan diakhiri menggunakan tag khusus. Tag khusus tersebut memiliki 4 cara penulisan, yaitu:

1. Cara I

```
<? echo “Sintak PHP”; ?>
```

2. Cara II

```
<? Php echo “Sintak PHP”; ?>
```

3. Cara III

```
<?= echo “Sintak PHP”; ?>
```

4. Cara IV

```
<% echo “Sintak PHP” %>
```



Contoh:

```
<?php
```

```
echo "Hanya untuk Test";
```

```
//statement ini diakhiri dengan titik koma
```

```
Echo"untuk Test saja"
```

```
/*Statement ini tidak diakhiri dengan titik koma */
```

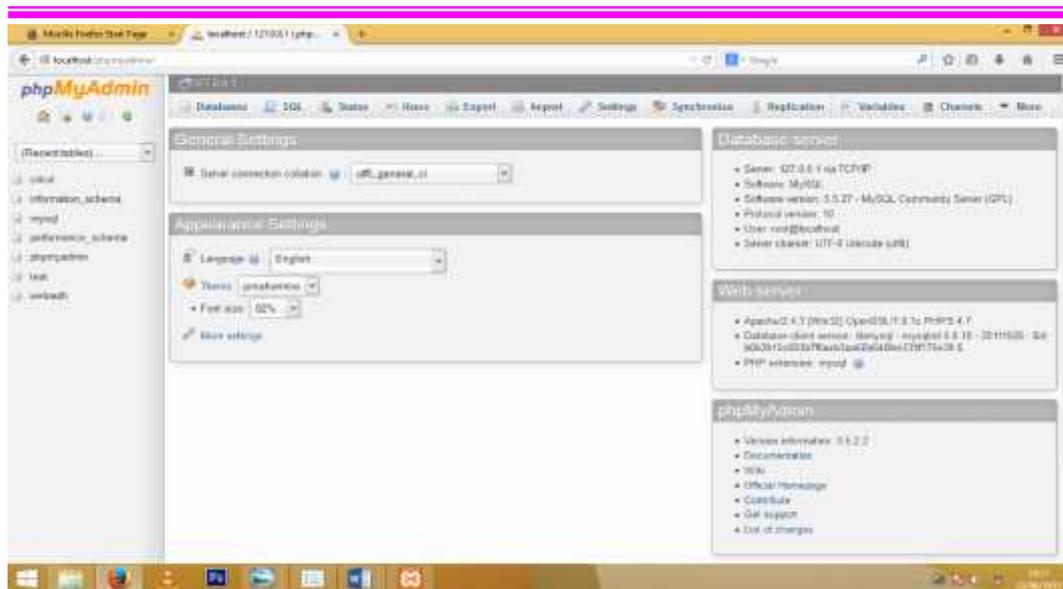
```
?>
```

Program tersebut disimpan di C:\xampp\htdocs dengan nama (misalnya latihan1.php). Kemudian dibuka pada web browser (misalnya, Internet Explorer atau Mozilla Firefox) dan isi alamat sebagai berikut: <http://localhost/latihan1.php>.

2.3.5. Sekilas Tentang XAMPP

2.3.5.1. Pengertian XAMPP

Menurut Buana (2014:4), *xampp* adalah perangkat lunak *opensource* yang bisa diunggah secara gratis dan bias dijalankan di semua sistem operasi seperti *Windows*, *Linux*, *Solaris* dan *Mac*. Xampp ini sendiri dibuat dan dikembangkan oleh *Apache Friends* Perangkat lunak tersebut berisi beberapa kumpulan perangkat lunak yang dibutuhkan, antara lain PHP, *Apache*, *MySQL*, dan *PHPMyAdmin*. Dengan melakukan instalasi Xampp, maka semua perangkat lunak yang dibutuhkan sudah terinstal di hardisk, termasuk file-file yang dibutuhkan. Xampp sendiri semakin waktu semakin berkembang, dimana setiap perubahan versi, dibuat lebih baik dan lebih lengkap dari versi sebelumnya.



Gambar 2.3 Tampilan *PhpMyAdmin*

2.3.6. Sekilas Tentang *MySQL*

2.3.6.1 Pengertian *MySQL*

Winarno, dkk. (2014:102), *MySQL* adalah sebuah software database yang merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk table-table yang saling berhubungan.

Enterprise (2014:2), *MySQL* adalah RDBMS (*Relation Database Management System*) yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

2.3.6.2. Fungsi-Fungsi *MySQL*

Buana (2014:93), menjelaskan tentang sejumlah fungsi yang berawalan `mysql_` yang digunakan untuk mengakses database server *MySQL*.

Tabel 2.7 Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan
1	<code>mysql_connect()</code>	Membuka koneksi ke server <i>MySQL</i>
2	<code>mysql_quit()</code>	Memutuskan koneksi dengan <i>MySQL</i>
3	<code>mysql_create_db()</code>	Membuat database <i>MySQL</i>
4	<code>mysql_drop_db()</code>	Menghapus database <i>MySQL</i>
5	<code>mysql_use()</code>	Memanggil database yang sudah ada



Lanjutan Tabel 2.7 Fungsi-Fungsi MySQL

6	<i>mysql_query()</i>	Menjalankan query atau perintah <i>MySQL</i>
7	<i>mysql_create_table()</i>	Membuat tabel <i>MySQL</i>
8	<i>mysql_describe()</i>	Mengecek table
9	<i>mysql_drop_table()</i>	Menghapus tabel <i>MySQL</i>
10	<i>mysql_alter()</i>	Mengubah struktur table <i>MySQL</i>
11	<i>mysql_add()</i>	Menambah kolom baru pada table
12	<i>mysql_change()</i>	Mengubah urutan posisi kolom pada tabel
13	<i>mysql_insert()</i>	Menambah data table
14	<i>mysql_update()</i>	Mengubah isi data table
15	<i>mysql_select</i>	Menampilkan data pada table yang sudah dibuat

(Sumber: Buana, 2014:93)