



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

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

Hal diatas senada dengan pendapat Wahyudi (2012:3), menyatakan bahwa komputer adalah peralatan (device) yang menerima data (input) dan menyimpan (storage) kemudian diproses (process) untuk menghasilkan data dalam bentuk lain (output).

Dari definisi diatas, dapat disimpulkan bahwa komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan agar secara otomatis menerima dan menyimpan data input, memprosesnya, dan menghasilkan output.

2.1.2. Pengertian Perangkat Lunak (*Software*)

Sukamto et.al (2013:2), “Perangkat lunak (software) adalah program computer terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (user manual)”.

Sutarman (2012:14), “Software adalah kumpulan program-program komputer yang memungkinkan hardware memproses data”.

Dari definisi diatas, dapat disimpulkan bahwa software adalah sebuah data yang diprogram dan disimpan secara digital dalam bentuk aplikasi, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca dan ditulis oleh komputer.

2.1.3. Pengertian Data

Ladjamudin (2013:8), “data adalah deskripsi dari sesuatu dan kejadian yang kita hadapi (*the description of things and events thet we face*)”.



Asropudin (2013:20), “data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata, dan merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi.

2.1.4. Pengertian Pengolahan Data

Menurut Ladjamudin (2013:9), “Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

Hal diatas dapat disimpulkan bahwa pengolahan data merupakan kegiatan manipulasi data yang diproses sehingga menghasilkan perubahan dalam bentuk informasi.

2.1.5. Pengertian *Framework*

Menurut Hakim (2010:3), “*Framework* dapat diartikan sebagai koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal”.

Pernyataan diatas dapat simpulkan bahwa *framework* merupakan sekumpulan fungsi-fungsi yang telah diorganisasikan sedemikian rupa sehingga dapat membantu *programmer* dalam membuat sebuah program tanpa membuatnya dari awal.

2.1.6. Pengertian *CodeIgniter*

Hakim (2010:8), “*CodeIgniter* adalah sebuah framework *PHP* yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis *PHP* dibandingkan jika menulis semua kode program dari awal”.



Noviyanto (2010:Page I), menyatakan bahwa *CI* (Kependekan dari *CodeIgniter*) merupakan suatu *Application Development Framework* atau secara mudahnya adalah seperangkat alat (*toolkit*) untuk membantu kita dalam membangun *web base application* (aplikasi berbasis web) dengan menggunakan bahasa pemrograman *PHP*.

Pernyataan diatas dapat disimpulkan bahwa *CodeIgniter* adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis web dengan menggunakan bahasa pemrograman *PHP*.

2.1.7. Metodologi Pengembangan Sistem

Sukamto et.al (2013:124), *Unified Process* atau dikenal juga dengan proses iteratif dan *incremental* merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan inkremental (bertahap dengan progres menaik). Iteratif bisa dilakukan didalam setiap tahap, atau iteratif tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental (bertambah menaik) dimana setiap iterasi akan memperbaiki iterasi berikutnya. Salah satu *Unified Process* yang terkenal adalah RUP (*Rational Unified Process*).

Sukamto et.al (2013:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh *Rational Software* yang diakuisisi oleh IBM di bulan Februari 2003.”

Sukamto et.al (2013:125), RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif. Berikut ini penjelasan untuk setiap fase pada RUP.



2.1.7.1. Inception (Permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2.1.7.2. Elaboration (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang focus pada purwarupa sistem (*prototype*).

2.1.7.3. Construction (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

2.1.7.4. Transition (Transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.1.8. Studi Kelayakan

Studi Kelayakan adalah proses evaluasi untuk menetapkan apakah program tersebut layak digunakan atau tidak.

Sutabri (2012:59), Adapun kelayakan yang dinilai dalam proyek pengembangan sistem informasi yang dimaksud adalah sebagai berikut:



1. Kelayakan operasional

Menyangkut apakah secara operasional sistem yang baru dapat dilaksanakan dengan sumber daya manusia yang tersedia dan metode training yang ditawarkan, pelayanan purna jual atau pemeliharaan serta efisiensi dan efektifitas sistem yang baru.

2. Kelayakan teknis

Menyangkut apakah *hardware /software* yang akan dikembangkan tersedia, jadwal pelaksanaan serta sistem keamanan data.

3. Kelayakan ekonomis

Menyangkut biaya untuk membuat dan menjalankan sistem baru serta keuntungan yang akan diperoleh dari sistem tersebut.

2.1.9. Pengujian Sistem

Sukamto et.al (2013:275), menyatakan bahwa *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.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan salah.

2.2. Pengertian Judul

2.2.1. Pengertian Sistem

Pertama menurut Sutarman (2012:5), “Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama”.

Sutabri (2012:22), “Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu”.



Sukanto et.al (2013:291), “Sistem adalah kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.”

Kesimpulan dari hal diatas yaitusekelompok komponen dan elemen yang saling berinteraksi untuk mencapai tujuan tertentu.

2.2.2. Pengertian Informasi

Sutarman (2012:14), “Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima”.

Kristanto (2008:7), “Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima”.

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

Secara keseluruhan, informasi merupakan kumpulan elemen, fakta (data) yang diolah dan diorganisasikan sehingga memberikan bentuk yang berguna dan memiliki arti bagi penerima.

2.2.3. Pengertian Sistem Informasi

Menurut Ladjamudin (2013:13), “Sistem Informasi dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.”

Jadi, sistem informasi merupakan suatu sistem yang terdiri dari komponen yang diorganisasikan untuk mengumpulkan, memproses menyimpan, menganalisis dan menyebarkan informasi.

2.2.4. Pengertian Koperasi

Pengertian koperasi menurut Kamus Besar Bahasa Indonesia (2016:451), berasal dari kata koperasi yang artinya bekerjasama. Sehingga istilah koperasi adalah melakukan pekerjaan secara bersama (gotong-royong).



2.2.5. Pengertian Simpan

Kamus Besar Bahasa Indonesia (2016:710), “Menaruh secara rapi dan terpelihara di tempat aman”.

2.2.6. Pengertian Pinjam

Kamus Besar Bahasa Indonesia (2016:613), “Memakai sesuatu milik orang lain untuk sementara waktu dan kalau sudah waktunya dikembalikan (tentang uang, barang, dsb)”.

2.2.7. Pengertian Bank

Kamus Besar Bahasa Indonesia (2016:102), “Lembaga yang menangani masalah keuangan termasuk mengatur lalu lintas keuangan”.

2.2.8. Pengertian Muamalat

Kamus Besar Bahasa Indonesia (2008:974), “Segala sesuatu yang berhubungan dengan amal kita terhadap masyarakat”.

2.2.9. Pengertian Indonesia

Kamus Besar Bahasa Indonesia, “Indonesia adalah nama negara kepulauan di Asia Tenggara yang terletak diantara benua Asia dan benua Australia”.

2.2.10. Pengertian Cabang

Kamus Besar Bahasa Indonesia (2016:170), “Bagian dari usaha atau perusahaan”.

2.2.11. Pengertian Judul Secara Keseluruhan

Dari uraian diatas dapat disimpulkan bahwa Sistem Informasi Koperasi Simpan Pinjam pada Bank Muamalat Indonesia Cabang Palembang adalah suatu sistem yang berfungsi untuk mengolah data koperasi simpan pinjam dan menyajikan informasi secara efektif dan efisien untuk mempermudah anggota



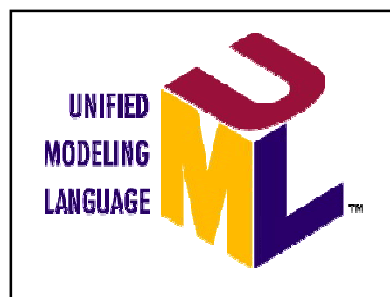
dalam melakukan proses simpan pinjam yang terdapat dalam koperasi Bank Muamalat Indonesia cabang Palembang.

2.3. Teori Khusus

2.3.1. UML (*Unified Modeling Language*)

Sukamto et.al(2013:133), “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.



Gambar 2.1. Tampilan Logo UML

2.3.2. Jenis-Jenis Diagram UML

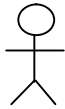
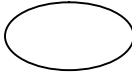

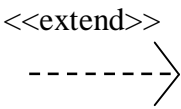

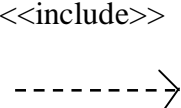
2.3.2.1. Use case Diagram

Sukamto et.al(2013:155), “*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.

Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.1. Simbol-simbol *Use case* Diagram

No.	Simbol	Nama	Deskripsi
1		<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.
2		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: Sukamto et.al(2013:156-158)

Komponen pembentuk diagram *use case* adalah:

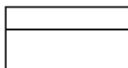


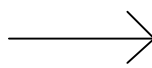
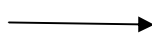


1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use case*, aktivitas / sarana yang disiapkan oleh bisnis / sistem.
3. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

2.3.2.2. Class Diagram

Sukamto et.al(2013:141), “*class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Adapun simbol-simbol yang digunakan dalam *classdiagram* adalah sebagai berikut:

Tabel 2.2. Simbol-simbol *Class Diagram*

No	Gambar	Nama	Deskripsi
1		<i>Class</i>	Kelas pada stuktur sistem.
2		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber: Sukamto et.al(2013:146-147)






2.3.2.3. Activity Diagram

Sukamto et.al(2013:161) “*activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.


Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

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

Tabel 2.3. Simbol-simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
3		Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir	Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Lanjutan Tabel 2.3. Simbol-simbol *Activity Diagram*

6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
---	---	-----------------	--



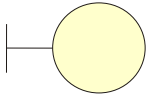

Sumber: Sukamto et.al(2013:162-163)

2.3.2.4. *Sequence Diagram*



Sukamto et.al(2013:165) , “diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

Sequence diagram menunjukkan urutan *event* kejadian dalam suatu waktu. Komponen *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. Simbol-simbol yang digunakan dalam *sequence* diagram adalah:

Tabel 2.4. Simbol-simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>An Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
2		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari <i>form</i>
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel

Lanjutan Tabel 2.4. Simbol-simbol *Sequence Diagram*

5		<i>A focus of control</i>	Menggambarkan tempat mulai dan berakhirnya sebuah <i>message</i> (pesan)
6		<i>A line of life</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber: Sukamto et.al(2013:165-166)

2.4. Teori Program

2.4.1. Basis Data (*Database*)

Sukamto et.al(2013:43), “basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Sutarman (2012:15), “*database* merupakan sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan diantaranya”.

Jadi, *database* atau basis data merupakan kumpulan data-data yang saling berhubungan dan terorganisasi yang disimpan dalam media pengingat (*hard disk*) agar dapat diakses dengan mudah dan cepat.

2.4.2. *PHP*

2.4.2.1. Pengertian *PHP*

Sidik (2012:5), “*PHP* adalah kependekan dari *PHP:HyperText Preprocessor* (rekursif, mengikut gaya penamaan di **nix*), merupakan bahasa utama *script server-side* yang disisipkan pada *HTML* yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi *desktop*”.

Nugroho (2013:153), “*PHP* adalah bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*website, blog* atau *aplikasi web*)”.

Kadir (2008:2), “*PHP* merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses



di server. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*”.

Jadi, *PHP (HyperText Preprocessor)* adalah bahasa pemrograman berbasis *WEB* yang dijalankan di server dan bisa digunakan untuk membuat aplikasi *desktop*.



Gambar 2.2 Tampilan Logo *PHP*

2.4.2.2. *ScriptPHP*

Kadir (2008:3) Skrip PHP berkedudukan sebagai tag dalam bahasa HTML. Sebagaimana diketahui, HTML (*HyperText Markup Language*) adalah bahasa standar untuk membuat halaman-halaman web. Sebagai contoh, berikut adalah kode HTML (disimpan dengan ekstensi *.htm* atau *.html*):

```
<HTML>
  <HEAD>
    <TITLE> Latihan Pertama </TITLE>
  </HEAD>
  <BODY>
    Selamat Belajar PHP.<BR>
  </BODY>
</HTML>
```

Adapun kode berikut adalah contoh kode PHP yang berada di dalam kode HTML:

```
<HTML>
  <HEAD>
```



```

<TITLE> Latihan Pertama </TITLE>
</HEAD>
<BODY>
Selamat Belajar PHP.<BR>
<?php
    printf("Tgl. Sekarang: %s ", Date("d F Y"));
?>
</BODY>
</HTML>

```

Kode di atas disimpan dengan ekstensi.php.

Perhatikan baris-baris berikut:

```

<?php
    printf("Tgl. Sekarang: %s ", Date("d F Y"));
?>

```

2.4.3. MySQL

Kadir (2013:2), "MySQL (baca: mai-se-kyu-el) merupakan *software* yang tergolong *DBMS(Database Management System)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara *men-download* (mengunduh) di internet secara gratis".

Anhar (2010:21), "MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data *SQL (Database Manajemen System)* atau *DBMS* dari sekian banyak *DBMS*, seperti *ORACLE, MS SQL, Postagre SQL*, dan lain-lain".

MySQL merupakan *DBMS* yang *multithread, multi-user* yang bersifat gratis dibawah lisensi *GNU General Public Licence (GPL)*.

Kelebihan dari *MySQL* adalah dapat digunakan untuk aplikasi *multi-user* (banyak pengguna) dan menggunakan bahasa *query* (permintaan) standar *SQL* (*Structured Query Language*).

Kekurangan dari *MySQL* adalah untuk koneksi ke bahasa pemrograman *visual* seperti *VB*, *Delphi*, dan *Foxpro*, *MySQL* kurang *support*, karena koneksi ini menyebabkan *field* yang dibaca harus sesuai dengan koneksi dari program visual tersebut, dan ini yang menyebabkan *MySQL* jarang dipakai dalam program visual serta data yang ditangani belum begitu besar.

2.4.3.1. Fungsi-fungsi *MySQL*

Untuk dapat menggunakan semua perintah *SQL* pada *PHP*, kita memerlukan fungsi bawaan *PHP* (*reserved word*) dengan beragam kemampuan, mulai dari menampilkan informasi, memanipulasi informasi, keperluan kondisional, keperluan perulangan, sampai pengaksesan berbagai *database*, termasuk *MySQL*. Berikut adalah daftar beberapa fungsi *PHP* yang secara khusus digunakan untuk mengakses dan mengelola *databaseMySQL*.

Tabel 2.5. Fungsi-fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
1	<i>MySQL_connect()</i>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu <i>host</i> .	<i>mysql_connect</i> (<i>host</i> , <i>nama_pemakai</i> , <i>password</i>)
2	<i>MySQL_close()</i>	Menutup hubungan ke <i>database MySQL</i> .	<i>mysql_close</i> (<i>pengenal_hubungan</i>)
3	<i>MySQL_select_db()</i>	Memilih <i>database</i>	<i>mysql_select_db</i> (<i>database</i> , <i>pengenal_hubungan</i>)



Lanjutan Tabel 2.5. Fungsi-fungsi *MySQL*

4	<i>MySQL_query()</i>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel.	<i>mysql_query</i> (<i>permintaan</i> , <i>pengenal_hubungan</i>)
5	<i>MySQL_db_query()</i>	Menjalankan suatu permintaan terhadap suatu <i>database</i> .	<i>mysql_db_query</i> (<i>database</i> , <i>permintaan</i> , <i>pengenal_hubungan</i>)
6	<i>MySQL_num_rows()</i>	Memperoleh jumlah baris dari suatu hasil permintaan (<i>query</i>) yang menggunakan SELECT .	<i>mysql_num_rows</i> (<i>pengenal_hasil</i>)
7	<i>MySQL_affected_rows</i> (<i>)</i>	Memperoleh jumlah baris yang dikenai operasi INSERT , DELETE , dan UPDATE .	<i>mysql_affected_rows</i> (<i>[pengenal_hubungan]</i>)
8	<i>MySQL_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan.	<i>mysql_num_fields</i> (<i>pengenal_hasil</i>)
9	<i>MySQL_fetch_row()</i>	Menghasilkan suatu array yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil.	<i>mysql_fetch_row</i> (<i>pengenal_hasil</i>)



Lanjutan Tabel 2.5. Fungsi-fungsi MySQL

10	<i>MySQL_fetch_array()</i>	Mempunyai kegunaan serupa dengan <i>mysql_fetch_row()</i> . Hanya saja, setiap kolom akan disimpan dua kali pada array hasil.	<i>mysql_fetch_array</i> (<i>pengenal_hasil</i>)
11	<i>MySQL_fetch_field()</i>	Memperoleh informasi suatu kolom.	<i>mysql_fetch_field</i> (<i>pengenal_hasil</i> [, <i>nomor_kolom</i>])
12	<i>MySQL_data_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke baris tertentu.	<i>mysql_data_seek</i> (<i>pengenal_hasil</i> , <i>nomor_baris</i>)
13	<i>MySQL_field_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke kolom tertentu.	<i>mysql_data_seek</i> (<i>pengenal_hasil</i> , <i>nomor_kolom</i>)
14	<i>MySQL_create_db()</i>	Menciptakan <i>databaseMySQL</i> .	<i>mysql_create_db</i> (<i>database</i> [, <i>pengenal_hubungan</i>])
15	<i>MySQL_drop_db()</i>	Menghapus <i>databaseMySQL</i> .	<i>mysql_drop_db</i> (<i>database</i> [, <i>pengenal_hubungan</i>])

Lanjutan Tabel 2.5. Fungsi-fungsi *MySQL*

16	<i>MySQL_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i> .	<code>mysql_list_dbs ([pengenal_hubungan])</code>
17	<i>MySQL_list_tables()</i>	Memperoleh daftar nama tabel dalam suatu <i>database</i> .	<code>mysql_list_tables (database [, pengenal_hubungan])</code>
18	<i>MySQL_list_fields()</i>	Memperoleh daftar nama kolom dalam suatu <i>database</i> .	<code>mysql_list_fields (database [, pengenal_hubungan])</code>

Sumber: Kadir (2008:360-379)

2.4.4. Adobe Dreamweaver Macromedia 8

Hadi (2006:2)“*Dreamweaver8* adalah sebuah program web editor yang digunakan untuk membuat dan mendesain *web*dengan mudah”.

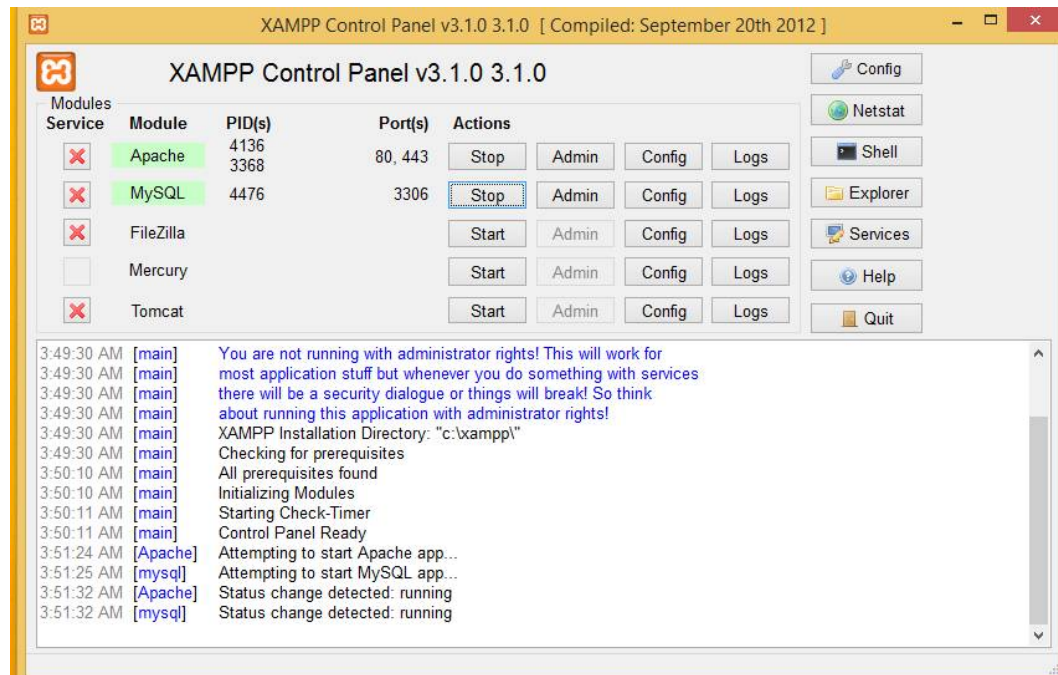


Gambar 2.3. Logo *Dreamweaver Macromedia 8*

2.4.5. Pengertian XAMPP

Sidik (2012:72) menyatakan, “XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl) merupakan paket *server web* PHP dan *database MySQL* yang paling populer di kalangan pengembang *web* dengan menggunakan PHP dan MySQL sebagai databasenya.

Nugroho (2013:1), “XAMPP adalah paket program web lengkap yang dapat anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legal. Pada saat buku ini dibuat, XAMPP sudah merilis versi XAMPP 1.7.7.



Gambar 3.3 Tampilan Awal XAMPP

Filename: BAB II
Directory: C:\Users\Acer\Documents
Template: C:\Users\Acer\AppData\Roaming\Microsoft\Templates\Normal.dot
m
Title:
Subject:
Author: vista
Keywords:
Comments:
Creation Date: 08/07/2016 19:49:00
Change Number: 19
Last Saved On: 29/09/2016 13:06:00
Last Saved By: ROSADI
Total Editing Time: 182 Minutes
Last Printed On: 29/09/2016 13:08:00
As of Last Complete Printing
Number of Pages: 20
Number of Words: 3.074
Number of Characters: 20.923