

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Aji R.B (2014) menyatakan SMA Kesatrian 1 Semarang sebagai suatu instansi pendidikan yang sedang berkembang selalu ingin mengikuti perkembangan teknologi informasi yang sedang berkembang, namun dalam hal pencatatan administasi keuangan siswa sekolah ini masih menggunakan cara yang manual, dengan cara pencatatan dan penyimpanan data transaksi dalam bentuk arsip, sehingga menyebabkan kesulitan bagi karyawan tata usaha dalam pencarian data-data transaksi pembayaran SPP juga dalam hal registrasi siswa, terutama pada saat data atau dokumen semakin banyak, sehingga sering terjadi penumpukan data siswa dan mengakibatkan sulitnya pencarian data siswa apabila sewaktu-waktu data itu diperlukan.

Selain itu, keadaan tersebut juga menyebabkan kurang efektifnya pembuatan laporan-laporan. Proses pembuatan laporan membutuhkan waktu yang relatif lama, karena harus membuat rekapitulasi dari dokumen-dokumen tersebut. Akan tetapi, meskipun proses rekapitulasi dilakukan, laporan-laporan yang disajikan pun sering tidak akurat. Sebagai seorang mahasiswa Informatika, penulis tertarik untuk meneliti dan mengembangkan sistem pembayaran SPP berbasis komputer pada SMA Kesatrian 1 Semarang, dengan harapan dapat membantu aktivitas sekolah terutama bagian tatausaha dalam mengetahui keuangan pembayaran registrasi siswa dan pembayaran SPP dan juga menyajikannya ke dalam bentuk laporan registrasi siswa dengan cepat, mudah, dan akurat, serta proses registrasi siswa dan pembayaran SPP akan berjalan lebih efektif dan efisien.

Rendy M.G (2016) menyatakan bahwa sistem pembayaran adminitrasi sekolah di SMK Kesatrian Solo Sukoharjo masih menggunakan program Microsoft Excel, administrasi dipisahkan untuk setiap tingkat dan di setiap tingkat dipegang oleh admin yang berbeda sehingga kurang mampu dalam memberikan informasi secara cepat dan tepat. Berdasarkan kondisi tersebut

penulis merancang dan membangun sebuah sistem pembayaran administrasi sekolah berbasis web, sehingga informasi pembayaran, maupun kekurangan administrasi siswa dapat diketahui secara cepat, akurat, dan tepat.

Abidin (2014) menyatakan Yayasan Az-Zahra adalah sebuah yayasan yang memiliki berbagai satuan pendidikan dan jumlah siswa yang banyak sehingga diperlukan layanan terbaik di semua bidang khususnya di bidang administrasi keuangan. Banyaknya transaksi SPP dan pembayaran lainnya dari beberapa jenjang pendidikan yang berbeda setiap hari dan hanya diterima seorang administrasi keuangan menyebabkan kesulitan dalam membuat laporan keuangan, terlebih pencatatan dan pelaporan masih menggunakan cara manual sehingga membutuhkan waktu lebih lama dalam pencatatan serta penyampaian laporan. Selain hal tersebut penempatan buku pembayaran SPP yang sering berpindah tempat, sehingga rawan hilang dan sabotase oleh pihak lain, Faktor lain yang diangkat penulis adalah Letak antara kantor keuangan dan sekolahan yang cukup jauh menyebabkan kepala Tata Usaha, serta Kepala Sekolah harus berjalan kaki untuk meminta laporan pembayaran SPP, oleh karena itu untuk meningkatkan efektifitas dan efisiensi kerja diperlukan sebuah sistem yang menyimpan data secara terpusat dan saling terintegrasi agar setiap proses transaksi, pencarian, maupun penyajian laporan menjadi data lebih cepat serta terjamin keamanannya maka diperlukan sebuah sistem pembayaran SPP berbasis *client server*.

Berdasarkan latar belakang dan beberapa masalah diatas maka penulis mencoba untuk menerapkan ilmu yang telah penulis pelajari selama kuliah dibidang teknologi informasi guna menyelesaikan masalah administrasi keuangandi Yayasan Az-Zahra Demak dengan Judul Penelitian “Aplikasi Pembayaran SPP Di Lingkungan Yayasan Az-Zahra Demak Berbasis *Client Server* Terintegrasi Dengan *SMS Gateway*”. Penulis berharap hasil penelitian dan program pembayaran SPP ini dapat mengatasi permasalahan-permasalahan yang selama ini terjadi di lingkungan Yayasan Az-Zahra Demak dan dapat meningkatkan layanan dalam hal administrasi keuangan.

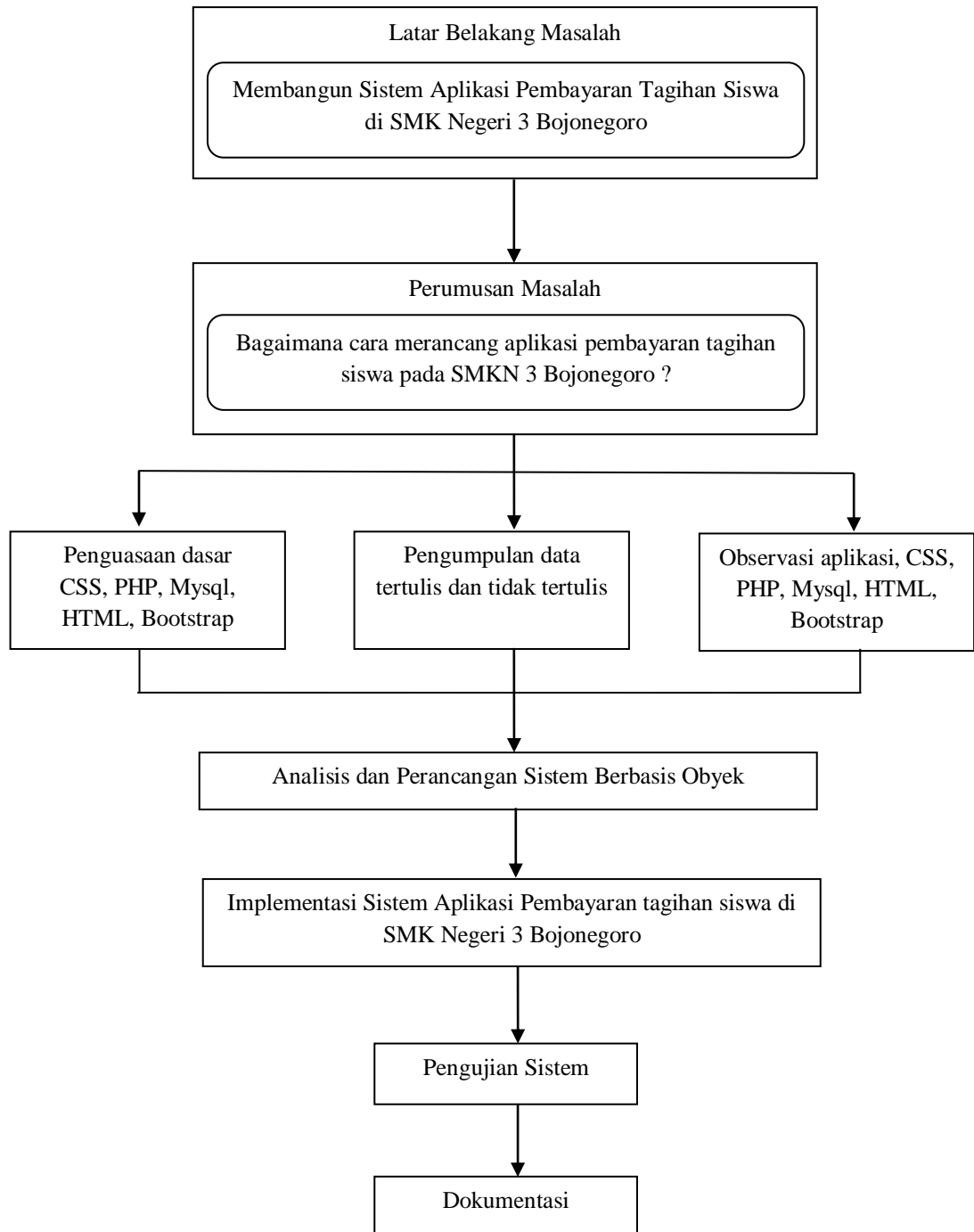
Berikut ini Tabel 2.1 yang menjelaskan tinjauan pustaka

Tabel 2.1 Tinjauan Pustaka

No	Judul dan Penulis	Tahun	Informasi Masalah	Metode Penyelesaian Aplikasi	Hasil
1.	Sistem pembayaran SPP berbasis komputer pada SMA Kesatrian I Semarang (Aji Raino Baswananda)	2014	SMA Kesatrian 1 Semarang dalam hal pencatatan administasi keuangan siswa masih menggunakan cara yang manual, dengan cara pencatatan dan penyimpanan data transaksi dalam bentuk arsip sehingga menyebabkan kesulitan bagi karyawan tata usaha dalam pencarian data-data transaksi pembayaran SPP	Metode Pengumpulan data : wawancara, observasi, studi kepustakaan Jenis Data : Kuantitatif dan kualitatif Sumber Data : Data primer, data sekunder Metode pengembangan Sistem : SDLC	Penulis mengembangkan sistem pembayaran SPP berbasis komputer pada SMA Kesatrian 1 Semarang, dengan harapan dapat membantu aktivitas sekolah terutama bagian tatusaha dalam mengetahui keuangan pembayaran siswa
2.	Aplikasi Pembayaran SPP di Lingkungan Yayasan Az-Zahra Demak Berbasis <i>Client Server</i> Terintegrasi Dengan Sms Gateway (Rohmad Abidin)	2014	Yayasan Az-Zahra dalam hal pencatatan dan pelaporan administasi keuangan siswa masih menggunakan cara manual dengan buku spp siswa. Selain itu walimurid sering tidak mengetahui status pembayaran anak-anaknya.	Metode Pengumpulan data : observasi, wawancara, kuisisioner Analisis Data : Kuantitatif dan kualitatif Metode pengembangan sistem : Menggunakan model Penelitian Pengembangan <i>Research and Development (R&D)</i>	Penulis Merancang dan Membangun Aplikasi Pembayaran SPP Di Lingkungan Yayasan Az-Zahra Demak Berbasis <i>Client Server</i> Terintegrasi Dengan SMS Gateway
3.	Pembuatan Aplikasi Adminitrasi Pembayaran Berbasis Web di SMK Kasatrian Solo Sukoharjo (Rendy Muhamad Gunawan)	2016	Sistem pembayaran adminitrasi sekolah di SMK Kesatrian Solo Sukoharjo masih menggunakan program Microsoft Excel, administrasi dipisahkan untuk setiap tingkat dan di setiap tingkat dipegang oleh admin yang berbeda sehingga kurang mampu dalam memberikan informasi secara cepat dan tepat	Metode Pengumpulan data : observasi, wawancara, dan riset perpustakaan Metode pengembangan sistem : SDLC (<i>System Development Life Cycle</i>) Atau Metode <i>Waterfall</i> Sumber Data : Data primer, data sekunder	penulis merancang dan membangun sebuah sistem pembayaran administrasi sekolah berbasis web, sehingga informasi pembayaran, maupun kekurangan administrasi siswa dapat diketahui secara cepat, akurat, dan tepat.

2.2 Kerangka Berfikir

Penjelasan kerangka berfikir dapat ditunjukkan pada gambar 2.1



Gambar 2.1 Kerangka Berfikir

2.3 Teori Pendukung

2.3.1 *Hypertext Markup Language (HTML)*

HTML (*Hypertext Markup Language*) adalah bahasa standar yang digunakan untuk pembuatan halaman web atau *word wide web*, dengan *hypertext* dan informasi lain yang akan ditampilkan pada halaman web. Dokumen *hypertext* bisa berisi teks, gambar, dan tipe informasi lain seperti data file, audio, video, dan program *executeable*. (Sutarman 2012)

1. Fungsi dan Kegunaan HTML

- a.) Sebagai bahasa *markup* internet, HTML memiliki banyak fungsi dan kegunaan. Beberapa manfaat dari HTML diantaranya : Untuk membuat halaman web bahasa HTML digunakan untuk membuat halaman web.
- b.) Sebagai pondasi bagi sebuah website. Sebuah rumah jika tidak memiliki pondasi maka akan cepat roboh, begitu juga dengan website. Jika tidak memiliki HTML sebagai pondasi, kita tidak dapat mengimplementasikan bahasa lainnya seperti CSS (bahasa untuk mendesain website), Javascript (bahasa untuk menambah perilaku website), dan PHP (bahasa pemrograman server website).
- c.) Untuk menandai teks pada halaman web, misalnya, kita dapat menandai sebuah teks menjadi bergaris bawah dengan menggunakan tag html <u>
- d.) Untuk menandai elemen/bagian pada halaman web, sebuah website memiliki beberapa bagian seperti header, navigasi, main dan footer. Kita dapat menandai setiap bagiannya dengan HTML.
- e.) Untuk menampilkan informasi dalam bentuk tabel
- f.) Untuk menambahkan objek seperti audio, video, gambar, dll dalam halaman web
- g.) Untuk membuat *online form*.

2.3.2 *Hypertext Preprocessor (PHP)*

PHP (*Personal Home Page*) adalah pemrograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan (Sibero 2012).

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: *Hypertext Preprocessing*.

Pada pertengahan tahun 1999, *Zend* merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, *Zend* merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Server web bawaan ditambahkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall *software server*.

Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 2017.

2.3.3 *Bootstrap*

Bootstrap merupakan sebuah *framework css* yang memudahkan pengembang untuk membangun website yang menarik dan responsif. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk mengembangkan dan pemeliharannya. *Bootstrap* adalah css tetapi dibentuk dengan *LESS*, sebuah *pre-processor* yang memberi fleksibilitas dari css biasa. *Bootstrap* memberikan solusi rapi dan seragam terhadap solusi yang umum, tugas interface yang setiap pengembang hadapi. *Bootstrap* dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan design butuhkan (Otto, 2011).

Keunggulan dalam menggunakan *bootstrap* adalah semua bagian untuk antarmuka pengguna menggunakan *style css*, *Bootstrap* dapat menggunakan *LESS preprocessor* sebuah teknologi yang mengurangi dan mengefisienkan penulisan kode CSS. *Bootstrap* dapat diintegrasikan dengan JavaScript untuk menjadikan lebih menarik dengan efek-efek yang dapat diberikan dengan *JavaScript* (Tectale, 2012).

Kelemahan dalam menggunakan *bootstrap* adalah dengan adanya penggunaan *bootstrap* menjadi tidak adanya keunikan didalam website karena akan samanya tampilan yang diberikan. Terdapat juga laporan bahwa sistem *grid* pada *bootstrap* tidak responsif (Tectale, 2012)

2.3.4 *Database*

Database adalah sekumpulan data *store* (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *oftical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya (Ladjamudin, 2013).

1. MySQL

MySQL adalah suatu *Relational Database Management System* (RDBMS) yang mendukung database yang terdiri dari sekumpulan relasi atau tabel *Relational Database Management System* (RDBMS) yang tersedia, tetapi MySQL khususnya sangat cocok untuk bekerja sama dengan PHP. Bagaimanapun, konsep dasar SQL tetap sama apa pun jenis database yang anda

gunakan. MySQL dilepaskan dengan suatu lisensi *open source*, dan tersedia secara cuma-cuma. MySQL bekerja pada berbagai sistem operasi, dan banyak masalah. MySQL bekerja dengan cepat dan baik dengan data yang besar. PHP menyediakan banyak fungsi untuk mendukung database MySQL (Santoso, 2014).

2. Kelebihan MySQL dibanding dengan database *database* lain

- a.) Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
- b.) Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis.
- c.) Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- d.) *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- e.) Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
- f.) Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).
- g.) Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
- h.) Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- i.) Konektivitas MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau NT.

- j.) Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
- k.) Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- l.) Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
- m.) Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun Oracle.

3. Kekurangan MySQL dibanding *database* yang lain

- 1) Banyak mengklaim kurang support terhadap pemrograman *Visual/Desktop*, sehingga sedikit yang menggunakan untuk aplikasi visual.
- 2) Karena berlisensi GPL sehingga sulit mendapatkan update untuk problem yang *urgent*, sehingga perusahaan skala menengah keatas lebih memilih RDBMS berlisensi dan disupport seperti Oracle dan MS SQL Server
- 3) Sangat diragukan dalam menangani data skala besar, karena ada beberapa opini yang pro dan kontra terhadap kemampuan MySQL terhadap pengolahan data yang besar.

2.3.5 *PhpMyadmin*

PhpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui *www (World Wide Web)*. *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perizinan (*permissions*), dan lain-lain)

1. Fitur dalam phpMyAdmin :

- a.) Antarmuka berbasis web.
- b.) Dukungan banyak fitur MySQL:
 - 1.) Menelusuri dan drop basis data (database), tabel, pandangan (view), bidang (fields) dan indeks.
 - 2.) Membuat, menyalin, drop, dan mengubah nama basis data, tabel, kolom dan indeks.
 - 3.) Pemeliharaan server, basis data dan tabel, dengan server konfigurasi.
 - 4.) Mengelola pengguna MySQL dan hak istimewa.
 - 5.) Mengelola prosedur penyimpanan.
- c.) Impor data dari CSV dan SQL
- d.) Ekspor data ke berbagai format: CSV, SQL, XML, PDF, ISO / IEC 26300 – *Open Document Text* dan *Spreadsheet, Word, Excel, Lateks* dan lain-lain
- e.) Membuat grafik PDF dari tampilan basis data anda.
- f.) Membuat kompleks *query* menggunakan *Query-by-example*. (QBE)
- g.) Pencarian global dalam basis data.
- h.) Transformasi data disimpan ke dalam format yang menggunakan satu set fungsi yang telah ditetapkan, seperti menampilkan data *blob-data* atau *download-link*. dan banyak lagi.

2. Fungsi-Fungsi Pada *Phpmyadmin*

- a.) Database berfungsi untuk membuat database
- b.) Browser untuk menampilkan data
- c.) Structure untuk melihat struktur table .
- d.) koneksi.php digunakan untuk koneksi PHP ke database
- e.) form.php berisi form yang digunakan untuk menginput data.
- f.) proses.php digunakan untuk proses menyimpan data ke database
- g.) tabel.php digunakan untuk menampilkan data dalam bentuk tabel
- h.) delete.php digunakan untuk menghapus data didatabase
- i.) update.php digunakan untuk mengubah data yang diinginkan
- j.) simpan.php digunakan untuk menyimpan kembali data hasil perubahan .

- k.) Status berfungsi untuk melihat detail informasi yang berkaitan dengan MySQL server.
- l.) Set Karakter berfungsi untuk melihat karakter (*charset*) yang didukung oleh server MySQL.
- m.) Mesin berfungsi untuk melihat semua mesin (*engines*) yang didukung oleh server *MySQL*. *Engine* yang biasanya digunakan secara umum adalah *MyISAM*. *Storage Engine* lainnya yang sering digunakan adalah InnoDB.
- n.) Pengaturan berfungsi untuk mengatur sistem dari Phpmyadmin itu sendiri.
- o.) Basis Data berfungsi untuk membuat databade baru
- p.) Tabel adalah tempat untuk menyimpan data dalam sistem *database relational*.
- q.) *Mylsam* digunakan untuk mengatur tabel *non-transactional*. *MylSAM* menyediakan sistem untuk penyimpanan dan pengambilan data dengan kecepatan tinggi. Selain itu ada fasilitas pencarian *full text*. *MylSAM* didukung di semua konfigurasi *MySQL* dan merupakan *storage engine* standar kecuali jika diubah konfigurasinya.
- r.) *Memory* menyediakan tabel di *memory*.
- s.) Merge memungkinkan pengumpulan tabel *MyISAM* yang identik untuk ditangani sebagai tabel tunggal. Seperti *MyISAM*, *Memory* dan *Merge* bisa menangani tabel non-transactional, dan keduanya juga ada secara *default* di *MySQL Storage engine Memory* dulu dikenal sebagai *Heap Innodb Dan Bdb menyediakan tabel yang transaction-safe*. InnoDB juga dimasukkan di paket distribusi *binary MySQL 5.0* standar. Jika menggunakan distribusi *source*, Anda bisa meng-aktifkan atau mematikan *storage engine* ini sesuai keinginan Anda.
- t.) *Example* merupakan *storage engine dummy* yang tidak melakukan apapun. Anda bisa membuat tabel menggunakan *engine* ini namun tidak ada data yang bisa disimpan ke dalamnya. Anda tidak bisa mengambil data/*retrieve* dari tabel. Tujuan storage engine ini adalah

sebagai contoh di *source code MySQL* untuk menunjukkan cara menulis *storage engine* baru. Sehingga tujuan utamanya adalah untuk pengembang aplikasi MySQL sendiri dan bukan untuk pengguna.

- u.) *Ndbcluster* (ATAU NDB) merupakan *storage engine* yang digunakan oleh oleh *MySQL cluster* untuk mengimplementasi-kan tabel yang dipartisi di beberapa komputer. NDB ada di distribusi *biner MySQL 5.0*. *Storage engine* sekarang didukung oleh beberapa platform *Unix* dan untuk *platform* lain, seperti *Windows* sedang dicoba untuk dikembangkan. NDB tidak didukung pada konfigurasi *MySQL 5.1* standar. *ARCHIVE* digunakan untuk menyimpan data berjumlah banyak tanpa indeks. *CSV* digunakan untuk menyimpan data dalam format file teks menggunakan format *comma-separated values/CSV*.
- v.) *Blackhole* bisa menerima data tapi tidak menyimpannya. Sehingga ketika mengambil data, hasilnya selalu kosong.
- w.) *SQL (Structured Query Language)* merupakan bahasa untuk *database* yang didesain agar bisa mengambil data dan melakukan manajemen data di *database* relasional. *SQL* juga bisa digunakan untuk membuat skema *database*, memodifikasinya, dan manajemen kontrol dari pengaksesan objek.
- x.) *Search* adalah sebuah fitur *PHPMyAdmin* yang mempermudah pencarian baik dalam tabel atau dalam *database*. Ketika Anda membuka tabel, fitur *Search* mencari data yang ada di tabel Sementara jika membuka *database*, fitur *Search* mencari data yang ada di semua tabel yang ada di *database*.
- y.) *Export Database* yang baik mengijinkan adanya portabilitas dalam pemindahan artikel. Maksud portabilitas adalah user bisa mengekspor *sql statement* yang mendefinisikan struktur tabelnya sekarang, sehingga nantinya hasil ekspor bisa digunakan (diimpor) di *database* lainnya, agar tercipta *database* atau tabel di tempat lain yang strukturnya sama persis.

- z.) Impor adalah kebalikan dari ekspor. Dengan impor, Anda bisa membuat database (baik struktur maupun datanya) dengan jalan mengimpor dari file yang diperoleh dari proses ekspor.
- aa.) *Operations*. Jika *Operations* diklik, tampilannya seperti berikut. Di sini Anda dapat melakukan berbagai operasi, seperti *Alter table* order untuk mengubah urutan tabel, *Move table to* untuk memindahkan tabel ke *database* lain, *Table options* untuk mengganti opsi tabel, *Copy table to* untuk menyalin tabel ke *database* lain dan sebagainya.
- bb.) *Empty* yang fungsinya untuk menghapus Isi tabel atau dalam *sql statement* menggunakan perintah *Truncate*. Caranya klik tab *Empty*. Ketika ada pertanyaan *Do you really want to: TRUNCATE TABLE nama_tabel*, klik saja OK.
- cc.) *Drop* yang fungsinya untuk menghapus tabel. Caranya adalah dengan mengklik tab *Drop* dan mengklik OK, ketika muncul pesan *Do you really want to DROP TABLE nama_tabel*.

2.3.6 Unified Modeling Language (UML)


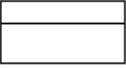

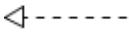


UML (*Unified Modelling Language*) merupakan standar bahasa yang banyak digunakan untuk mendefinisikan kebutuhan, membuat analisis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahudin, 2013). UML yang digunakan dalam perancangan perangkat lunak ini meliputi *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. *Class Diagram*

Class Diagram adalah diagram yang bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. Tabel simbol *class diagram* disajikan pada Tabel 2.2

Tabel 2.2 Tabel Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
3		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2. *Package Diagram*


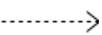







Package Diagram adalah diagram yang bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

3. *Usecase Diagram*

Usecase Diagram adalah diagram yang bersifat statis. Diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan

memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. Tabel simbol *use case diagram* disajikan pada Tabel 2.3

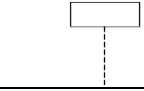
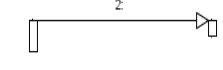
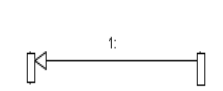
Tabel 2.3 Tabel Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
7		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
8		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
9		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

4. *Sequence Diagram*

Sequence diagram adalah Diagram yang bersifat dinamis dan interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Tabel simbol *sequence diagram* disajikan pada Tabel 2.4

Tabel 2.4. Tabel Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi




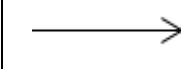
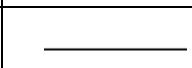
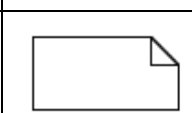
5. *Communication Diagram*

Communication Diagram adalah diagram yang Bersifat dinamis, selain itu juga sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

6. *Statechart Diagram*

Statechart Diagram adalah diagram yang bersifat dinamis, yang memperlihatkan keadaan-keadaan pada sistem, memuat status, transisi, serta aktivitas. Tabel simbol *statechart diagram* disajikan pada Tabel 2.5

Tabel 2.5. Tabel Simbol *StateChart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

7. *Activity Diagram*

Activity Diagram adalah diagram yang bersifat dinamis. Selain itu *Activity Diagram* merupakan tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek. Tabel simbol *activity diagram* disajikan pada Tabel 2.6

Tabel 2.6. Tabel Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

8. *Component Diagram*

Component diagram adalah diagram yang bersifat statis, diagram ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

9. *Deployment Diagram*

Deployment Diagram adalah diagram yang bersifat statis, diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*), memuat simpul-simpul beserta komponen-komponen yang di dalamnya.

Kesembilan diagram tersebut tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya *data flow diagram*, *entity relationship diagram*, dan sebagainya.

2.4 Teori Pengujian Sistem

Pengujian sistem merupakan proses mengeksekusi sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan

Dalam pembuatan perangkat lunak tentunya hal yang sering dihadapi adalah "*bug*" atau kesalahan "*error*" pada proses-proses tertentu. Untuk menghindari banyaknya *bug* maka diperlukan pengujian perangkat lunak, sebelum perangkat lunak yang telah kita buat diberikan ke pelanggan atau selama perangkat lunak masih terus dikembangkan.

2.4.1 *Black Box testing*

Black box testing adalah 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 Shalahuddin dan Rosa (2011).

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input dan melakukan pengetesan pada fungsional program Mustaqbal, dkk (2015)

Pengujian *blackbox* merupakan tahap yang berfokus pada pernyataan fungsional perangkat lunak. *Test case* ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Apakah pemasukan data telah berjalan sebagaimana mestinya dan apakah informasi yang tersimpan dapat dijaga kemutahirannya Luqman (2012).

Dengan demikian, pengujian *blackbox* memungkinkan pembuat perangkat lunak menemukan kesalahan dalam beberapa hal yaitu :

1. Fungsi-fungsi yang tidak benar atau salah
2. Kesalahan *interface*, kinerja, inialisasi, dan kesalahan terminasi
3. Kesalahan dalam struktur data atau akses *database* eksternal.