

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem informasi mempunyai peranan penting dalam membantu menyediakan informasi untuk berbagai tingkatan manajemen. Sistem yang dibutuhkan adalah sistem yang memperlancar proses kegiatan yang sedang berjalan.

2.1.1. Pengertian Sistem

Menurut Murdick (2013:3) “Sistem adalah seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan bersama”.

Sedangkan menurut Pratama (2014:7) “Sistem adalah sebagai sekumpulan prosedur yang saling berkaitan dan saling berhubungan untuk melakukan tugas bersama-sama”.

Dari kutipan diatas dapat disimpulkan sistem adalah serangkaian komponen, unsur maupun elemen yang terorganisir yang saling membutuhkan antara satu dengan yang lainnya.

2.1.2. Pengertian Sistem Informasi

Menurut Elisabet & Rita (2017:2) “Sistem informasi merupakan suatu kombinasi teratur dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi”.

Sedangkan menurut Mahatmyo (2014:6) “Sistem informasi (*information system*) adalah serangkaian prosedur formal di mana data dikumpulkan, diproses menjadi informasi dan didistribusikan ke pengguna”.

Sistem informasi adalah kumpulan atau susunan yang terdiri dari perangkat keras dan perangkat lunak serta tenaga pelaksanaannya yang bekerja dalam sebuah proses berurutan dan secara bersama-sama saling mendukung untuk menghasilkan suatu produk (Sholikhah, Sairan, Syamsiah, 2017).

Dari pengertian diatas dapat disimpulkan sistem informasi adalah sekumpulan prosedur organisasi yang dilaksanakan untuk mencapai suatu tujuan yaitu memberikan informasi bagi pengambil keputusan dan untuk mengendalikan organisasi.

2.1.3. Pengertian Sistem Informasi Akuntansi

Menurut Mahatmyo (2014:9) “Sistem informasi akuntansi merupakan Sekelompok struktur dalam sebuah entitas yang mengelola sumber daya fisik dan sumber daya lain untuk mengubah data ekonomi menjadi informasi akuntansi, agar dapat memenuhi kebutuhan informasi berbagai pihak”.

Sedangkan menurut Hery (2015:9) “Sistem Informasi Akuntansi adalah merancang sistem pemrosesan data akuntansi. Data transaksi (*input*) diproses sedemikian rupa secara sistem menghasilkan sebuah informasi (*output*) yang berguna dalam proses pengambilan keputusan”.

Dari pengertian diatas dapat disimpulkan bahwa sistem informasi akuntansi merupakan sistem informasi berbasis komputerisasi yang mengola data keuangan yang berhubungan dengan data transaksi dalam siklus akuntansi dan menyajikannya dalam bentuk laporan keuangan kepada manajemen perusahaan.

2.1.4. Penerimaan Kas

Menurut Mulyadi (2016:379) “Penerimaan kas adalah kas yang diterima perusahaan baik yang berupa uang tunai maupun surat – surat berharga yang mempunyai sifat dapat segera digunakan, berasal dari transaksi perusahaan maupun penjualan tunai, pelunasan piutang, atau transaksi lainnya yang dapat menambah kas perusahaan”.

Menurut Surjaweni (2015:121) “Penerimaan kas adalah suatu prosedur catatan yang dibuat untuk melaksanakan kegiatan penerimaan uang yang berasal dari berbagai macam sumber yaitu dari penjualan tunai, penjualan aktiva tetap, pinjaman bank dan setoran barang baru”.

Berdasarkan kesimpulan di atas dapat disimpulkan bahwa penerimaan kas merupakan prosedur yang menangani suatu peristiwa atau kejadian yang mengakibatkan terjadinya penambahan uang dalam kas yang berasal dari penjualan tunai maupun piutang yang melibatkan bagian-bagian yang saling berkaitan satu sama lain.

2.1.5. Pengeluaran Kas

Menurut Sujarweni (2015:123) menyatakan bahwa “Pengeluaran kas merupakan system yang membahas keluarnya uang yang digunakan untuk pembelian tunai maupun kredit dan pembayaran”.

Menurut Standar Akuntansi Keuangan (2012) “Pengeluaran kas adalah alat pembayaran yang siap dan bebas dipergunakan secara bebas untuk membiayai kegiatan perusahaan” (PSAK No. 2 Tahun 2012).

Dari pengertian diatas dapat disimpulkan bahwa pengeluaran kas merupakan suatu hal yang melibatkan catatan-catatan atas transaksi-transaksi

yang mengakibatkan berkurangnya saldo-saldo kas tunai baik yang berasal dari pembelian tunai, pembayaran utang, pengeluaran transfer maupun pengeluaran pengeluaran lainnya.

2.1.6. Jurnal

Menurut Bahri (2016:26) “Jurnal adalah pencatatan yang sistematis dan kronologis atas transaksi keuangan yang terjadi pada suatu perusahaan. Jurnal merupakan langkah awal dalam siklus akuntansi. Jurnal berbentuk kolom-kolom yang berisi tentang tanggal (*date*), nama rekening/keterangan (*account name*), referensi (*ref*), debet dan kredit”.

a. Jurnal Pengeluaran Kas (*Cash Payments Journal*)

Menurut Bahri (2016:340) “Jurnal pengeluaran kas berfungsi untuk mencatat seluruh transaksi pengeluaran kas”.

Pengeluaran	Rp. XXXXXXXX (Debet)
Kas	Rp. XXXXXXXX (Kredit)

b. Jurnal Penerimaan Kas (*Cash Receipt Journal*)

Menurut Bahri (2016:340) “Jurnal penerimaan kas berfungsi untuk mencatat seluruh transaksi penerimaan kas”.

Kas	Rp. XXXXXXXX (Debet)
Penerimaan	Rp. XXXXXXXX (Kredit)

Jurnal merupakan catatan akuntansi yang pertama digunakan untuk mencatat, mengklasifikasi dan meringkas data keuangan dan data yang lainnya (Kabuhung, 2013).

Dari pengertian diatas dapat disimpulkan bahwa jurnal merupakan catatan akuntansi yang digunakan untuk mencatat laporan keuangan perusahaan yang

menjadi dasar bagi penentuan ke akun mana suatu transaksi dicatat, berapa jumlah uang yang dicatat, disisi mana dicatat, dan keterangan singkat tentang transaksi.

2.2. Peralatan Pendukung

Dalam pembuatan tugas akhir ini penulis menggunakan beberapa peralatan pendukung membuat penulis terbantu dalam pengerjaan tugas akhir adalah sebagai berikut :

2.2.1. *Unified Modelling Language (UML)*

Menurut Mulyani (2016:35) “*Unified Modeling Language* selanjutnya disebut UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

Menurut Shofwan dan Dian (2018:166) “UML adalah bahasa yang dapat digunakan untuk spesifikasi, visualisasi, dan dokumentasi sistem *objek-oriented software* pada fase pengembangan”.

Dari penjelasan diatas dapat disimpulkan bahwa UML adalah bahasa yang digunakan untuk pemodelan dari sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek.

2.2.2. *Use Case Diagram*

Menurut Sukanto dan M. Salahudin (2014:155) “*Use case* atau diagram *case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

Menurut Murad (2013:57) “Diagram *Use Case* adalah diagram yang bersifat status yang memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas)”.

Use case menggambarkan fungsi-fungsi sistem dari sudut pandang pengguna eksternal dan dalam sebuah cara yang mudah dipahami. *Use case* merupakan penyusunan kembali lingkup fungsional sistem yang disederhanakan lagi. *Use case diagram* adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal pengguna. *Use case* diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use case* diagram dapat digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan jelas bagaimana proses dari sistem tersebut, bagaimana cara aktor menggunakan sistem, serta apa saja yang dapat dilakukan pada suatu system (Nugraha, Syarif, Dharmawan, 2018).

Menurut pengertian di atas dapat disimpulkan bahwa *use case diagram* merupakan pemodelan untuk kelakuan sistem informasi yang tindakannya saling terkait baik terotomatisasi maupun secara manual untuk tujuan melengkapi satu tugas bisnis tunggal.

2.2.3. Activity Diagram

Menurut Sukanto dan Salahuddin (2013:161) ”*Activity* diagram menggambarkan *work flow* (aliran kerja) atau aktifitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak”.

Menurut Murad (2013:57) “*Activity diagram* merupakan diagram yang bersifat dinamis. *Activity diagram* adalah tipe khusus dari diagram *state* yang

memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem dan berfungsi untuk menganalisa proses”.

Menurut para ahli diatas dapat disimpulkan bahwa *activity diagram* merupakan diagram yang menggambarkan tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.

2.2.4. Sequence Diagram

Menurut Vidia (2013:20) “*Sequence diagram* dibuat berdasarkan *activity diagram* dan *class diagram*. *Sequence diagram* menggambarkan aliran pesan yang terjadi antar kelas yang dideskripsikan pada *class diagram* dengan menggunakan operasi yang dimiliki kelas tersebut. Untuk aliran pesan, *sequence diagram* merujuk pada alur sistem *activity diagram* yang telah dibuat sebelumnya”.

Menurut Sukamto dan M.Salahudin (2014:165) “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek”.

Menurut beberapa ahli diatas dapat disimpulkan bahwa *sequence diagram* adalah kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan interaksi antar objek.

2.2.5. Deployment Diagram

Menurut Sukamto dan M. Shalahuddin (2013:154) “*Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”.

Menurut Yurindra (2017:32) “*Deployment diagram* merupakan gambaran proses-proses berbeda pada suatu sistem yang berjalan dan bagaimana relasi didalamnya. Hal ini yang mempermudah *user* dalam pemakaian sistem yang telah dibuat dan diagram tersebut merupakan diagram yang statis”.

Dari pengertian ahli diatas dapat disimpulkan bahwa *deployment diagram* merupakan gambaran proses berbeda pada suatu sistem yang berjalan dan bagaimana relasi didalamnya.

2.2.6. Entity Relationship Diagram

Menurut Sukamto dan Shalahuddin (2015:50) “ERD (*Entity Relationship Diagram*) adalah teori himpunan dalam bidang sistematika, ERD digunakan untuk permodelan basis data rasional”.

Sedangkan menurut Pratama (2014:49) “ERD adalah diagram yang menggambarkan keterkaitan antar tabel beserta dengan *field-field* didalamnya pada suatu *database* sistem”.

Dari pengertian diatas bisa disimpulkan bahwa ERD adalah sebuah hubungan antar *entity* atau obyek-obyek dasar yang divisualkan dengan diagram.

2.2.7. Logical Record Structure

Menurut Wulandari (2013:17) “*Logical Record Structure* dibentuk dengan nomor dari *tipe record*. Beberapa *tipe record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik”.

Menurut Lestari (2013:12) “*Logical Record Structure* dibentuk dengan nomor *tipe record*, beberapa *tipe record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik”.

Dari pengertian menurut para ahli diatas dapat disimpulkan bahwa *logical record structure* merupakan transpormasi dari ERD ke LRS yang melalui proses kardinalitas dan menghasilkan atribut-atribut yang saling berelasi.

2.2.8. User Interface

Menurut Sukamto dan M. Shalahuddin (2013:109) “Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain. Metode pada antar muka yang diimplementasikan pada suatu kelas harus sama persis dengan yang ada pada antarmuka. Antarmuka atau *interface* biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas tapi mengakses antarmukanya”.

Antarmuka pengguna merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi (Purnia, Dini Silvi, 2014).

Maka dapat disimpulkan bahwa *user interface* merupakan serangkaian tampilan grafis yang dapat dimengerti oleh pengguna komputer dan diprogram sedemikian rupa sehingga dapat terbaca oleh sistem operasi komputer dan beroperasi sebagaimana mestinya.

2.2.9. Blackbox Testing

Menurut Sukamto dan Shalahuddin (2013:275) “*Black Box Testing* 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 *black-box* juga merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkapkan kelas kesalahan daripada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi (Wijayanto, Ridho, 2014).

Menurut pengertian diatas dapat disimpulkan bahwa *blacbox testing* ialah pengujian yang memfokuskan kepada perilaku perangkat lunak berdasarkan kebutuhan fungsional dari perangkat lunak yang bersangkutan.

2.2.10. Basis Data (*Database*)

Menurut Sukanto dan Shalahuddin (2014:43) mengemukakan bahwa “basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.

Menurut Pratama (2014:18) mengemukakan bahwa “*Database* adalah sistem informasi menjadikan data dan informasi terkumpul secara terpusat pada satu tempat”.

Dari kutipan diatas dapat disimpulkan basis data (*database*) adalah suatu sistem pendukung dalam pembuatan aplikasi untuk menyimpan data yang di proses oleh *user* saat menjalankan suatu aplikasi.

2.2.11. Xampp

Menurut Dadan dan Kerendi (2015:28) “Xampp adalah salah satu aplikasi *web server* apache yang terintegrasi dengan MySQL dan phpmyadmin”.

Menurut Aryanto (2016:4) “Xampp merupakan sebuah aplikasi perangkat lunak pemrograman dan *database* yang didalamnya terdapat berbagai macam

aplikasi pemrograman seperti: Apache HTTP Server, MySQL *database*, bahasa pemrograman PHP dan perl”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa Xampp merupakan *software* atau perangkat lunak yang berfungsi untuk menjalan website berbasis *PHP* dan menggunakan pengolahan data *MySQL* dikomputer *local* yang menghubungkan antara *Apache*, *HTTP Server*, *MySQL database*, bahasa pemrograman *PHP* dan *Perl*.

2.2.12. Crystal Reports 8.5

Menurut Aminudin (2016:158) “*Crystal report* adalah suatu *form* khusus berbentuk seperti lembaran format naskah yang ingin dicetak”.

Menurut Atmoko (2013:3) “*Crystal Report* merupakan komponen yang akan kita gunakan untuk membuat *report* atau laporan dari program yang akan kita buat, agar dapat dipahami oleh pengguna, yang report tersebut diambil dari kumpulan data dari *table* yang tersimpan didalam *database SQL Server*”.

Dari penjelasan diatas dapat disimpulkan bahwa *crystal report* adalah *software* yang dapat digunakan dalam pembuatan aplikasi guna untuk mencetak laporan yang tersimpan pada basis data (*database*) yang telah di proses pada saat menjalankan suatu sistem aplikasi oleh pengguna yang bersangkutan.

2.2.13. Microsoft Visual Basic

Menurut Anhar (2016:1) “Visual Basic merupakan bahasa pemrograman yang menawarkan *Integrated Development Environment (IDE)* visual untuk membuat program perangkat lunak berbasis GUI (*Graphical User Interface*) pada sistem operasi Microsoft Windows”.

Menurut Raharjo (2016:1) mengemukakan bahwa “visual basic adalah *software* yang mendukung proses pengembangan dan eksekusi program dilingkungan windows”.

Dari pengertian para ahli diatas dapat disimpulkan bahwa *Microsoft visual basic* merupakan salah satu bahasa pemrograman yang digunakan dalam melakukan pembuatan aplikasi.

2.2.14. MySQL

Menurut Nugroho (2013:26) “*MySQL* adalah *software* atau program *database server*. Sedangkan *SQL* adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database server* termasuk dalam *MySQL* itu sendiri. *SQL* juga dipakai dalam *software database server* lain, seperti *SQL Server*, *Oracle*, *PostgreSQL* dan lainnya”.

Menurut Buana (2014:2) “*MySQL* merupakan *database server* yang paling sering digunakan dalam pemrograman PHP. *MySQL* digunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah dan menghapus data yang berada dalam *database*”.

MySQL menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database* yang terbagi menjadi tiga bagian, yaitu DDL, DML, dan DCL.

1. DDL (*Data Definition Language*)

Data Definition Language (DDL) adalah suatu tata Bahasa definisi data pada *MySQL*, DDL digunakan untuk mendefinisikan suatu *database*, *table*, *table-space*, *log file group*, *server*, dan *index*. DDL umumnya digunakan

mendefinisikan suatu wadah data atau *record*. DDL terdiri dari *create*, *alter*, *drop*, *rename* (Sibero, 2013:98).

berikut penjelasan mengenai masing-masingnya:

a. Create

Create digunakan untuk mendefinisikan suatu struktur, wadah yang digunakan sebagai media suatu data/*record* dan atribut pendukung lainnya.

b. Alter

Alter digunakan untuk mengubah definisi suatu struktur, wadah yang digunakan sebagai media suatu data/*record* dan atribut pendukung lainnya.

c. Drop

Drop digunakan untuk menghapus suatu struktur, wadah yang digunakan sebagai media suatu data/*record* dan atribut pendukung lainnya.

d. Rename

Rename digunakan untuk mengganti nama suatu struktur, wadah yang digunakan sebagai media suatu data/*record*. Penggunaan *rename* hanya berlaku untuk *database* dan *table*.

2. DML (*Data Manipulation Language*)

Data Manipulation Language adalah suatu tata bahasa memanipulasi data pada *MySQL*, DML digunakan untuk memanipulasi data pada *table database* (Sibero,2013:104).

Berikut komponen DML:

a. Select

Select berfungsi untuk menyeleksi, memilih atau menampilkan data-data yang ada dalam tabel, baik menampilkan semua kolom maupun sebagian kolom berdasarkan kondisi.

b. Insert

Insert berfungsi untuk memasukkan atau menyimpan data dari luar kesistem dalam *database*. Ada beberapa cara dalam memasukkan data yaitu dengan menyamakan kolom dan data, menyebutkan kolom, tanpa menyebutkan kolom, memasukkan hanya sebagian pada kolom.

c. Update

Update berfungsi untuk memperbaharui data lama menjadi data yang baru.

d. Delete

Delete berfungsi untuk menghapus atau menghilangkan baris data (*record*) dari *table*.

3. *DCL (Data Control Language)*

Data Control Language adalah bagian inti dari *SQL* yang mempunyai kemampuan untuk mengatur hak akses terhadap sebuah basis data (*database*). *DCL* terbagi menjadi 2:

a. GRANT (Memberikan Hak Akses)

GRANT merupakan perintah untuk memberikan hak izin akses bagi *user* di *MySQL* agar dapat mengakses *database*, *table* dan kolom.

b. REVOKE (Mencabut Hak Akses)

REVOKE merupakan kebalikan dari perintah *GRANT* yaitu menghapus atau mencabut kembali izin akses *user MySQL* yang sebelumnya telah diberikan.

Didalam *SQL*, *Key* terbagi menjadi beberapa jenis diantaranya adalah sebagai berikut:

1. *Primary Key*

Kunci utama adalah atribut merupakan kunci calon yang telah dipilih untuk mengidentifikasi setiap *record* secara unik. Kunci utama harus merupakan atribut yang benar-benar unik dan tidak boleh ada nilai *NULL*. Kunci utama adalah suatu nilai dalam basis data yang digunakan untuk mengidentifikasi suatu baris dalam *table*.

2. *Foreign Key*

Jika sebuah kunci utama terhubungan ke tabel lain, maka keberadaan kunci utama tersebut di sebut sebagai kunci tamu. Kunci tamu adalah Sebuah kumpulan atribut dalam satu relasi yang digunakan untuk menunjuk ke suatu baris pada relasi yang lain (harus berkorespondensi dengan kunci utama pada relasi yang kedua), seperti *logical pointer*.

3. *Candidate Key*

Kunci calon adalah salah satu rangkaian yang mempunyai nilai unik untuk membedakan atau mengidentifikasi nilai-nilai kombinasi yang unik diantara semua kejadian yang spesifik dari entetitas. Kunci calon ini tidak boleh berisi atribut dari tabel yang lain. kombinasi dari atribut yang dapat digunakan untuk mengidentifikasi secara unik rekor *database* tanpa data apapun yang asing.

4. *Secondary Key*

Kunci sekunder adalah sebuah atribut atau kombinasi yang digunakan hanya untuk tujuan pengambilan data.

5. *Alternate Key*

Kunci Alternatif adalah kunci alternatif yang tidak terpilih. Misal : dalam suatu entitas terdapat dua atribut yang bisa dijadikan sebagai kunci. Sementara yang boleh dijadikan kunci hanya satu, maka anda harus memilih salah satu. Atribut yang dipilih disebut kunci utama. sedangkan atribut yang tidak dipilih disebut dengan kunci.

6. *Composite Key*

Dalam desain basisdata, kunci komposit adalah kunci yang terdiri dari 2 atau lebih atribut yang secara unik mengidentifikasi suatu kejadian entitas. Setiap atribut yang membentuk kunci senyawa adalah kunci sederhana dalam haknya sendiri.

Secara garis besar, *database MySQL* mempunyai 3 macam tipe data, yaitu:

1. Tipe Data *Numeric*

Tipe Data *Numeric* pada *database MySQL* terbagi atas beberapa macam tipe data, yaitu:

a. *INT*

INT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan -2.147.483.648 s/d 2.147.483.647. Tipe data ini mempunyai ukuran 4 byte (32 bit).

b. *TINYINT*

TINYINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -128 s/d 127. Tipe data ini mempunyai ukuran 1 byte (8 bit).

c. *SMALLINT*

SMALLINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -32.768 s/d 32.767. Tipe data ini mempunyai ukuran 2 byte (16 bit).

d. *MEDIUMINT*

MEDIUMINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -8.388.608 s/d 8.388.607. Tipe data ini mempunyai ukuran 3 byte (24 bit).

e. *BIGINT*

BIGINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -8.388.608 s/d 8.388.607. Tipe data ini mempunyai ukuran 8 byte (64 bit).

f. *FLOAT*

FLOAT digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif persisi tunggal. Tipe data ini mempunyai ukuran 4 byte (32 bit).

g. *DOUBLE*

DOUBLE digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif persisi ganda. Tipe data ini mempunyai ukuran 8 byte (64 bit).

h. *DECIMAL*

DECIMAL digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif persisi ganda. Tipe data ini mempunyai ukuran 8 byte (64 bit).

i. REAL

REAL digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif. Tipe data ini mempunyai ukuran 8 byte (64 bit).

j. NUMERIC

NUMERIC digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif. Tipe data ini mempunyai ukuran 8 byte (64 bit).

2. Tipe Data *Date & Time*

Tipe Data *Date & Time* pada *database MySQL* terbagi atas beberapa macam tipe data, yaitu:

a. DATE

DATE digunakan untuk menyimpan data tanggal dalam format YY:MM:DD.

b. DATETIME

DATETIME digunakan untuk menyimpan data tanggal dan waktu dalam format YY:MM:DD HH:MM:SS.

c. TIME

TIME digunakan untuk menyimpan data waktu dalam format HH:MM:SS.

d. YEAR

YEAR digunakan untuk menyimpan data tahun.

3. Tipe Data *String*

Tipe Data *String* pada *database MySQL* terbagi atas beberapa macam tipe data, yaitu:

a. CHAR

CHAR digunakan untuk menyimpan data karakter/*string* dengan ukuran tetap. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 karakter.

b. VARCHAR

VARCHAR digunakan untuk menyimpan data karakter/*string* dengan ukuran dinamis. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 karakter untuk *MySQL* Versi 4.1 dan mempunyai jangkauan antara 0 s/d 65.535 untuk *MySQL* versi 5.0.3

c. BLOB

BLOB adalah tipe data yang digunakan untuk menyimpan data biner seperti *files, image, suara, dll.* Tipe data ini mempunyai 216-1 byte.

d. TINYBLOB

TINYBLOB digunakan untuk menyimpan data biner seperti *file, image, dan suara.* Tipe data ini mempunyai 255 byte.

e. MEDIUMBLOB

MEDIUMBLOB digunakan untuk menyimpan data biner seperti *file, image, dan suara.* Tipe data ini mempunyai 244-1 byte.

f. LONGBLOB

LONGBLOB digunakan untuk menyimpan data biner seperti *file, image, dan suara.* Tipe data ini mempunyai 232-1 byte.

g. TEXT

TEXT digunakan untuk menyimpan data *text.* Tipe data ini mempunyai jangkauan antara 0 sampai dengan 65.535 (216-1) karakter.

h. TINYTEXT

TINYTEXT digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 untuk *MySQL* Versi 4.0 dan mempunyai jangkauan antara 0 s/d 65.535 untuk *MySQL* versi 5.0.3.

i. MEDIUMTEXT

MEDIUMTEXT digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 224-1 karakter.

j. LONGTEXT

LONGTEXT digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 232-1 karakter.

k. ENUM

ENUM digunakan untuk menyimpan data *enumerasi* (kumpulan data).

1. *SET*

SET digunakan untuk menyimpan data himpunan data.

Dari pengertian diatas dapat disimpulkan bahwa *MySQL* merupakan aplikasi *database* yang bertujuan memudahkan penyimpanan dan pengaksesan data dan dapat diakses dengan cara yang mudah dan cepat.

2.2.15. *PhpMyAdmin*

Menurut Nugroho (2013:71) “*phpMyAdmin* adalah *tools* yang dapat digunakan dengan mudah untuk manajemen *database MySQL* secara *visual* dan *Server MySQL*, sehingga kita tidak perlu lagi harus menulis *query SQL* setiap akan melakukan perintah operasi *database*. *Tool* ini cukup populer, anda dapat mendapatkan fasilitas ini ketika menginstal paket *triad phpMyAdmin*, karena termasuk dalam *xampp* yang sudah di install”.

Menurut Buana (2014:2) “*phpMyAdmin* adalah salah satu aplikasi yang digunakan untuk memudahkan dalam melakukan pengelolaan *database MySQL*. *phpMyAdmin* merupakan aplikasi web yang bersifat *opensource*”.

Dari pengertian diatas dapat disimpulkan bahwa *phpMyAdmin* merupakan sebuah *software* berbasis pemrogram *PHP* yang dipergunakan untuk administrasi *database MySQL*.

2.2.16. ODBC

ODBC merupakan sebuah set *Application Programming Interface* (API) untuk mengakses *database* melalui *driver* yang disediakan *provider database* di lingkungan sistem operasi *windows* (Yulansari, Sukadi, 2013).

Open Database Connectivity (ODBC) merupakan suatu kerangka kerja umum untuk mengakses dan mengubah isi *database* (Sabanise, Afriliana, 2017).

Dapat disimpulkan bahwa ODBC merupakan suatu aplikasi yang memudahkan *user* dalam pengolahan data serta menghubungkan pengolahan data antar *database* sehingga memudahkan *user* untuk mengakses data.