

BAB II

LANDASAN TEORI

2.1 Sistem

Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan. Organisasi terdiri dari sejumlah sumber daya manusia, material, mesin, uang dan informasi. Sumber data tersebut juga bekerja sama menuju tercapainya suatu tujuan tertentu yang ditentukan oleh pemilik atau manajemen. Sistem terdapat dua kelompok pendekatan sistem dalam mendefinisikan sistem yaitu pendekatan pada prosedur dan pendekatan pada komponen-komponen atau elemen-elemen (Yakub, 2012).

2.2 Informasi

Informasi adalah data yang olah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Informasi juga disebut juga data yang diproses atau data yang memiliki arti. Informasi adalah data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakannya (Yakub, 2012).

2.3 Sistem Informasi

Sistem informasi merupakan kombinasi teratur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi.

Sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sedangkan menurut penulis sistem informasi merupakan sekumpulan komponen yang saling berhubungan memproses, menyimpan dan mendistribusikan informasi guna untuk menunjang pengambilan keputusan sebuah organisasi (Yakub, 2012).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.4 Jadwal

Menurut kamus besar bahasa Indonesia edisi ke tiga tahun 2007 jadwal dapat dilihat pada Tabel 2.1 sebagai berikut:

Table 2.1 Tabel Pengertian Jadwal

jadwal/jad·wal/ n	pembagian waktu berdasarkan rencana pengaturan urutan kerja; daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci: -- <i>penataran itu sudah disetujui oleh pimpinan</i> ; -- acara jadwal kegiatan yang akan dilakukan;
menjadwalkan/men·jad·wal·kan/ v	menetapkan jadwal: <i>panitia sudah ~ penataran itu setahun sebelum diselenggarakan</i> ;
terjadwal/ter·jad·wal/ v	sudah dijadwalkan; sudah direncanakan; masuk dalam jadwal: <i>kegiatan itu sudah ~ dalam agenda saya</i> ;
penjadwalan/pen·jad·wal·an/ n	proses, cara, perbuatan menjadwalkan atau memasukkan dalam jadwal

(Sumber: Kamus Bahasa Indonesia Edisi Ke Tiga Tahun 2007)

2.5 World Wide Web Consortium

World Wide Web Consortium (W3C) adalah suatu konsorsium yang bekerja untuk mengembangkan standar-standar untuk *World Wide Web*. Spesifikasi teknologi-teknologi utama yang dipakai sebagai basis utama web, seperti URL (*Uniform Resource Locator*), HTTP (*Hypertext Transfer Protocol*), dan HTML (*HyperText Markup Language*) dikembangkan dan diatur oleh badan ini.

Misi dari W3C bertujuan untuk mendorong semua potensi penuh dari dunia web yang bisa dikembangkan dengan menyediakan protokol dan panduan untuk menjamin pertumbuhan jangka panjang dari web itu sendiri.

W3C dibuat pada 20 Oktober 1994 oleh Tim Berners-Lee, didirikan oleh Massachusetts Institute of Technology (MIT). W3C bekerja dengan komunitas

global untuk membuat standard internasional *client* dan *server* yang memungkinkan perdagangan dan komunikasi *online* melalui internet. W3C juga menghasilkan *software* acuan. W3C *Netscape Communications Corporation* adalah salah satu anggota pendiri. Konsorsium ini dijalankan oleh MIT LCS, INIRA *Institute national the Recherche en Informatique* sebuah lembaga penelitian ilmu komputer perancis, bekerja sama dengan CERN *Conseil Européen pour la Recherche Nucleaire*, tempat lahirnya Web.

W3C didanai oleh industri yang menjadi anggotanya, tetapi produknya tersedia gratis. Direktur W3C adalah tim Berners-Lee yang menemukan world wide web di CERN. W3C (*World Wide Web Consortium*), standar dari berbagai macam penyedia jasa untuk pembangunan dari teknologi yang berhubungan dengan Web, seperti HTML. W3C bekerja dengan tujuan umum membuat Web dapat diakses oleh semua user (lepas dari batasan budaya, pendidikan, keahlian, lokasi, keadaan lingkungan dan psikososial).

Karena Web begitu penting (meliputi segala aspek), tidak ada satu organisasinya pun yang dapat berdiri sendiri mengembangkan setiap bagian teknologi itu, W3C memberikan suatu wadah untuk bisa mengembangkan secara bersama. Perusahaan anggota utama dari organisasi itu adalah: *IBM, Microsoft, America Online, Apple, Adobe Macromedia, Sun Microsystems* selain anggota-anggota tersebut masih banyak lagi lembaga-lembaga baik pemerintahan maupun swasta yang turut andil dalam perkembangan W3C.

2.6 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOD).

1. Use Case Diagram

Use case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem itu digunakan, bukan bagaimana sistem dan elemen-elemennya saling berinteraksi (Martin Fowler, 2005).

Komponen notasi dasar yang dipunyai oleh *use-case diagram* adalah *actor*, *use-case*, dan *association*. Notasi yang terdapat pada *use-case diagram* dapat dilihat pada Table 2.2 di bawah.

Tabel 2.2 Simbol *Use Case Diagram*

Notasi	Keterangan	Simbol
<i>Actor</i>	<i>Actor</i> adalah pengguna sistem, <i>actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan output, maka aplikasi tersebut juga bias dianggap sebagai <i>actor</i>	 Actor
<i>Use Case</i>	<i>Use Case</i> digambarkan sebagai lingkaran <i>elips</i> dengan nama <i>Use Case</i> dituliskan didalam <i>elips</i> tersebut.	 UseCase
<i>Association</i>	<i>Association</i> adalah digambarkan dengan sebuah garis yang berfungsi menghubungkan <i>actor</i> dengan <i>use case</i> .	
<i>Generalisasi</i>	<i>Generalisasi</i> adalah untuk memperlihatkan struktur pewaris yang terjadi.	




(Sumber: Martin Fowler, 2005)

2. Activity Diagram

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja (Martin Fowler, 2005).

Simbol *activity diagram* dapat dilihat pada Table 2.3 di bawah ini.

Tabel 2.3 Simbol *Activity Diagram*

Notasi	Keterangan	Simbol
<i>Initial</i>	Titik awal untuk memulai suatu aktivitas	
<i>Final</i>	Titik akhir untuk mengakhiri suatu aktivitas	
<i>Activity</i>	Menandakan sebuah aktivitas	
<i>Decision</i>	Pilihan untuk pengambilan keputusan	
<i>Fork/Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.	
<i>Rake</i>	Menunjukkan adanya dekomposisi / untuk mengakhiri suatu aliran	
<i>Time</i>	Tanda waktu	
<i>Send</i>	Tanda pengiriman	
<i>Receive</i>	Tanda penerimaan	
<i>Flow Final</i>	Aliran akhir	

(Sumber: Munawar, 2005)

3. Class Diagram




Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

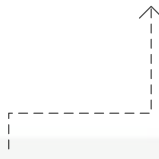


objek. *Class* menggambarkan keadaan (*atribut/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda atau fungsi*) (Martin Fowler, 2005). Simbol *class* diagram dapat dilihat pada Table 2.4 di bawah ini.

Tabel 2.4 Simbol *Class Diagram*

Notasi	Keterangan	Simbol
<i>Class</i>	<i>Class</i> adalah blok – blok pembangun pada pemrograman berorientasi objek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan <i>property/atribut class</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .	
<i>Assosiation</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).	
<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i>	

Hak Cipta Dilindungi Undang-Undang

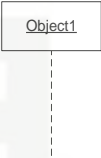



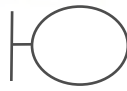
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

	terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
<i>Dependency</i>	Kadang kala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.	
<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “memunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid	
<i>Generalization</i>	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi objek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “ <i>parent</i> ”-nya/induknya.	


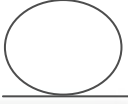
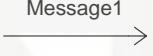

4. Sequence Diagram

Sequence diagram adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal. (Munawar (2005)). *Symbol sequence diagram* dapat dilihat pada Table 2.5 di bawah.

Tabel 2.5 Simbol Sequence Diagram

Notasi	Keterangan	Simbol
Object (Partisipan)	<i>Object</i> atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan object, maka <i>actor</i> juga dapat diurutkan sebagai kolom.	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang akan melakukan sebuah aksi.	
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.	
<i>Boundary</i>	<i>Boundary</i> terletak di antara sistem dengan dunia sekelilingnya. Semua <i>form</i> , laporan-laporan, antar muka ke perangkat keras seperti <i>printer</i> atau <i>scanner</i> dan antar muka ke sistem lainnya adalah termasuk dalam kategori.	

Tabel 2.5 Simbol *Sequence Diagram* (Lanjutan)

<i>Control</i>	<i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan.	
<i>Entity</i>	<i>Entity</i> digunakan menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data	
<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah <i>horizontal</i> antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> .	
<i>Self-message</i>	<i>Self-message</i> atau panggilan mandiri mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri	

(Sumber: Munawar, 2005)

2.7 Basis Data (*Database*)

Basis data adalah sekumpulan data yang diproses dengan bantuan computer yang memungkinkan data dapat diakses dengan mudah dan cepat, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berrelasi, Basisdata adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Basis data dapat diartikan kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. sedangkan menurut pendapat penulis basis data adalah kumpulan elemen-elemen yang saling terintegrasi yang diproses dengan sistem komputerisasi sehingga menghasilkan suatu informasi (Kustiyahningsih dan Anamisa, 2011).

Tujuan awal dan yang utama dalam pengelolaan data pada suatu basis data adalah agar kita dapat memperoleh/menemukan kembali data yang diperlukan

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dengan mudah dan cepat. Selengkapya pemanfaatan basis data dapat dilakukan untuk memenuhi sejumlah tujuan seperti berikut ini:

1. Kecepatan dan kemudahan

Lebih cepat dan mudah dalam menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut.

2. Efisiensi ruang penyimpanan

Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena bisa melakukan penekanan jumlah *redundancy* data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.

3. Keakuratan

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan atau batasan (*constraint*) tipe data, domain data, keunikan data dan sebagainya yang secara ketat sangat berguna untuk menekan ketidak akuratan pemasukan atau penyimpanan data.

4. Ketersediaan

Karena kepentingan pemakaian data, sebuah basis data dapat memiliki data yang disebar dibanyak lokasi geografis.

5. Kelengkapan

Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang maka tidak bisa hanya dengan menambah *record-record* data saja, tetapi juga melakukan perubahan struktur basis data, baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan *field-field* baru pada suatu tabel.

6. Keamanan

Ada sejumlah sistem (*aplikasi*) pengelola basis data yang tidak menerapkan aspek keamanan dalam penggunaan basis data, tetapi untuk

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

sistem yang lebih besar dan serius, aspek keamanan juga diterapkan dengan lebih ketat.

7. Kebersamaan pemakaian

Pemakai basis data sering kali tidak terbatas pada satu pemakai saja, atau disatu lokasi saja atau oleh satu sistem/ aplikasi saja. Basis data yang dikelola oleh sistem (*aplikasi*) yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data atau *deadlock* (Kustiyahningsih dan Anamisa, 2011).

2.8 MySQL

Database secara sederhana dapat kita sebut sebagai gudang data, secara teori *database* adalah kumpulan data atau informasi yang kompleks. Dalam basis data, *database* dapat kita kelola menggunakan program MySQL (Nugroho, 2008).

Didalam *database* dikategorikan menjadi 3 bagian sub perintah yaitu:

1. *Data Definition Language* (DDL), seperti perintah *Create* untuk membuat *database* baru, tabel baru dan *view* baru, *Alter* untuk merubah struktur tabel yang telah dibuat dan *Drop* untuk menghapus *database* dan tabel yang sudah dibuat.
2. *Data Manipulation Language* (DML), seperti perintah *Insert* untuk memasukkan data kedalam tabel, *Select* untuk mengambil data dari tabel, *Update* untuk mengubah data di tabel dan *Delete* untuk menghapus data di tabel.
3. *Data Control Language* (DCL), seperti perintah *Grant* yang digunakan untuk memberikan hak akses oleh *administrator* dan perintah *Revoke* yang digunakan untuk mencabut hak akses oleh administrator (Nugroho: 2008).

2.9 Tipe Data

Tipe data adalah suatu bentuk pemodelan data yang dideklarasikan pada saat melakukan pembuatan tabel. Tipe data ini akan mempengaruhi pada setiap data yang akan dimasukkan kedalam sebuah tabel. Data yang dimasukkan harus sesuai dengan *type* data yang dideklarasikan, secara umum database memiliki dua bentuk data yaitu data dengan karakter dan integer atau angka (Nugroho, 2009).

Ada tiga kelompok tipe data yang didukung oleh MySQL yaitu:

1. Data Numerik

Adapun tipe data Numerik pada MySQL yang akan digunakan pada sistem dapat dilihat pada Table 2.6 di bawah ini:

Tabel 2.6 Jenis Tipe Data Numerik

Type	Bytes	Keterangan
<i>Tinyint</i>	1	Type ini merupakan bentuk numeric yang paling kecil, type <i>TINYINT</i> mulai dari angka -128 sampai dengan 127
<i>Smallint</i>	2	<i>SMALLINT</i> yang memiliki penyimpanan data lebih besar dari <i>TINYINT</i> mulai dari -32768 sampai dengan 32767
<i>Mediumint</i>	3	Type <i>numeric</i> yang sederhana adalah <i>MEDIUMINT</i> mulai dari -8388608 sampai dengan 8388607
<i>Int</i>	4	<i>INT</i> merupakan tipe yang sangat sering digunakan dalam pembuatan <i>database</i> , karena tipe ini cukup menampung data yaitu -2147483648 sampai 2147483647
<i>Bigint</i>	8	Bentuk terbesar dalam data <i>numeric</i> mulai -9223372036854775808 sampai 9223370236854775807.

(Sumber: Nugroho, 2009)

2. Data untuk *Date* dan *Time*

Date dan *time* adalah salah satu bentuk data yang sangat penting dalam sebuah *database*, karena dengan menggunakan *date* dan *time* kita dapat

menyimpan data waktu dan tanggal, tanggal lahir, tanggal mulai dan waktu mulai dan sebagainya. (Nugroho, 2009). Tipe data *date* dan *time* dapat dilihat pada Tabel 2.7 di bawah ini.

Tabel 2.7 Jenis Tipe Data *Date* Dan *Time*

Type	Keterangan
<i>Date time</i>	yyyy-mm-dd hh:mm:ss'
<i>Date</i>	yyyy-mm-dd
<i>Time stamp</i>	Yyyy-mm-dd hh:mm:ss'
<i>Time</i>	'hh:mm:ss'
<i>Year</i>	Yyyy

(Nugroho, 2009)

3. Data *String*

Bentuk *String* adalah salah satu bentuk yang harus dimiliki dan dukungan oleh setiap *database*, data string dapat menyimpan semua data baik itu *numeric* maupun waktu dan tanggal. *String* dapat diartikan sebagai bentuk yang berupa karakter misalnya nama, alamat, email, pesan maupun semua data yang berbentuk data dari alphabet. (Nugroho, 2009). Tipe data string dapat dilihat pada Tabel 2.8 di bawah ini.

Tabel 2.8 Jenis *Type* Data *String*

Type	Jumlah karakter/huruf
<i>Char</i>	1 – 225
<i>Varchar</i>	1 – 225
<i>Tinytext</i>	1 – 225
<i>Text</i>	1 – 65535
<i>Mediumtext</i>	1 – 16777215
<i>Longtext</i>	1 – 4294967295

(Sumber: Nugroho, 2009)

2.10 PHP (*Hypertext preprocessor*)

Hypertext preprocessor (PHP) adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memprogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Contoh terkenal dari aplikasi PHP adalah CMS WordPress, forum (phpBB) dan Media Wiki (*software* di belakang Wikipedia). PHP juga dapat dilihat sebagai

