

BAB 2

TINJAUAN PUSTAKA

2.1 Pilkada

Sarundajang (2012) menuturkan bahwa, Pemilihan Kepala Daerah dan Wakil Kepala Daerah sudah dilaksanakan sejak tahun 2005 berdasarkan Undang-Undang Nomor 32 Tahun 2004 tentang pemerintah daerah (UU 32/2004). Kemudian sejak berlakunya Undang-Undang Nomor 22 Tahun 2007 (UU 22/2007) tentang penyelenggaraan pemilihan umum, pemilukada dimasukkan dalam rezim pemilu. Sehingga secara resmi bernama Pemilihan Umum Kepala Daerah dan Wakil Kepala daerah. Pemilukada pertama kali diselenggarakan berdasarkan Undang-Undang ini adalah pemilukada DKI Jakarta 2007.

Sejak itu pula nomenklturnya berubah menjadi Pemilihan Umum Kepala Daerah dan Wakil Kepala Daerah (pemilukada). Menurut UU 22/2007 di atas di pasal 1 butir 4 dikatakan bahwa yang dimaksud dengan pemilu Kepala Daerah dan Wakil Kepala Daerah adalah pemilu untuk memilih Kepala Daerah dan Wakil Kepala Daerah secara langsung dalam Negara Kesatuan Republik Indonesia berdasarkan Pancasila dan Undang-Undang Dasar Negara Republik Indonesia Tahun 1945.

Dasar hukum yang mendasari Pemilihan Kepala Daerah (pemilukada) langsung ini adalah amandemen kedua pada tahun 2002 pasal 18 UUD 45, ayat (4) yang menyebutkan Gubernur, Bupati, dan Walikota masing-masing sebagai Kepala Daerah Pemerintahan Daerah Provinsi, Kabupaten, dan Kota dipilih secara demokratis

.

2.2 *Quick count*

Menurut Juwairiah (2009), *quick count* adalah metode perhitungan secara cepat hasil pemilihan umum, yang datanya diperoleh dari Tempat Perhitungan Suara (TPS) sampel di lapangan. Berbeda dengan teknologi *polling*, sampel tidak diperoleh dari para *responden* yang ditanyai satu persatu, melainkan diperoleh dari hasil rekap resmi di lapangan. Dengan *quick count*, hasil perhitungan suara bisa diketahui dua sampai tiga jam setelah perhitungan suara di TPS ditutup.

Untuk mensukseskan sebuah perhitungan *quick count*, tiap-tiap lembaga memiliki teknologi yang berbeda-beda. Namun, teknologi *Short Message Service* (SMS) yang merupakan salah satu fitur dari *handphone* cukup populer digunakan oleh lembaga-lembaga penghitung *quick count*. Dahulu teknologi ini bukanlah bernama *quick count*, tetapi *Paralel Vote Tabulation* atau tabulasi suara pemilih secara paralel.

Menurut Mukhlas (2010), Terdapat beberapa cara kerja *quick count* yang umum dilakukan oleh para lembaga *survey* sebagai berikut:

1. Sistem

Pertama mempersiapkan perangkat keras serta sistem pendukung untuk bisa memberikan data secara cepat ke pusat pengolah data lembaga *survey* yang melakukan metode *quick count* ini. Perangkat ini mulai dari komputer untuk memasukkan data hingga ponsel untuk mengirim SMS hasil pemilu ke *server* tempat menerima data.

2. TPS

Pemilihan TPS sebagai tempat pengambilan data. TPS yang diambil secara acak berdasarkan pertimbangan jumlah penduduk, jumlah pemilih terbaru, penyebarannya pemilih seperti tersebar dalam berapa kelurahan, dan sebagainya.

3. Relawan

Mempersiapkan relawan untuk mengambil sampel dan meng-*input*-kannya ke sistem data. Jumlah relawan ini cukup banyak untuk mengambil data dari TPS yang telah dipilih.

4. Survey

Data yang telah didapat akan diolah di pusat data dengan menerapkan ilmu statistik, dari olahan data inilah lembaga *survey* bisa menghitung secara cepat siapa pemenang pemilu.

Jika dilihat dari cara kerja *quick count*, dapat diartikan bahwa hasil perhitungan *quick count* bukanlah hasil perhitungan dari seluruh TPS yang melakukan pemungutan suara, melainkan dengan menggunakan prinsip ilmu statistika. Jadi, lembaga *survey* yang menyelenggarakan *quick count* ini hanya mengambil sampel dari sekian banyak TPS yang ada dan diambil dari TPS yang memiliki jumlah populasi yang banyak dan berbagi pertimbangan lainnya. Walaupun hasil *quick count* ini tidak pernah tepat dan pasti, tetapi hasil dari *quick count* yang diselenggarakan oleh lembaga survei yang jujur tidak pernah meleset dari siapa yang memenangkan dari pemilihan umum tersebut.

2.3 *Short Message Service (SMS)*

Saputra (2011) menjelaskan, SMS merupakan fitur yang digunakan untuk berkiriman pesan dalam format teks. SMS ini dapat dinikmati oleh seluruh pengguna *handphone*. Dengan adanya SMS, dapat dipastikan bahwa tiap pesan yang masuk itu pasti terbaca oleh pemilik *handphone* tersebut ditambah sekarang tarif operator yang tarif SMS-nya murah bahkan gratis. Saat ini SMS digunakan tidak terbatas untuk komunikasi antar sesama operator saja, melainkan SMS dibuat secara otomatis menggunakan komputer, terlebih lagi menyangkut pengiriman dalam jumlah banyak (massal).

Menurut Wahidin (2010), layanan SMS memiliki beberapa keunggulan yaitu:

1. Mudah dan efisien

SMS mudah untuk digunakan bahkan oleh orang awam sekalipun serta efisien karena informasi dapat langsung diterima oleh yang berkepentingan.

2. Jangkauan luas

Informasi melalui media SMS dapat menjangkau ke seluruh nusantara hingga ke pelosok desa, hal ini tidak terlepas dari hampir sebagian masyarakat yang telah memiliki *handphone*.

3. Relatif lebih murah

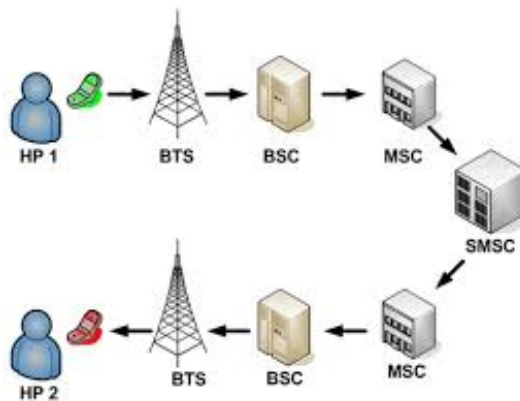
Biaya SMS relatif lebih murah bila dibandingkan dengan media penyampaian informasi seperti surat pos, telepon, dan fax karena SMS tidak membedakan tarif untuk lokal atau interlokal.

4. Informasi *realtime*

Informasi yang disampaikan melalui SMS dapat langsung diterima dalam hitungan detik, seandainya pun ada gangguan informasi tersebut tidak akan hilang dan tetap akan tersampaikan.

Saputra (2011) menjabarkan bahwa, setiap mengirimkan pesan melalui SMS, pesan tersebut tidak langsung sampai ke nomor *handphone* tujuan, tapi melewati beberapa proses terlebih dahulu yaitu pesan akan ditangkap oleh *Base Transceiver Station* (BTS) terlebih dahulu dilanjutkan ke *Base Station Controller* (BSC). Kemudian akan sampai ke tahap *Mobile Switching Centre* (MSC). MSC selanjutnya akan meneruskan atau *men-forward* pesan tersebut kepada *Short Message Service Center* (SMSC). Pada tahap inilah pesan tersebut disimpan untuk sementara jika nomor tujuan yang ditujukan sedang tidak aktif atau berada di luar jangkauan. Jika nomor tujuan sudah aktif maka akan diteruskan melewati MSC, BSC kemudian diterima oleh jaringan BTS nomor tujuan, kemudian dikirimkan kepada pengguna

nomor *handphone* tersebut.



Gambar 2.1: Alur Pengiriman SMS (Saputra,2011)

Pada umumnya terdapat 3 jenis SMS, menurut Wahidin (2010) yaitu :

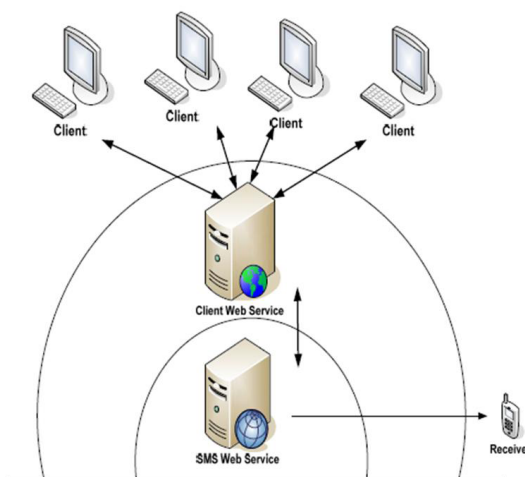
1. SMS premium yaitu jenis aplikasi SMS dua arah dengan menggunakan 4 digit nomor unik seperti 9090, 6278, 6288 dan sebagainya. Program disini dapat berupa SMS *quiz*, SMS *polling*, dan lain-lain.
2. SMS *broadcast* yaitu jenis aplikasi satu arah berupa pengiriman pesan ke banyak nomor sekaligus. Jenis SMS ini sangat cocok sebagai wahana untuk mempromosikan produk atau jasa kepada masyarakat. Ditambah lagi dengan memungkinkannya menggunakan pengirim berupa *alpha sender-id* (*sender-id* berupa nama organisasi atau institusi) dan SMS kampanye.
3. SMS *gateway* adalah jenis SMS dua arah, dengan keunikan bahwa semua tarif yang diberlakukan adalah tarif SMS normal sesuai dengan apa yang diberlakukan oleh operator.

2.3.1 SMS Gateway

Saputra (2011) menuturkan bahwa SMS *gateway* merupakan komunikasi dua arah, mengirim dan menerima SMS dan biasanya digunakan untuk SMS *keyword*,

polling, ataupun informasi lainnya. Wahidin (2010) juga menjelaskan bahwa, SMS *gateway* adalah SMS dengan keunikan bahwa semua tarif yang diberlakukan adalah tarif SMS normal sesuai dengan apa yang diberlakukan oleh operator. Karena sifatnya yang dua arah, maka jenis SMS ini sangat cocok dijadikan sebagai SMS *Center* organisasi atau institusi, dalam rangka meningkatkan kualitas komunikasi antara anggota komunitas organisasi atau institusi.

Katankar dan Thakare (2010) mengusulkan konsep desain layanan pesan singkat menggunakan SMS *gateway* dengan beberapa tingkatan yaitu tingkat pertama, dimana *user* akan mengakses aplikasi berbasis *web* dengan memasukkan informasi dasar pengiriman SMS seperti nomor penerima dan pesan yang dikirim ke penerima. Pesan dikirim ke tingkat kedua yaitu pada tingkat *web server* dimana akan mengubah pesan dengan menerapkan beberapa *header* keamanan untuk SMS dan menyebarkannya ke SMS *gateway* lalu pesan akan dikirim ke telepon seluler penerima.



Gambar 2.2: Konsep Layanan SMS Gateway (Katankar dan Thakare,2010)

Jika dilihat di kamus *gateway* berarti pintu gerbang. Pola kerja SMS *gateway* yaitu dengan cara mengubah pesan pada lalu lintas jaringan seluler dari media lain atau

sebaliknya. SMS *gateway* memungkinkan transmisi atau penerimaan pesan SMS dengan atau tanpa menggunakan telepon seluler. Contoh penerapan SMS *gateway* adalah untuk meneruskan *e-mail* kepada telepon seluler penerima. Sistem ini dikembangkan untuk meningkatkan keamanan *gateway* pengguna. Pesan juga dapat dikirim ke ratusan nomor tanpa harus mengetik nomor satu persatu karena semua nomor akan diambil secara otomatis dari *database* tersebut.

SMS *gateway* kadang dibutuhkan lebih dari satu modem untuk berbagai macam urusan yang berbeda-beda. Karena pengakses SMS cukup banyak dan multi operator, maka dibutuhkan layanan yang murah pulsa juga untuk semua operator. Tentu SMS ke sesama operator akan jauh lebih murah dibandingkan dengan berbeda operator. Tujuannya adalah mengurangi pengeluaran pulsa.

2.4 Operator Telepon Seluler

Telepon seluler (ponsel) atau *handphone* (HP) adalah sebuah perangkat telekomunikasi elektronik tanpa kabel, yang dapat dibawa kemana-mana. Telepon seluler pertama kali ditemukan oleh seorang karyawan Motorola, Martin Cooper pada tanggal 03 April 1973. Saat ini Indonesia mempunyai dua jaringan telepon nirkabel yaitu sistem GSM (*Global System for Mobile Telecommunications*) dan sistem CDMA (*Code Division Multiple Access*). Telepon seluler berfungsi untuk melakukan dan menerima panggilan telepon, umumnya juga berfungsi dalam pengiriman dan penerimaan pesan singkat atau sering disebut juga SMS (*Short Message Service*), pesan gambar, *video call*, hingga televisi online. (<http://id.wikipedia.org/wiki/Telepon-genggam>)

Operator adalah perusahaan atau pihak penyelenggara penyedia atau pemberi jasa tertentu. Operator telepon seluler adalah pihak penyelenggara jaringan dan layanan telepon seluler. Saat ini di Indonesia operator telepon seluler GSM adalah Telkom-

sel (Produksi SIM *card* Simpati, As, Halo), Indosat (Produsi SIM *card* Mentari, IM3, Matrix), Excelcomindo (Produksi SIM *card* XL Bebas, XL Jempol), Hutchison (Produksi SIM *card* Three '3'), Natrindo Telepon seluler (Produksi SIM *card* Axis). Sementra operator telepon seluler CDMA adalah Telkom Flexy, Mobile-8, Bakrie Telekom Esia, Sampoerna Ceria, Smart Telecom, Indosat Starone. (<http://www.elektroindonesia.com>)

Berikut Kode-kode Operator Seluler :

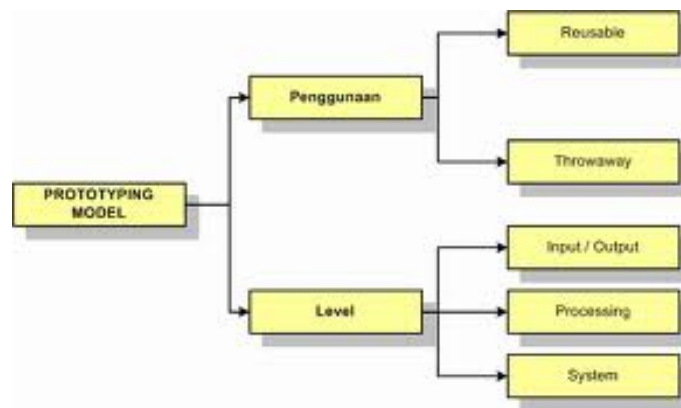
Telkomsel	
0811 Kartu HALO	0812 SimPATI, Kartu HALO
0813 Kartu HALO	0821 SimPATI
0822 SimPATI	0823 Kartu As
0852 Kartu As	0853 Kartu As
Indosat	
0814 Indosat	0815 Mentari, Matrix
0816 Mentari, Matrix	0855 Matrix
0856 IM3	0857 IM3
0858 Mentari	
XL	
0817 XL Prabayar, XL Pascabayar	0818 XL Prabayar, XL Pascabayar
0819 XL Prabayar, XL Pascabayar	0859 XL Prabayar, XL Pascabayar
0877 XL Prabayar	0878 XL Prabayar
0879 XL Prabayar	
Sampoerna Telekom	
0828 Ceria	
Smartfren Telekom	
0881 Smartfren	0882 Smartfren
0883 Smartfren	0884 Smartfren
0887 Smartfren	0888 Smartfren
0889 Smartfren	
Hutchison	
0896 3	0897 3
0898 3	0899 3

Gambar 2.3: Kode Operator Seluler

2.5 Metode *Prototype*

Mulyono dalam bukunya Rekayasa Perangkat Lunak (2008) menyebutkan, bahwa *prototype* merupakan salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau komponen-komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan. (Howard,1997)

Model *prototype* dapat diklafikasikan menjadi beberapa tipe seperti terlihat pada gambar 2.4 berikut :



Gambar 2.4: Klafikasi *prototyping* model (Harris,2003)

Harris (2003) menjelaskan, bahwa tahap-tahap dalam *prototype* merupakan tahap-tahap yang dipercepat. Strategi utama dalam *prototype* adalah kerjakan yang lebih mudah dahulu dan sampaikan hasil kepada pengguna sesegera mungkin. Ada enam tahapan yang terdapat pada pembagian *prototype*.

Tahapan-tahapnya dapat dijelaskan sebagai berikut:

1. Identifikasi kandidat *prototyping*.

Kandidat dalam kasus ini meliputi *user interface*, *file-file* transaksi utama, dan fungsi-fungsi pemrosesan sederhana.

2. Rancangan bangun *prototyping*.

Dalam melakukan perancangan sistem dibutuhkan bantuan *software* seperti *word processor*, *database*, pengolahan grafik, dan sebagainya

3. Uji *prototyping*.

Tahap ini untuk memastikan *prototype* dapat dengan mudah dijalankan untuk tujuan demonstrasi.

4. Evaluasi dengan USD (*User System Diagram*).

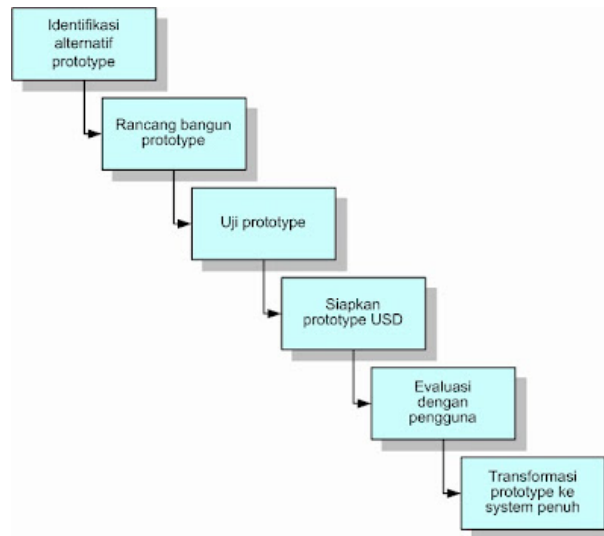
Dalam tahap ini dilakukan identifikasi bagian-bagian dari perangkat lunak yang *prototype*.

5. Evaluasi dengan pengguna.

Tahap ini digunakan untuk mengevaluasi *prototype* dan melakukan perubahan jika diperlukan.

6. Transformasikan *prototype*.

Pada tahap ini transformasi *prototype* menjadi perangkat lunak yang beroperasi penuh dengan melakukan penghilangan kode-kode yang tidak dibutuhkan, penambahan program-program yang memang dibutuhkan dan perbaikan kemudian melakukan pengujian perangkat lunak secara berulang.



Gambar 2.5: Tahapan-tahapan *prototyping* model (Harris,2003)

2.6 Gammu

Menurut Wahidin (2010), Gammu adalah nama sebuah proyek yang ditujukan untuk membangun aplikasi, *script* dan *drivers* yang dapat digunakan untuk semua fungsi yang memungkinkan pada telepon seluler atau alat sejenisnya. Gammu bukanlah suatu aplikasi, melainkan sebagai salah satu modul *SMS gateway* yang dapat digabungkan dengan bahasa pemrograman apa saja. Gammu merupakan *codebase* yang stabil dan sempurna untuk berbagai macam model telepon yang tersedia di pasaran. Gammu sendiri bekerja dengan *command line* atau sebagai *daemon*.

Berikut ini beberapa kelebihan dari Gammu:

1. Dapat dijalankan baik di *Operating System* (OS) Windows maupun LINUX based.
2. Mendukung banyak perangkat yang cocok atau sesuai, baik itu HP maupun GSM Modem.
3. Menggunakan MySQL *database*, yang tentunya juga *open-source*.

4. *Interface* untuk aplikasi, bisa menggunakan *client-server* maupun *web based* (PHP,ASP,ASP.Net maupun J2EE)

Gammu mendukung berbagai macam model telepon seluler dengan berbagai jenis koneksi dan *type*. Beberapa contoh HP dan modem beserta jenis koneksinya pada tabel 2.1 :

Tabel 2.1 macam-macam jenis HP atau modem dan koneksinya

Merk/Type	Connection	Merk/Type	Connection
Nokia 3110 classic	at115200	Sony Ericsson J300i	at115200
Nokia E51	at115200	Sony Ericsson C702	at19200
Nokia N95	at115200	Sony Ericsson C702	at19200
Nokia 6070	Fbus	Sony Ericsson K800i	at19200
Nokia 6820	bluephonet	LG CU-500	at115200
Nokia 6030	fbuspl2303	LG KE820	at19200
Nokia 7120 supernova	bluephonet	LG KE970	at19200
Nokia 5130 Xpress-Music	bluephonet	LG KF750	at115200
Nokia 5300	bluephonet	LG KG300	at19200
Nokia N73	dku2phonet	LG CU-500	at115200
Samsung SGH- i607	at19200	LG KE820	at19200
Samsung SGH- Z300	at19200	LG KE970	at19200
Samsung SGH- Z400	at19200	Huawei E160	at19200
Samsung SGH- Z150	at115200	Huawei E160e	at19200
Samsung SGH- E310	at19200	Huawei E160G	at115200
Samsung SGH- E770	at19200	Huawei E169	at19200
MOTOROLA K1	at19200	Huawei E176G	at115200
MOTOROLA L6	at19200	Huawei E220	at19200
MOTOROLA L7	Atblueat	Huawei E226	at19200

Sumber : <http://www.gammu.org/support/phone/phonedb.php>

2.7 Konfigurasi Gammu

Gammu memerlukan konfigurasi untuk dapat berinteraksi dengan perangkat telepon ataupun modem GSM. File konfigurasi untuk gammu dapat berisi beberapa bagian yang masing-masing bagian mengkonfigurasi satu perangkat mo-

dem/telepon. Dalam kondisi *default*, gammu mencoba semua konfigurasi dalam urusan *numeric* ([gammu],[gammu1],[gammu N],...)Saputra (2010).

Saputra (2011) juga menuturkan, ada dua *file* utama dalam gammu yang perlu di konfigurasikan , diantaranya:

1. Gammurc, merupakan tempat konfigurasi dari perangkat yang digunakan. Ada dua penting yang perlu diingat yaitu nomor *port* yang terhubung adalah com5 dengan *maximum port speed* adalah 115200. Kedua data tersebut pasti hasilnya akan berbeda-beda.
2. SMSdrc, file ini berfungsi untuk menampung konfigurasi *database*.

Gammu memiliki konfigurasi yang berbeda-beda untuk parameternya. Parameter utama adalah parameter untuk koneksi (*connection*) dan perangkat (*device*). *Connection* merupakan suatu *protocol* yang digunakan untuk berbicara atau berkomunikasi dengan perangkat telepon ataupun modem.

Parameter yang berhubungan langsung dengan perangkat adalah parameter *device*. Parameter ini mulai ada pada gammu versi 1.27.95. pada versi sebelumnya, nama parameter ini adalah *port*. Untuk kabel atau *serial port*, yaitu dengan memasukkan nama perangkatnya. Contoh :

Di linux : port = /dev/ttyUSB0

Di Windows : port= com5

2.8 Konfigurasi SMSD

Menurut wahidin (2010), gammu SMSD adalah sebuah *daemon* yaitu suatu program yang berjalan dalam sebuah sistem. Gammu SMSD membaca konfigurasi dari *file config*. Lokasi ini dapat dispesifikasikan pada *command line*, dinyatakan

default path yang digunakan */etc/gammu smsdrc*. Konfigurasi file *gammu SMSD* terdiri dari setidaknya dua bagian yaitu *[gammu]* dan *[SMSD]*.

Gammu SMSD harus dijalankan setiap kali ingin menggunakan aplikasi SMS. Saputra (2011) menuturkan bahwa, terlebih dahulu *gammu SMSD* diinstal agar dapat dijalankan di dalam *service*. Kemudian buka jendela *command prompt* lalu ketikkan *gammu-SMSd.exe c SMSdrc I* maka *service* *gammu SMSD* akan terinstal di dalam *service*.

2.9 Hypertext Preprocessor (PHP)

Purwanto (2001) menjelaskan bahwa, PHP di ciptakan oleh Rasmus Lerdorf, seorang pemrograman C yang handal. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada *homepage*-nya. Rasmus adalah salah seorang pendukung *open source*. Pada tahun 1996, PHP telah digunakan oleh banyak *website* di dunia. PHP adalah salah satu bahasa *server-side* yang paling populer, setara dalam jumlah pemakaian dengan *mod perl*, di bawah CGI dan ASP.

Anhar (2010) juga menjelaskan bahwa, PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server side HTML embedded scripting*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru (*up to date*). Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

Kriteria yang harus diperhatikan dalam penulisan *skrip* PHP, menurut Nugroho (2004) yaitu sebagai berikut :

1. Setiap halaman yang mengandung skrip PHP harus disimpan dengan bentuk ekstensi PHP sesuai dengan program PHP yang mendukungnya : *namafile*.

php.

2. Setiap skrip PHP harus didahului dengan pembuka PHP (<?) dan kemudian diakhiri dengan penutup (?>).
3. Setiap baris skrip isi didahului pernyataan cetak yang dibedakan menjadi dua yaitu print dan echo.
4. Setiap akhir baris perintah harus diakhiri dengan titik koma(;).
5. Semua bentuk variabel harus diawali tanda dolar.
6. Penulisan *comment* atau keterangan didahului dengan tanda pembuka (*) dan diakhiri dengan tanda (*) atau menggunakan tanda *slash* ganda (/) yang hanya digunakan untuk menuliskan pesan yang informasinya satu baris saja.
7. Semua *listing program* HTML yang akan digabungkan dalam skrip PHP harus dihilangkan tanda petik ganda dan dapat digantikan dengan tanda petik tunggal atau menghilangkannya.

Purwanto (2001) juga menuturkan bahwa, PHP mudah dibuat dan cepat dijalankan. PHP dapat berjalan dalam *web server* yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi LINUX, UNIX, Windows, Windows NT dan Macintosh. Menurut Kasiman (2006), PHP tidak terbatas pada hasil keluaran HTML (*Hypertext Markup Languages*). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, *file* PDF, movies Flash dan dapat menghasilkan text seperti XHTML dan *file* XML lainnya.

2.10 MySQL

Kadir (2011), menjelaskan MySQL dikenal sebagai *database server* terutama semenjak internet menjadi populer. *Software* ini banyak digunakan pada berbagai

website. MySQL tergolong sebagai *open source* dan sangat handal, sehingga menjadi pilihan dalam mengelola data berbagai organisasi. Selain itu, MySQL dapat berjalan pada beberapa *platform*, dari Windows sampai ke Linux.

Menurut Saputra (2011), MySQL merupakan perangkat lunak yang juga bersifat *open source*. Sesuai namanya, bahasa standar yang digunakan adalah SQL. SQL *Structure Query Language* merupakan bahasa standar untuk pengolahan *database*.

Program-program aplikasi yang mendukung MySQL:

1. PHP
2. Borland Delphi, Borland C++ Builder
3. Visual Basic 5.9/6.0 dan .Net
4. Visual FoxPro
5. Cold Fusion, dan masih banyak lagi (Nugroho, 2008).

Anhar (2010) menyatakan beberapa kelebihan MySQL, antara lain:

1. MySQL dapat berjalan dengan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, dan masih banyak lagi.
2. Bersifat *open source*, MySQL didistribusikan secara *open source* (gratis), di bawah lisensi GNU *General Public Licence* (GPL).
3. Bersifat *multiuser*, MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
4. MySQL memiliki kecepatan yang baik dalam menangani *query* (perintah SQL). Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.

5. Dari segi *security* atau keamanan data, MySQL memiliki beberapa lapisan *security*, seperti *level subnet mask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* yang terenskripsi.
6. MySQL juga memiliki *interface* (antarmuka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

2.11 Web Server

Menurut Solichin (2009), *web server* merupakan sebuah *software* dalam server yang berfungsi menerima permintaan berupa halaman web melalui HTTP atau HTTPS dari *clien* yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. Beberapa *web server* yang banyak digunakan di internet yaitu *Apache Web Server*, *Internet Information Service (IIS)*, *Xitami Web Server* dan *Sun Java System Web Server*.