BAB II

TINJAUAN PUSTAKA

2.1 Kota Cimahi

Tanggal 18 Oktober 2001 terbentuklah Kota Cimahi yang disahkan oleh Menteri Dalam Negeri. Kewenangan Kota Cimahi sebagai Daerah Otonom mencakup seluruh kewenangan bidang pemerintahan, termasuk kewenangan wajib yaitu pekerjaan umum, kesehatan, pendidikan dan kebudayaan, perhubungan, industri dan perdagangan, penanaman modal, lingkungan hidup, pertahanan, koperasi dan tenaga kerja kecuali bidang politik luar negeri, pertahanan keamanan, peradilan, moneter fisikal, agama serta kewenangan bidang lain sesuai dengan peraturan Perundang-undangan Nomor I tahun 2003 tentang Kewenangan Kota Cimahi sebagai Daerah Otonom. [1]

2.1.1 Visi Kota Cimahi Tahun 2007-2012

"Dengan Iman, Taqwa, Optimis dan Cerdas, Jadikan Cimahi Kota Maju, Agamis, Nyaman, Tertib, Aman dan Produktif"

2.1.2 Misi Kota Cimahi

1. Meningkatkan sarana perekonomian dan lapangan kerja
2. Meningkatkan kualitas pendidikan dan Kesehatan
3. Meningkatkan penataan dan Penegakan hukum
4. Meningkatkan infrastruktur Kota

5. Mengendalikan pembangunan agar berwawasan lingkungan
6. Meningkatkan kemitraan dengan dunia usaha

2.1.3 Strategi Pokok Pembangunan.

Untuk mewujudkan visi dan misi Kota Cimahi tahun 2007 – 2012 , pemerintah Kota Cimahi harus mengembangkan dua strategi pokok penyelenggaraan pemerintahan dan pembangunan yaitu strategi internal dan strategi eksternal serta fokus pada pembangunan.

2.2 Dinas Perhubungan Kota Cimahi

Dinas Perhubungan sebagai salah satu organisasi Perangkat Pemerintah Kota yang bertugas untuk merumuskan dan melaksanakan kebijakan teknis operasional dibidang Angkutan, Lalu lintas dan Teknik Sarana serta melaksanakan urusan ketatausahaan dinas. [2]

2.2.1 Visi Dinas Perhubungan

" Menjadi pengelola Transportasi yang Profesional dalam rangka menunjang Mobilitas Masyarakat Kota Cimahi yang Tertib dan Lancar."

Penetapan visi ini didasarkan bahwa untuk mencapai cita – cita bangsa Indonesia pada umumnya yaitu masyarakat yang sejahtera lahir dan batin harus menjadi cita – cita yang harus diperjuangkan.

Transportasi merupakan perpindahan orang/barang dari Tempat asal menuju tempat tujuan dengan melalui prasarana jalan & rel.

Profesional adalah memiliki kemampuan teknis di bidang transportasi, yang mampu bersaing, mampu menggunakan teknologi dan mampu menghadapi perubahan dan tantangan.

Mobilitas yaitu gerakan kendaraan di jalan dan rel.

Masyarakat adalah pengguna jasa transportasi.

Tertib dalam arti kata selalu mentaati aturan – aturan yang berlaku dalam berlalu lintas & berkomunikasi.

Lancar dalam arti kata pergerakan lalu lintas tidak mengalami hambatan yang berarti.

2.2.2 Misi Dinas Perhubungan

1. Mewujudkan manajemen transportasi perkotaan yang terpadu sesuai dengan Tata Ruang Kota Cimahi
2. Meningkatkan keselamatan Lalu lintas melalui pelayanan aparatur, penyediaan sarana prasarana dan peningkatan kesadaran masyarakat dalam berlalu lintas

2.2.3 Tujuan Dinas Perhubungan

1. Terwujudnya sistem jaringan transportasi jalan yang efektif & efisien guna memenuhi kebutuhan masyarakat.
2. Meningkatkan kualitas aparatur perhubungan dan disiplin masyarakat dalam rangka menciptakan Lalu lintas yang tertib & teratur.
3. Meningkatkan kualitas & kuantitas sarana & prasarana dalam rangka menunjang pelayanan bagi masyarakat.

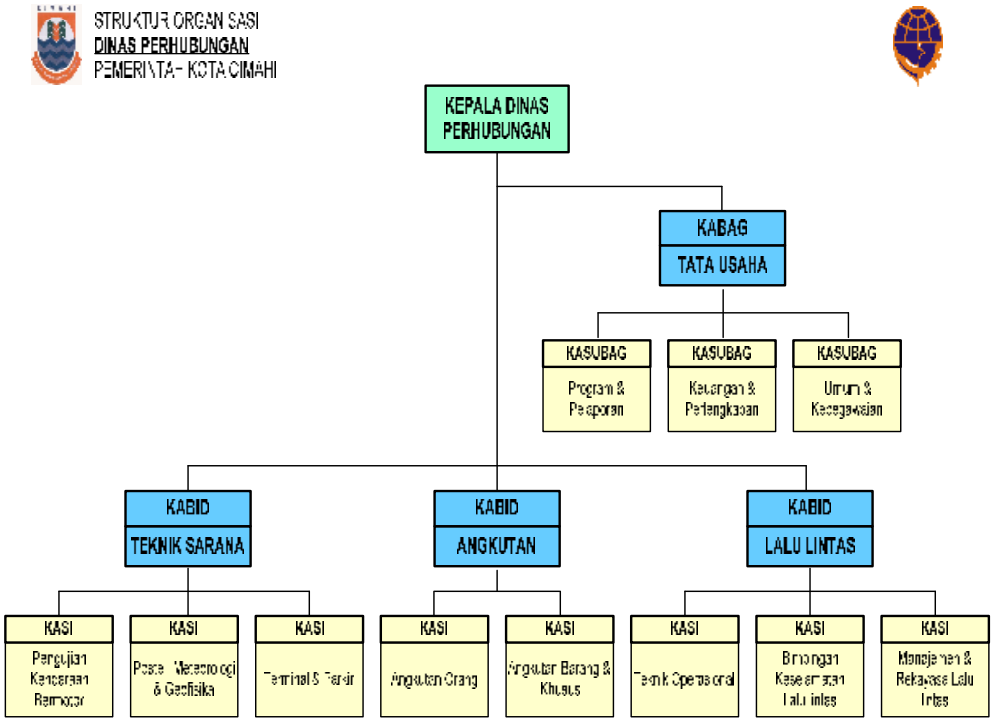
2.2.4 Sasaran Dinas Perhubungan

Sasaran adalah hasil yang akan dicapai secara nyata oleh instansi pemerintah dalam rumusan yang lebih spesifik, terukur, dalam kurun waktu yang lebih pendek dari tujuan. Sasaran yang ingin dicapai oleh Dinas Perhubungan Kota Cimahi diantaranya sebagai berikut :

1. Terwujudnya keterpaduan intra dan antar moda transportasi darat sesuai kebutuhan masyarakat.
2. Meningkatnya pengetahuan dan keterampilan aparatur di bidang teknis dan Fungsional
3. Meningkatnya pengetahuan dan keterampilan aparatur dibidang administrasi.
4. Meningkatnya disiplin masyarakat dalam berlalulintas.
5. Meningkatnya fasilitas perlengkapan & fasilitas pendukung lalu lintas jalan.
6. Meningkatnya fasilitas simpul transportasi (Gd PKB, Terminal, Stasiun KA)
7. Terselenggaranya Pelayanan Perhubungan
8. Terselenggaranya komputerisasi pelayanan perijinan

2.2.5 Struktur Organisasi Dinas Perhubungan

Dibawah ini merupakan struktur organisasi yang ada di Dinas Perhubungan Kota Cimahi saat ini.



Gambar 2.1 Stuktur Organisasi Dinas Perhubungan

Susunan organisasi Dinas Perhubungan kota Cimahi dapat dilihat sebagai berikut

1. Kepala Dinas
2. Kepala Bagian Tata Usaha, membawahi :
   a). Kasubag. Program dan Pelaporan
   b). Kasubag. Umum dan Kepegawaian
   c). Kasubag. Keuangan dan Perlengkapan
3. Kepala Bidang Angkutan, membawahi ;
   a). Kepala Seksi Angkutan Orang
   b). Kepala Seksi Angkutan Barang dan Khusus
4. Kepala Bidang Lalu Lintas, membawahi :

a). Kepala Seksi Manajemen dan Rekayasa Lalu Lintas

b). Kepala Seksi Teknik Bimbingan Keselamatan

c). Kepala Seksi Teknik Operasional

5. Kepala Bidang Teknik Sarana, membawahi :

a). Kepala Seksi Parkir dan Terminal

b). Kepala Seksi Pengujian Kendaraan dan Perbengkelan

c). Kepala Seksi Pos,Telekomunikasi, Meteorologi dan Geofisika

6. Kelompok Jabatan Fungsional

2.3 Landasan Teori

2.3.1 Sistem Informasi Geografis (SIG)

Pada dasarnya, istilah sistem informasi geografis merupakan gabungan dari tiga unsur pokok : sistem, informasi dan geografis. Dengan demikian, pengertian terhadap tiga unsur-unsur pokok ini sangat membantu dalam memahami SIG.

Sistem informasi adalah suatu sistem untuk menyajikan informasi guna mendukung fungsi operasi, manajemen dan pengambilan keputusan dalam organisasi. Dengan memperhatikan pengertian sistem informasi, maka SIG adalah sistem yang berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografis. SIG dirancang untuk mengumpulkan, menyimpan dan menganalisis objek-objek dan fenomena dimana lokasi geografis merupakan karakteristik yang penting untuk dianalisis. Dengan demikian, SIG merupakan sistem komputer yang mempunyai empat kemampuan berikut dalam menangani data yang bereferensi geografis : (a) masukan, (b)

manajemen data (penyimpanan dan pemanggilan data), (c) analisis dan manipulasi data, (d) keluaran.

SIG juga merupakan sejenis perangkat lunak yang digunakan untuk pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran informasi geografis berikut atribut-atributnya. [2]

2.3.1.1 Subsistem SIG

SIG dapat diuraikan menjadi beberapa subsistem berikut :

1. Data Input : subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spansial dan atribut dari berbagai sumber. Subsistem ini pula yang bertanggungjawab dalam mengkonversi atau mentransformasikan format-format data-data aslinya kedalam format yang dapat digunakan oleh SIG.
2. Data Output : Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basisdata dalam bentuk softcopy maupun dalam bentuk hardcopy seperti : tabel, grafik, peta dan lain-lain.
3. Data Management : Subsistem ini mengorganisasikan baik data spansial maupun atribut kedalam sebuah basisdata sedemikian rupa sehingga mudah dipanggil, di-update dan di-panggil.
4. Data manipulation dan Analysis : Sub sistem ini menentukan inforasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

2.3.1.2 Komponen SIG

Sistem SIG terdiri dari beberapa komponen berikut :

1. Perangkat keras

   Pada saat ini SIG tersedia untuk berbagai *platform* perangkat keras yang dapat digunakan oleh banyak orang secara bersamaan dalam jaringan komputer yang luas, berkemampuan tinggi, memiliki ruang penyimpanan yang besar, dan mempunyai kapasitas memori yang besar. Walaupun demikian, fungsionalitas SIG tidak ierikat secara ketat terhadap karakteristik-karakteristik fisik perangkat keras sehingga keterbatasan memori pada PC-pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk SIG adalah komputer (PC), *mouse, digitizer*,*printer*,*plotter* dan *scanner*.

2. Perangkat lunak

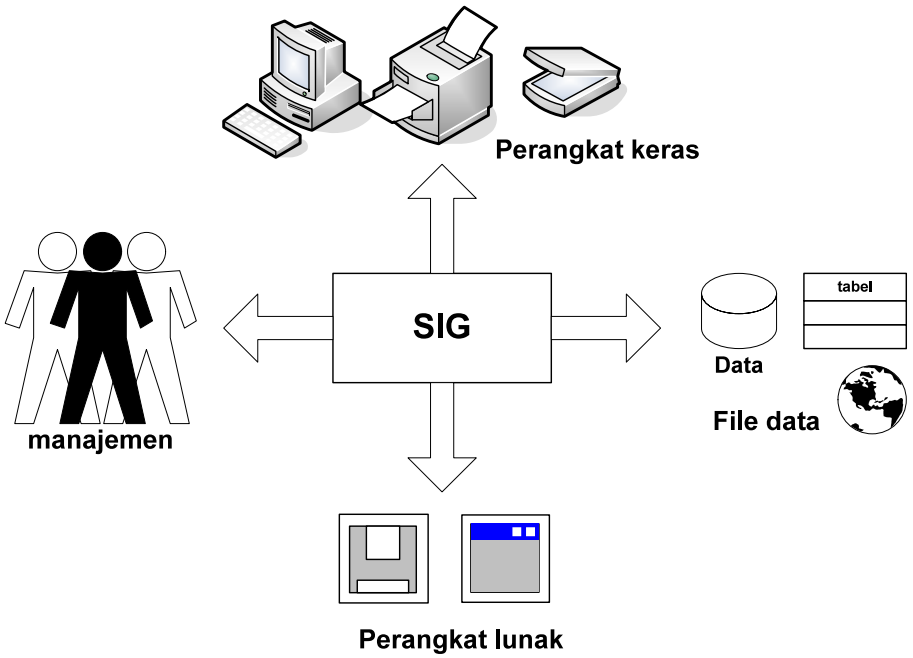
   SIG juga merupakan perangkat lunak yang tersusun secara modular dimana basisdata memegang peranan kunci.

3. Data & informasi geografis

   SIG data mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara tidak langsung dengan cara meng-*import*-nya dari perangkat-perangkat lunak SIG yang lain maupun secara langsung dengan cara mendijitasi data spasialnya dari peta dan memasukkan data atributnya dari tabel-tabel dan laporan dengan menggunakan *keyboard.*

4. Manajemen

   Proyek SIG akan berhasil jika di-*manage* dengan baik dan dikerjakan oleh orang-orang yang memiliki keahlian yang tepat pada semua tingkatan. [2]



Gambar 2.2 Komponen SIG

2.3.2 Pemrograman Berbasis Objek

Pemrograman berbasis objek adalah metode implementasi dimana diorganisasikan sebagai kumpulan objek yang bekerja sama, masing-masing objek mempresentasikan *instant* dari kelas, dan kelas-kelas itu anggota suatu hirarki kelas-kelas yang disatukan lewat keterhubungan pewarisan.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat. Pemrograman orientasi objek menekankan konsep berikut:

1. Class

   Kumpulan atas definisi sata dan fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah *class* secara tipikal sebaiknya dapat dikenali oleh seorang *non-programmer* sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalan sebuah *class* sebaiknya bersifat mandiri dan independen. Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyerderhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. Objek

   Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer, objek merupakan dasar dari sebuah modularitas dan struktur dalam sebuah program komputer berorientasi objek.

3. Abstraksi

   Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagian model dari ”pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek yang lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. Enkaptulasi

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak, hanya metode dalam objek dengan cara yang tidak layak, hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. Polimorfisme

   Melalui penerimaan pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan, metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu dimana pesan tersebut dikirim.

6. Inheritas

   Mengatur *polimorfisme* dan *enkaptulasi* dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi perilaku mereka tanpa harus mengimplementasikan ulang perilaku tersebut (bahasa berbasis objek tidak selalu memiliki *inheritas*).

2.3.3 UML (*Unified Modelling Language*)

UML adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. UML dapat digunakan untuk

membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Karena UML menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak pada bahasa berorientasi objek.

2.3.3.1 Tujuan Perancangan UML

Perancangan UML mempunnyai beberapa tujuan utama. Tujuan-tujuan tersebut adalah:

1. Menyediakan bahasa pemodelan visual yang *ekspresif* dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk bahasa pemodelan.
5. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

2.3.3.2 Diagram dan Teknik Pemodelan UML

Diagram menggambarkan banyak hal, penggunaan notasi yang terdefinisi baik dan *ekspresi*f adalah penting pada proses pengembanngan perangkat lunak, yaitu:

1. Notasi standar memungkinkan pengembang mendeskripsikan rumusan arsitektur dan kemudian membuat komunikasi secara tidak ambigu.

2. Notasi yang bagus membebaskan otak untuk berkonsentrasi pada masalah-masalah yang lebih lanjut.
3. Notasi yang baik memungkinkan melakukan *eliminasi* keperluan *konsistensi* dan kebenaran keputusan menggunakan alat yang otomatis.

UML menyediakan beberapa diagram *visual* yang menunjukan berbagai aspek dalam sistem. Diagram-diagram yang disediakan dalam UML adalah sebagai berikut:

a. Diagram kelas (*Class diagram*)

   Diagram ini menunjukan sekumpulan kelas, *interface* dan kolaborasi dan keterhubungannya. Diagram kelas ditujukan untuk pandangan statik terhadap sistem.

b. Diagram objek (*Object diagram*)

   Diagram ini menunjukan sekumpulan objek dan keterhubungannya. Diagram ini menunjukan potongan statik dari instan-instan yang ada di diagram kelas. Diagram ini untuk memperlihatkan satu prototipe atau kasus tertentu yang mungkin terjadi. Diagram objek menyediakan notasi grafis formal guna memodelkan objek, kelas, dan saling keterhubungan. Diagram objek berguna untuk *abstract modelling* dan perancangan program-program sesungguhnya. Pada pendekatan ini, bentukan dasar dari sistem perangkat lunak adalah objek atau kelas.

c. Diagram *use-case* (*Use case diagram*)

   Diagram ini menunjukan sekumpulan kasus fungsional dan aktor (jenis kelas khusus) dan keterhubungannya.

d. Diagram sekuen (*Sequence diagram*)

   Diagram ini menunjukan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis urutan waktu dari pesan-pesan yang terjadi.

e. Digram kolaborasi (*Collaboration diagram*)

   Diagram ini juga merupakan diagram interaksi. Diagram ini menekankan pada organisasi struktur dari objek-objek yang mengirim dan menerima pesan.

f. Diagram *statechart* (*Statechart diagam*)

   Diagram ini berisi status, transisi, kejadian dan aktivitas. *Statechart* merupakan pandangan dinamis dari sistem. Diagram ini penting dalam memodelkan perilaku antarmuka, kelas, kolaborasi dan menekankan pada urutan kejadian.

g. Diagram aktivitas (*Activity diagram*)

   Diagram ini untuk menunjukan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem. Diagram ini penting untuk memodelkan fungsi sistem dan menekankan pada aliran kendali di antara objek-objek.

h. Diagram komponen (*Component diagram*)

   Diagram ini menunjukan organisasi dan hubungan ketergantungan diantara sekumpulan komponen. Diagram ini merupakan pandangan statik terhadap implementasi sistem.

i. Diagram pengembangan (*Deployment diagram*)

   Menunjukan konfigurasi pemrosesan saat jalan dan komponen-komponen yang terdapat di dalamnya. Pilihan model dan diagram yang digunakan

dipengaruhi oleh bagaimana persoalan ditangani dan bagaimana solusi dibentuk. Setiap sistem komplek perlu didekati melalui sekumpulan pandangan model yang hampir *independent*.

2.3.4 PhP

PHP yang juga dikenal sebagai "*Hypertext Preprocessor*", adalah sebuah bahasa pemrograman yang digunakan untuk pengolahan data-data berbasis web. Tujuan utama bahasa ini adalah untuk mempermudah pembuatan halaman web yang dinamik atau mudah diupdate dan cepat dalam pengolahan datanya. Dalam aplikasinya php dapat digabungkan dengan sintak HTML. Sintaks PHP mirip dengan bahasa C dan memiliki beberapa fungsi PHP tersendiri.

PHP pertama kali dibuat oleh Rasmus Lerdroft, seorang programmer C. Semula PHP digunakannya untuk menghitung jumlah pengunjung di dalam webnya. Kemudian ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan script PERL yang dibuatnya untuk membuat halaman webnya menjadi dinamis. Kemudian pada tahun1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah mampu mengakses database dan dapat terintegrasi dengan HTML.

Pada tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang softwarenya. Versi yang kemudian yaitu PHP 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah

terintegrasinya Zend Engine yang dibuat oleh Zend Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP3 *scripting engine*. Yang lainnya adalah build in HTTP session, tidak lagi menggunakan library tambahan seperti pada PHP3. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Dalam hal ini, aplikasi akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan oleh web server.

Versi terbaru adalah PHP 5.0 yang pengembangannya di sisi keamanan dan mengarah ke pemrograman yang terstruktur dan bersifat OOP(*Object Oriented Program*) yaitu pemrograman yang berorientasi ke object.

Ketika web dinamik konten/isi semakin berkembang dan membutuhkan update yang cepat, situs-situs yang statispun semakin ditinggalkan Karena dianggap sudah tidak memenuhi keinginan pasar karena situs tersebut harus tetap dinamis setiap hari. Pada saat ini bahasa PERL dan CGI sudah jauh ditinggalkan sehingga sebagian besar programmer web banyak beralih ke bahasa server-*side scripting* yang lebih dinamis seperti PHP.

Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan system database di dalam web. Sistem database yang dapat didukung oleh PHP antara lain :

1. MySQL
2. Oracle
3. Sybase

4. PostgreSQL
5. SQL lite

PHP dapat berjalan di berbagai system operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh

2.3.5 MySql

MySQL adalah suatu perangkat lunak database relasi (*Relational Database Management System* atau RDBMS), seperti halnya ORACLE, Postgresql, MS SQL, dan sebagainya. MySQL AB menyebut produknya sebagai database open source terpopuler di dunia. Berdasarkan riset dinyatakan bahwa bahwa di *platform* Web, dan baik untuk kategori *open source* maupun umum, MySQL adalah database yang paling banyak dipakai. Menurut perusahaan pengembangnya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya.

2.3.5.1 Keunggulan MySQL

Penyebab utama MySQL begitu populer di kalangan Web adalah karena ia memang cocok bekerja di lingkungan tersebut. Pertama, MySQL tersedia di berbagai *platform* Linux dan berbagai varian Unix. Sesuatu yang tidak dimiliki Access, misalnya padahal Access amat popular di *platform* Windows. Banyak server Web berbasiskan Unix, ini menjadikan Access otomatis tidak dapat dipakai karena ia pun tidak memiliki kemampuan client-server/networking.

Kedua, fitur-fitur yang dimiliki MySQL memang yang biasanya banyak dibutuhkan dalam aplikasi Web. Misalnya, klausa LIMIT SQL-nya, praktis untuk melakukan paging. Atau jenis indeks field FULLTEXT, untuk *full text searching*. Atau sebutlah kekayaaan fungsi-fungsi builtinnya, mulai dari memformat dan memanipulasi tanggal, mengolah string, *regex*, *enkripsi* dan *hashing*. Yang terakhir misalnya, praktis untuk melakukan penyimpanan password anggota situs.

Ketiga, MySQL memiliki *overhead* koneksi yang rendah. Soal kecepatan melakukan transaksi atau kinerja di kondisi load tinggi mungkin bisa diperdebatkan dengan berbagai *benchmark* berbeda, tapi kalau soal yang satu ini MySQL-lah juaranya.

2.3.6 MapServer

2.3.6.1 Pengantar MapServer

Dalam pengertian teknis yang paling dasar, MapServer adalah program CGI yang terpasang dan berjalan tapi tidak aktif dalam server (aktif hanya saat dipanggil). Saat rekues/permintaan dikirimkan ke mapserver, maka akan digunakan informasi yang dikirimkan lewat URL dan mapfile untuk membuat (*generate*) peta yang diinginkan. Permintaan ini bisa juga termasuk permintaan untuk membuat legenda, peta referensi, batang skala, dan variabel lain yang dikirimkan ke CGI tadi.

Mapserver sangat bisa dikembangkan dan dikustomisasi. Juga bisa dibangun untuk mendukung berbagai format data input dan output. Ini dilakukan saat file biner mapserver dikompilasi. Silakan kunjungi situs utama Mapserver untuk

melihat daftar fitur lengkapnya. Banyak juga fitur yang didukung tetapi tidak secara built-in, dan bisa dipanggil dengan penggunaan OGR.

2.3.6.2 MapScript

Mapscript menyediakan antar muka pemrograman (scripting) pada Mapserver untuk membangun aplikasi webmapping. Penggunaan Mapscript tidak bergantung pada Mapserver CGI, artinya walaupun Mapserver CGI tidak ada tetapi jika mapscript tersedia, anda masih bisa mengembangkan webmapping berbasis mapserver dengan mapscript ini. Mapscript merupakan modul tambahan untuk bahasa pemrograman umum yang kita kenal untuk memanggil fungsi-fungsi yang disediakan mapserver.

2.3.6.3 Anatomi Aplikasi MapServer

Sebuah aplikasi mapserver sederhana terdiri dari:

1. **Mapfile** - adalah file konfigurasi terstruktur untuk aplikasi mapserver anda. Berisi konfigurasi wilayah peta anda, lokasi data dalam server dan format output dan lainnya. Juga berisi layer-layer peta anda, sumber data, proyeksi dan simbologi. Biasanya adalah file teks dengan ekstensi **.map**.
2. **Data Geografis** - Mapserver bisa menggunakan beberapa jenis/format sumber data geografis. Format utama yang didukung adalah ESRI Shapefile.
3. **Halaman HTML** - merupakan jembatan antara user dan mapserver. Berupa file HTML dalam server/hosting anda. Mapserver bisa dipanggil untuk membuat peta statis dalam halaman HTML anda, atau untuk membuatnya interaktif anda bisa membuat form dalam HTML ini.

CGI adalah program yang 'stateless', setiap rekues yang dikirim adalah selalu baru, dia tidak bisa mengingat apa yang sudah dilakukannya sebelumnya lewat permintaan anda. Itulah sebabnya, aplikasi anda harus mengirim terus parameter-parameter yang diinginkan (layer apa saja, di wilayah mana peta ditampilkan, mode aplikasi, dsb) yang bisa ditentukan lewat variabel tersembunyi dalam form atau lewat URL.

Aplikasi/form mapserver sederhana bisa terdiri dari dua halaman HTML:

a. **File inisialisasi** - menggunakan form dengan variabel tersembunyi untuk mengirim kueri awal ke webserver dan mapserver. Form ini bisa ditempatkan secara terpisah atau digantikan dengan mengirim variabel lewat URL.
b. **File templat** - untuk mengatur bagaimana tampilan peta, legenda dan informasi lainnya disajikan dalam halaman HTML lewat browser. Dengan mengirim referensi variabel Mapserver CGI pada templat HTML, anda akan meminta mapserver menerjemahkannya menjadi nilai variabel bergantung pada status terakhirnya (misalnya nama peta, peta referensi, besar/cakupan peta, dsb). Templat ini juga akan menentukan bagaimana nantinya pengunjung berinteraksi dengan aplikasi mapserver (perbesaran/pengecilan, menggeser peta, kueri, dsb).

4. **Mapserver CGI** - file biner yang dieksekusi untuk menerima permintaan/rekues dan menerjemahkannya menjadi gambar peta, data,

dsb. Berada di direktori cgi-bin atau direktori lain yang ditentukan pada webserver. Di RuangWeb.com ini berada di /cgi-sys pada domain anda.

5. **Webserver** - aplikasi yang menyajikan halaman HTML pada browser. RuangWeb.com menggunakan Apache sebagai webservernya.