



UNIVERSITAS INDONESIA

**RANCANG BANGUN SIMULASI PENGOLAHAN DATA
RADAR MENGGUNAKAN MAPSERVER UNTUK APLIKASI
MONITORING LALU LINTAS UDARA**

SKRIPSI

MUHAMMAD KAUKAB
0606042771

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPARTEMEN TEKNIK ELEKTRO
DEPOK
DESEMBER 2008**



UNIVERSITAS INDONESIA

**RANCANG BANGUN SIMULASI PENGOLAHAN DATA
RADAR MENGGUNAKAN MAPSERVER UNTUK APLIKASI
MONITORING LALU LINTAS UDARA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana

MUHAMMAD KAUKAB

0606042771

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPARTEMEN TEKNIK ELEKTRO
DEPOK
DESEMBER 2008**

ii

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Muhammad Kaukab

NPM : 0606042771

Tanda Tangan :

Tanggal : 23 Desember 2008

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Muhammad Kaukab
NPM : 0606042771
Program Studi : Teknik Elektro
Judul Skripsi : Rancang Bangun Simulasi Pengolahan Data Radar
Menggunakan *Mapserver* Untuk Aplikasi
Monitoring Lalu Lintas Udara

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima
sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar
Sarjana Teknik pada Program Studi Teknik Elektro
Fakultas Teknik, Universitas Indonesia**

DEWAN PENGUJI

Pembimbing : Ir. Ny. Hj. Rochmah M. Eng. Sc. ()

Penguji : Dr. Ir. Arman Djohan Diponegoro M. Eng ()

Penguji : Dr. Ir. Dodi Sudiana M. Eng ()

Ditetapkan di : Depok
Tanggal : 23 Desember 2008

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada :

- (1) Ir. Ny. Hj. Rochmah N Sukardi M. Eng. Sc, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dan memberikan banyak masukan dalam penyusunan skripsi ini;
- (2) Dr. Ir. Arman Djohan M. Eng, selaku dosen penguji yang telah membimbing dan menyediakan waktu, tenaga, dan pikiran dalam membimbing dan mengevaluasi hasil dari skripsi ini;
- (3) Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral;
- (4) Muhammad Haekal, ST selaku kakak dan pihak PT. Angkasa Pura II, Bandara Soekarno Hatta yang telah banyak membantu dalam usaha memperoleh data yang saya perlukan, dan banyak memberi masukan dalam proses pengerjaan skripsi ini; dan
- (5) Sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu pengetahuan di Indonesia.

Depok, 23 Desember 2008

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Kaukab
NPM : 0606042771
Program Studi : Teknik Telekomunikasi
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**“Rancang Bangun Simulasi Pengolahan Data Radar Menggunakan
Mapserver Untuk Aplikasi Monitoring Lalu Lintas Udara”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 23 Desember 2008

Yang menyatakan

(Muhammad Kaukab)

Muhammad Kaukab
Departemen Teknik Elektro

Dosen Pembimbing
Ir. Ny. Hj. Rochmah M. Eng. Sc

**RANCANG BANGUN SIMULASI PENGOLAHAN DATA RADAR
MENGUNAKAN MAPSERVER UNTUK APLIKASI MONITORING
LALU LINTAS UDARA**

ABSTRAK

Skripsi ini membahas mengenai sebuah rancang bangun simulasi pengolahan data radar dengan menggunakan sistem pemetaan *Map server* berbasis *web*. Aplikasi ini merupakan pengembangan teknologi yang ditujukan pada sistem monitoring lalu lintas udara. Pengembangan ini mempermudah kinerja sistem monitoring lalu lintas udara sehingga dapat dilakukan dengan mudah tanpa melihat batas lokasi. Dengan menggunakan pemetaan *Mapserver* berbasis *web* memudahkan sistem untuk dipetakan sesuai dengan standar koordinat yang terhubung pada *web server* dengan memanfaatkan teknologi jaringan komputer. Informasi monitoring data radar ini akan ditampilkan melalui *web server* dalam bentuk *web* yang dapat dengan mudah diakses oleh *user-user* tertentu.

Kata kunci: Radar, *Map Server*, *Web Server*

Muhammad Kaukab Electrical Engineering Department	Counselor Ir. Ny. Hj. Rochmah M. Eng. Sc
--	---

**DESIGN SIMULATION BUILT USING RADAR DATA PROCESSING
MAPSERVER APPLICATION FOR MONITORING AIR TRAFFIC**

ABSTRACT

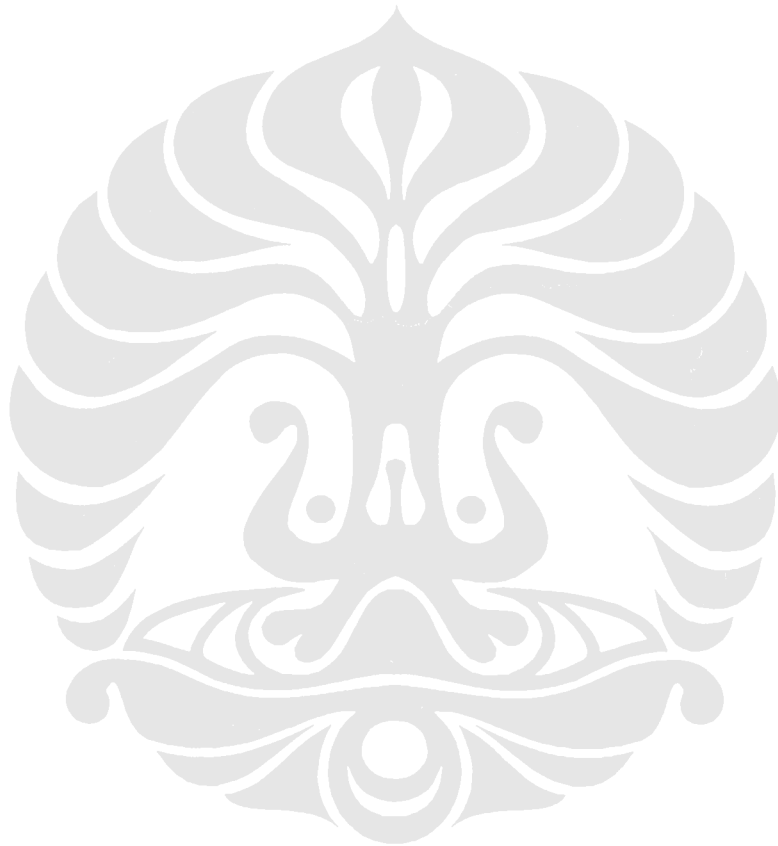
This essay discusses the design of a wake simulation data processing system by using radar mapping of Map-based web server. This application is the development of technology aimed at the system of monitoring air traffic. Facilitate the development of this system of monitoring the performance of air traffic so it can be done easily without limit locations. By using mapping Map server facilitate web-based system to be in accordance with the standards of coordinates that is connected to the web server by using computer network technology. Information monitoring radar data will be displayed through a web server in the form of web that can be easily accessed by certain users.

Keywords : Radar, Map Server, Web Server

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
LEMBAR PENGESAHAN	iv
KATA PENGANTAR	v
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
ABSTRAK	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR SINGKATAN	xiii
1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Batasan Masalah	3
1.4. Metodologi Penyelesaian Masalah	3
1.5. Sistematika Penulisan	4
2. DASAR TEORI	5
2.1. Sistem Komunikasi Radar	5
2.2. Peralatan Radar	7
2.2.1. Klasifikasi Sistem Radar	7
2.2.2. PSR dan SSR	10
2.3. Sistem Informasi Geografis (SIG)	11
2.3.1. Definisi SIG	11
2.3.2. Sub Sistem Utama SIG	13
2.4. Sistem Jaringan Komputer	14
2.4.1. Jenis Jaringan Komputer yang Digunakan.....	14
2.4.2. Protokol	15
2.4.3. Hardware yang Dibutuhkan	16
3. PERANCANGAN SISTEM MONITORING RADAR MENGUNAKAN MAP SERVER	19
3.1. Perancangan Sistem Radar Dalam Pengembangan	19
3.2. Spesifikasi Radar Dalam Perencanaan	20
3.3. Perancangan Sistem <i>Hardware</i> dan <i>Software</i>	21
3.3.1. Aplikasi Kondisi <i>Real</i>	21
3.3.2. Aplikasi Kondisi Simulasi	23
3.3.2.1. Komponen Sebagai <i>RDP server</i>	24
3.3.2.2. Komponen Sebagai <i>Server</i>	26
3.3.2.3. Komponen Sebagai <i>Client</i>	30
3.4. Diagram Alir Sistem Monitoring Data Radar	31
4. ANALISIS PENGOLAHAN DATA RADAR	33
4.1. Format Data Radar yang Dipergunakan	34

4.2. Pengujian <i>Sample Data</i>	35
4.3. Analisis Program	41
5. KESIMPULAN	47
6. DAFTAR ACUAN	48
7. DAFTAR PUSTAKA	50
8. LAMPIRAN	51



DAFTAR TABEL

Tabel 2 Perbedaan setiap protokol dengan karakteristiknya	16
Tabel 4.1 Format <i>Secondary Message</i>	34
Tabel 4.2 Format Kode <i>Message</i>	35
Tabel 4.3 Penjelasan Posisi X	37
Tabel 4.4 Penjelasan Posisi Y	38
Tabel 4.5 Penjelasan Mode <i>Flight Level</i>	39
Tabel 4.6 Penjelasan Mode <i>SSRCode</i>	39



DAFTAR GAMBAR

Gambar 2.1 Blok Diagram Radar Primer.....	6
Gambar 2.2 Blok diagram alur sinyal radar primer	6
Gambar 2.3 Skema Pengelompokkan Radar	8
Gambar 2.4 Komponen-komponen SIG	12
Gambar 2.5 Penggunaan Protokol Ethernet	16
Gambar 2.6 Penggunaan Protokol Token Ring	17
Gambar 2.7 Protokol FDDI dengan semua <i>stationnya</i> berfungsi	18
Gambar 2.8 Protokol FDDI dengan satu <i>stationnya</i> down	18
Gambar 2.9 Perangkat pada jaringan Komputer	20
Gambar 2.10 Perangkat Hub/Switch	22
Gambar 2.11 Perangkat Repeater	22
Gambar 3.1 Skema pembahasan perancangan radar	25
Gambar 3.2 Alur data aplikasi monitoring data Radar	26
Gambar 3.3 Sistem monitoring data radar kondisi <i>real</i>	27
Gambar 3.4 Sistem monitoring data Radar dalam kondisi simulasi	28
Gambar 3.5 Tampilan raw data dari <i>RDP Server</i>	30
Gambar 3.6 Tampilan data matang lebih spesifik dari <i>RDP Server</i>	31
Gambar 3.7 Tampilan <i>database server</i> sebagai pengolah data	32
Gambar 3.8 Tampilan monitoring data radar pada <i>client</i>	35

DAFTAR SINGKATAN

ATC	: Air Traffic Control
ATM	: Asynchronous Transfer Mode
CSMA/CD	: Carrier Sense Multiple Access/Collision Detection
CW – Radar	: Continuous Wave – Radar
FDDI	: Fiber Distributed Data Interface
FL	: Flight Level
GIS	: Geography Information System
ISP	: Internet Service Providers
LAN	: Local Area Network
MAN	: Metropolitan Area Network
NIC	: Network Interface Cards
Pos X	: Posisi X
Pos Y	: Posisi Y
PPI	: Plan Position Indicator
PSR	: Primer Surveillance Radar
RADAR	: Radio Detection and Ranging
RAID	: Redundant Array of Inexpensive Disks
RDP	: Radar Data Processing
SIG	: Sistem Informasi Geografis
SSR	: Secondary Surveillance Radar
Transponder	: Transmitting Responder
VMWare	: Virtual Machine Ware
WAN	: Wide Area Network

BAB I PENDAHULUAN

1.1. Latar Belakang

Radar merupakan salah satu teknologi yang cukup berkembang dalam dunia penerbangan. Radar dapat menggantikan fungsi mata manusia untuk memantau objek pada jarak jauh dalam jangkauan yang telah ditentukan dalam spesifikasinya. Radar merupakan suatu sistem yang memanfaatkan gelombang elektromagnetik untuk mengidentifikasi jarak, ketinggian, arah, maupun kecepatan benda yang diam atau bergerak. Istilah radar diciptakan pada tahun 1941 yang merupakan singkatan dari *Radio Detection and Ranging*. Radar digunakan dalam banyak konteks, termasuk dalam bidang pengendalian lalu lintas udara (*Air Traffic Control*) maupun laut, yaitu untuk mendeteksi pesawat ataupun kapal, baik ketika berada di landasan/dermaga maupun di udara/laut.

Selain teknologi radar yang saat ini sedang berkembang adalah teknologi SIG (Sistem Informasi Geografis). SIG atau juga dikenal sebagai *Geographic Information System* (GIS) telah mengalami perkembangan yang berarti, seiring kemajuan teknologi informasi. Bergulirnya peningkatan kebutuhan akan perlunya informasi pemetaan bumi dalam rangka pengelolaan sumber daya alam, menjadi pemicu peningkatan SIG di Indonesia. SIG merupakan sistem informasi berbasis komputer yang menggabungkan antara unsur peta (geografis) dan informasinya tentang peta tersebut (data atribut) yang dirancang untuk mendapatkan, mengolah, memanipulasi, analisa, memperagakan dan menampilkan data spasial untuk menyelesaikan perencanaan, mengolah dan meneliti permasalahan. Dengan definisi ini, maka terlihat bahwa aplikasi SIG dilapangan cukup luas terutama bagi bidang yang memerlukan adanya suatu sistem informasi tidak hanya menyimpan, menampilkan, dan menganalisa data atribut saja, tetapi juga unsur geografisnya.

Salah satu aplikasi teknologi GIS dapat dikembangkan untuk aplikasi radar, dimana GIS yang akan digunakan adalah berbasis *web* (*Web-GIS*). *Web-GIS* merupakan salah satu wujud perkembangan GIS untuk keperluan yang lebih luas sehingga dapat dilakukan secara *online*. Salah satunya adalah mengembangkan Web-GIS untuk sistem monitoring radar. Salah satu sistem Web-GIS yang populer saat ini adalah *Mapserver*. *Mapserver* merupakan aplikasi

freeware dan *open source* yang memungkinkan untuk menampilkan data spasial (peta) di *web*. Saat ini karena sifatnya yang terbuka (*open source*), pengembangan *Mapserver* dilakukan oleh pengembang dari berbagai negara, termasuk perkembangannya untuk sistem monitoring radar.

Dalam hal ini radar digunakan untuk sistem navigasi yang digunakan pada lalu lintas bandar udara. Untuk rancang bangun pengolahan data radar ini dibuat dengan bantuan *software Mapserver* yang berfungsi untuk sistem pemetaannya, sehingga pergerakan objek (dalam hal ini pesawat terbang) dapat terpantau dengan baik dalam ruang lingkup pergerakan yang luas.

1.2. Tujuan

Tujuan pembuatan karya tulis skripsi yang berjudul **“Rancang Bangun Simulasi Pengolahan Data Radar Menggunakan *Mapserver* Untuk Aplikasi Monitoring Lalu Lintas Udara”** ini adalah:

1. Memberikan penjelasan serta pemahaman mengenai sistem komunikasi Radar yang digunakan sistem navigasi pada aplikasi lalu lintas udara.
2. Mengaplikasikan secara langsung disiplin ilmu Teknik Telekomunikasi pada perancangan dan pembuatan sistem monitoring data radar dengan perkembangannya menggunakan *Mapserver* sebagai proses *monitoringnya*
3. Merancang bangun dalam pembuatan sistem monitoring radar dengan menggunakan program *Mapserver* sebagai sistem pemetaannya untuk *monitoring* data radar berbasis *web*.
4. Mengembangkan teknologi-teknologi pendukung yang akan menjalankan *Mapserver* sebagai salah satu perangkat lunak untuk menampilkan data radar dalam *monitoringnya*.
5. Untuk memenuhi syarat kelulusan dalam memperoleh Gelar Sarjana.

1.3. Batasan Masalah

Mengingat permasalahan pada sistem komunikasi radar mempunyai ruang lingkup yang luas dan agar pembahasan lebih terarah dan lebih khusus, maka permasalahan ini akan dibatasi pada beberapa hal. Pada dasarnya pembahasan tentang sistem komunikasi radar ini difokuskan penggunaannya sebagai alat

navigasi untuk komunikasi lalu lintas udara. Adapun bahasannya dapat melingkupi :

1. Parameter-parameter yang digunakan dalam perancangan dan pembuatan sistem monitoring data radar yang digunakan dalam sistem lalu lintas udara (*Air Traffic Control*).
 2. Rancang bangun sistem pemetaan dalam monitoring pergerakan pesawat dengan menggunakan *Mapserver* untuk menampilkan data Radar secara *real time* (dalam hal ini diambil beberapa cuplik data Radar)
 3. Merancang bangun sistem basis data yang mengkoordinasikan data Radar yang diperoleh untuk ditampilkan dalam pemetaan dengan menggunakan *Mapserver*
- Adapun beberapa hal lainnya yang disinggung dalam karya tulis skripsi ini hanya merupakan suatu informasi saja karena sulit dipisahkan dengan permasalahan pokok.

1.4. Metodologi Penyelesaian Masalah

Metodologi dalam penyelesaian masalah yang telah dilakukan dalam pembuatan karya tulis skripsi ini adalah :

1. Metode observasi, yaitu metode yang dilakukan dengan cara pengamatan langsung ke bandar udara Internasional Soekarno-Hatta.
2. Metode eksperimen, yaitu merancang pemetaan yang digunakan serta menganalisa sistem dan cara kerjanya, serta melakukan percobaan dengan perangkat lunak yang telah dibuat.
3. Metode kepustakaan, yaitu dengan membaca buku-buku referensi yang berkaitan dengan skripsi ini.
4. Metode diskusi, yaitu dengan melakukan diskusi tidak hanya dengan dosen pembimbing, namun juga pada pihak-pihak yang berpengalaman secara teknis.

1.5. Sistematika Penulisan

Pada penulisan karya tulis skripsi ini dibagi dalam 5 bab, sehingga dapat menjelaskan dan menguraikan permasalahan yang ada dalam penyelesaian skripsi ini. Pembagian bab terdiri dari :

BAB I PENDAHULUAN

Pada bab pendahuluan ini berisikan tentang latar belakang, tujuan, batasan masalah, metodologi penyelesaian masalah, dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini memaparkan pengetahuan mengenai sistem komunikasi radar, *software Mapserver*, serta teknologi-teknologi penunjang dalam fungsinya untuk menjalankan *Mapserver* sebagai salah satu perangkat lunak untuk menampilkan data radar dalam sistem *monitoring* data radar.

BAB III PERANCANGAN SISTEM MONITORING RADAR MENGUNAKAN *MAP SERVER*

Bab ini berisikan tentang perancangan sistem *hardware* serta *software* penunjang dalam menampilkan data radar secara visual dengan menggunakan *Mapserver*.

BAB IV ANALISIS DATA

Bab ini berisikan tentang analisis data yang diperoleh dan diaplikasikan kembali ke dalam program *Mapserver* hingga membentuk sebuah fungsi monitoring data radar dengan teknik baru.

BAB V PENUTUP

Berisikan tentang kesimpulan dari pembahasan masalah yang telah diuraikan pada BAB II, BAB III, BAB IV dan saran-saran.

BAB II DASAR TEORI

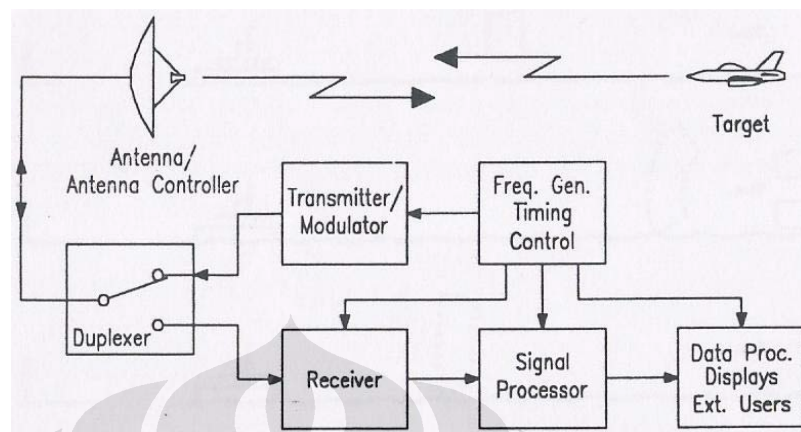
2.1 Sistem Komunikasi Radar

Kata RADAR merupakan akronim dari frase *Radio Detection And Ranging*. Dari ketiga kata tersebut dapat dikombinasikan menjadi sebuah sistem yang memiliki fungsi khusus pada penggunaannya dalam sistem keamanan. Dalam beberapa tahun ini, radar telah kehilangan arti aslinya menjadi representasi dari setiap sensor elektromagnetik aktif, yang beroperasi dalam porsi gelombang mikro dari spektrum gelombang elektromagnetik, yang menggunakan sumbernya sendiri untuk mengiluminasi sebuah region ruang angkasa kemudian mengukur energi yang dipantulkan (*reflective energy*) yang dibangkitkan (*generated*) oleh target yang teriluminasi di daerah itu.

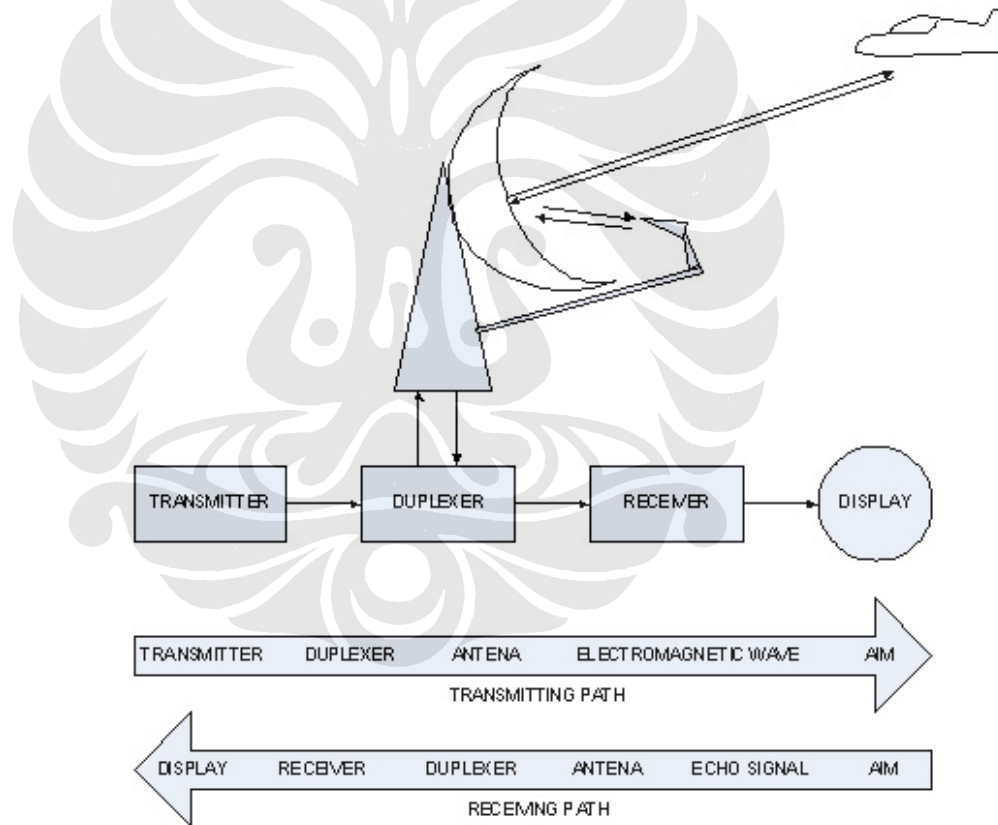
Radar merupakan sebuah sistem penginderaan jauh yang aktif, yang menyediakan sumber iluminasinya sendiri. Gelombang radio ditransmisikan sebagai pulsa dengan energi tinggi dari energi *microwave* ke arah bumi. Pulsa berinteraksi dengan atmosfer dan target. Porsi dari energi yang dipancarkan kembali, diterima oleh target kemudian diukur intensitas dan waktu penundaan diantara transmisi dan penerimaan sinyal yang kembali.

Pendeteksian intensitas *backscatter portion* dari energi yang dipantulkan oleh target untuk mengambil bagian. *Ranging* dapat terpenuhi dengan melakukan pengukuran jeda waktu dari pulsa dengan durasi pendek yang ditransmisikan oleh radar. Sebuah radar juga mampu mengukur *range*/posisi dari target yang diiluminasi beserta kecepatan radialnya.

Pengukuran kecepatan radial (*radial velocity*) dari sebuah target yang bergerak, direalisasikan dengan cara mengukur pergeseran frekuensi Doppler (*doppler frequency shift*), yang diproduksi oleh target, dimana perbedaan sinyal yang dipancarkan dan sinyal yang diterima akan dihitung. Kekuatan dan bentuk pulsa dari radar juga akan membawa informasi tentang karakteristik bentuk dan materialnya, dari target yang dipantulkan. Pada gambar 2.1 dan 2.2 menampilkan blok diagram sinyal radar yang sederhana.



Gambar 2.1. Blok diagram radar primer



Gambar 2.2 Blok diagram alur sinyal radar primer

Pada gambar diatas terdapat sifat monostatis pada *transmitter* dan *receiver* radar dengan lokasi yang sama. Pada sistem ini terdapat antena tunggal diantara proses transmisi dan penerimaan. Pada sistem *dual* antena juga bersifat monostatis dengan proses transmisi dan penerimaan pada lokasi yang sama.

Antena radar memancarkan sinyal gelombang mikro (*microwave*) pada sasaran objek. Sasaran tersebut akan memantulkan kembali sinyal *microwave* tersebut kepada alat penerima. Kemudian sinyal listrik diteruskan oleh antena penerima yang disebut *echo*. Sinyal radar itu sendiri dihasilkan oleh *transmitter* yang kuat dan diterima oleh penerima yang sangat sensitif.

Seluruh target menghasilkan pantulan yang berpenyerakan, dan pantulan ini disebarkan ke segala arah. Sinyal yang direfleksikan disebut juga *scattering*. *Backscatter* merupakan hasil refleksi yang berlawanan dengan pancaran yang terjadi.

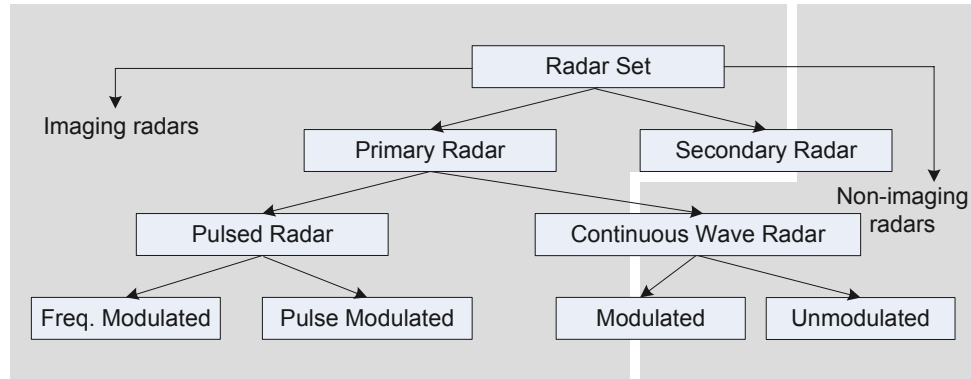
Sinyal radar dapat ditampilkan pada *Plan Position Indicator* (PPI) atau pada tampilan sistem radar yang lain. Sebuah PPI memiliki sebuah vektor rotasi dengan radar pada sumber, dimana mengindikasikan arah tujuan dari antena dan sudut awal dari target.

2.2 Peralatan Radar

2.2.1 Klasifikasi Sistem Radar

Radar adalah sistem yang digunakan untuk mendeteksi, mengukur jarak dan membuat *map* benda-benda seperti pesawat dan hujan. Gelombang radio kuat dikirim dan sebuah penerima mendengar gema yang kembali. Dengan menganalisa sinyal yang dipantulkan, pemantul gema dapat ditentukan lokasinya dan kadang-kadang ditentukan jenisnya. Walaupun sinyal yang diterima kecil, akan tetapi sinyal radio dapat dengan mudah dideteksi dan diperkuat.

Gelombang radio radar dapat diproduksi dengan kekuatan yang diinginkan, mendeteksi gelombang yang lemah dan kemudian diperkuat beberapa kali. Radar mempunyai teknologi dan kualitas yang berbeda-beda tergantung pada informasi yang diinginkan. Skema pengelompokan radar tersebut terlihat pada gambar 2.3 dibawah ini.



Gambar 2.3 Skema pengelompokan radar

a. *Imaging Radar / Non-Imaging Radar*

Suatu *imaging* radar membentuk suatu gambar menyangkut *area* atau obyek yang diamati. *Imaging radar* telah digunakan untuk memetakan bumi, planet, asteroid, dan benda angkasa lain serta untuk menggolongkan target pada sistem militer. Implementasi yang khas dari suatu sistem radar *Non-Imaging* adalah radar pengukur tinggi dan kecepatan. Ini disebut juga sebagai *scatterometers*, semenjak adanya pengukuran properti *scattering* pada objek atau daerah yang sedang diamati.

b. *Radar Primer*

Suatu radar primer memancarkan sinyal frekuensi tinggi yang dipantulkan pada target. Gema yang telah tiba selanjutnya diterima dan dievaluasi. Ini berarti bahwa radar primer tidak sama dengan radar sekunder, yang mana radar primer menerima sinyal yang dipancarkannya sebagai suatu gema.

c. *Radar Sekunder*

Pada radar sekunder, pesawat udara harus mempunyai suatu *transponder* (*transmitting responder*) diatas pesawat dan *transponder* ini bereaksi terhadap interogasi oleh pemancaran suatu kode sinyal jawaban. Respon ini dapat berisi lebih banyak informasi, dibanding suatu unit radar primer yang bisa memperoleh suatu ketinggian, suatu kode identifikasi atau juga permasalahan teknis manapun diatas pesawat seperti *loss* suatu *radio contact*.

d. Pulsed Radar

Pulsed radar memancarkan suatu impuls sinyal frekuensi tinggi bertenaga tinggi. Setelah sinyal ini, suatu retakan lebih panjang mengikuti di mana gema dapat diterima, sebelum suatu sinyal yang baru dipancarkan. Arah, jarak dan kadang-kadang jika perlu ketinggian atau tinggi target dapat ditentukan dari pengukuran posisi antena dan waktu penyebaran sinyal pulsa.

e. Continuous-Wave Radar (CW Radar)

CW Radar mentransmisikan suatu sinyal frekuensi tinggi secara terus-menerus. Gema sinyal diterima dan diproses. Penerima tidak perlu dipasang pada tempat yang sama dengan *transmitter*. Tiap-Tiap perusahaan pemancar radio sipil dapat bekerja sebagai suatu pemancar radar pada waktu yang sama, jika suatu penerima jarak jauh membandingkan waktu propagasi pada *direct* sinyal dan sinyal pantul. Melalui uji coba diketahui bahwa penempatan yang benar dari suatu pesawat udara dapat dihitung dari evaluasi sinyal oleh tiga stasiun televisi berbeda.

f. Unmodulated CW Radar

Sinyal yang dipancarkan dari peralatan ini mempunyai amplitudo dan frekuensi yang tetap. Peralatan ini dikhususkan dalam pengukuran kecepatan, contoh penggunaan perangkat ini adalah sebagai pengukur kecepatan oleh polisi.

g. Modulated CW Radar

Sinyal yang dipancarkan mempunyai amplitudo yang tetap, sedangkan frekuensinya dimodulasikan. Keuntungan peralatan ini adalah evaluasinya dilaksanakan tanpa jeda dan hasil pengukurannya tersedia secara terus-menerus. Radar ini digunakan pada jarak pengukuran yang tidak terlalu besar dan memerlukan suatu pengukuran berlanjut, seperti pada pengukuran ketinggian pesawat udara atau sebagai radar cuaca /*wind profiler*.

2.2.2 PSR dan SSR

a. PSR (*Primer Surveillance Radar*)

Unit radar primer mempunyai kualitas terbaik. Radar ini bekerja dengan gema pasif. Impulse frekuensi tinggi yang dipancarkan terpantul oleh target dan kemudian diterima oleh unit radar yang sama. Jadi penyebab gema yang dipantulkan adalah pemancaran *impulse* yang dikirimkan oleh unit radar.

b. SSR (*Secondary Surveillance Radar*)

Unit radar sekunder mempunyai prinsip kerja yang berbeda dengan radar primer. Radar sekunder bekerja dengan sinyal jawab aktif. Unit radar sekunder memancarkan dan juga menerima *impuls* frekuensi tinggi, yang disebut *Interogation* atau penyelidikan. Frekuensi tersebut tidak dipantulkan, tetapi diterima oleh target dengan suatu *transponder* yang dapat menerima dan memproses. Setelah target menjawab dengan frekuensi yang lain, tanggapan telegramnya dipancarkan.

Kedua sistem mempunyai keuntungan dan kerugian dalam kaitannya dengan prinsip yang berbeda tersebut. Jika satu menyimpan informasi tentang arah, jarak dan tingginya target dengan radar primer, kemudian radar pengawasan yang sekunder masih menyediakan informasi tambahan, identifikasi sinyal dan juga ketinggian target.

Kooperasi target (*transponder*) diperlukan untuk menjangkau pengurangan drastis pada daya transmisi dalam hal cakupan maksimum yang sama. Karena daya transmisi mempengaruhi persamaan radar pada radar primer dengan dua arah, pada radar pengawasan sekunder hanya satu arah. Persamaan perbedaan antara daya radar primer dengan sekunder dinyatakan dengan persamaan berikut :

$$PPSR \sim \frac{1}{R^4} \dots\dots\dots (2.1)$$

$$PSSR \sim \frac{1}{R^2} \dots\dots\dots (2.2)$$

Keterangan : P = Power

R = Jarak antara radar dengan objek

Suatu faktor > 1000 dapat diasumsikan sebagai nilai pemandu. Dalam hal ini pemancar akan semakin sederhana, semakin kecil dan semakin murah. *Receiver* menjadi semakin tidak sensitif, karena daya aktif lebih tinggi dibanding daya gema pasif.

Karena frekuensi pemancaran dan frekuensi penerimanya berbeda, maka tidak ada gangguan yang muncul. Pada sisi lain mustahil terjadi suatu perubahan frekuensi karena adanya gangguan. Gangguan khusus pada peralatan radar sekunder membuat diperlukannya pengukuran kawat tambahan

2.3 Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) atau yang biasa disebut *Geographical Information System* (GIS) pertama kali ditemukan pada tahun 1960 yang bertujuan untuk menyelesaikan dan menganalisa permasalahan geografis. Beberapa tahun kemudian perkembangan SIG berkembang tidak hanya seputar permasalahan geografi saja tetapi sudah merambah ke berbagai bidang seperti :

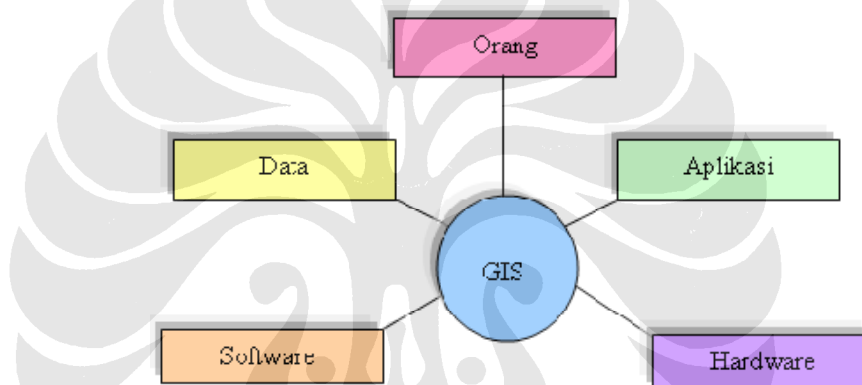
- navigasi dan *vehicle routing* (lintasan terpendek)
- peneliti: *spatial data exploration*
- utilitas (listrik, PAM, telpon) *inventory and management*
- pertahanan (*military simulation*), dll

2.3.1 Definisi SIG

SIG (Sistem Informasi Geografi) merupakan sistem informasi berbasis komputer yang menggabungkan antara unsur peta (geografis) dan informasinya tentang peta tersebut (data atribut) yang dirancang untuk mendapatkan, mengolah, memanipulasi, menganalisis, memperagakan dan menampilkan data spasial untuk menyelesaikan perencanaan, mengolah dan meneliti permasalahan. Dengan definisi ini, maka terlihat bahwa aplikasi SIG dilapangan cukup luas terutama untuk bidang yang memerlukan adanya suatu sistem informasi tidak hanya

menyimpan, menampilkan, dan menganalisa data atribut saja tetapi juga unsur geografisnya.

Ada beragam definisi dari para pakar mengenai SIG tersebut, intinya SIG adalah sebuah sistem untuk pengelolaan, penyimpanan, pemrosesan, analisis dan penayangan (*display*) data yang terkait dengan objek yang akan ditampilkan, salah satunya adalah permukaan bumi. Sistem tersebut untuk dapat beroperasi membutuhkan perangkat keras (*hardware*) dan perangkat lunak (*software*) juga manusia yang mengoperasikannya (*brainware*). Secara rinci SIG tersebut dapat beroperasi membutuhkan komponen-komponen sebagai berikut :



Gambar 2.4 Komponen-komponen SIG

Orang yang menjalankan sistem, dapat mengoperasikan, mengembangkan bahkan memperoleh manfaat dari sistem. Kategori orang yang menjadi bagian dari SIG ini beragam, misalnya operator, analis, programmer, administrator *database* bahkan *stakeholder*.

Aplikasi merupakan kumpulan dari prosedur-prosedur yang digunakan untuk mengolah data menjadi informasi. Misalnya penjumlahan, klasifikasi, rotasi, koreksi geometri, dan sebagainya.

Data yang digunakan dalam SIG dapat berupa data grafis dan data atribut. Data grafis/spasial ini merupakan data yang merupakan representasi fenomena permukaan bumi yang memiliki referensi (koordinat) lazim berupa peta, foto

udara, citra satelit dan sebagainya atau hasil dari interpretasi data-data tersebut. Sedangkan data atribut misalnya data sensus penduduk, catatan survei, data statistik lainnya. Kumpulan data-data dalam jumlah besar dapat disusun menjadi sebuah basisdata. Jadi dalam SIG juga dikenal adanya *database* yang lazim disebut sebagai *database* spasial.

Perangkat lunak SIG adalah program komputer yang dibuat khusus dan memiliki kemampuan pengelolaan, penyimpanan, pemrosesan, analisis dan penayangan data spasial. Ada pun merk perangkat lunak ini cukup beragam, misalnya Arc/Info, ArcView, ArcGIS, Map Info, TNT Mips (MacOS, Windows, Unix, Linux tersedia), GRASS, bahkan ada Knoppix GIS dan masih banyak lagi.

Perangkat keras ini berupa seperangkat komputer yang dapat mendukung pengoperasian perangkat lunak yang dipergunakan. Dalam perangkat keras ini juga termasuk didalamnya *scanner*, *digitizer*, *GPS*, *printer* dan *plotter*.

2.3.2 Sub Sitem Utama SIG

SIG terdiri dari empat subsistem utama :

1. **Sub-sistem Masukan**, perangkat untuk menyediakan data sampai siap dimanfaatkan oleh pengguna yang berupa peralatan pemetaan terestris, fotogrametri, digitasi, *scanner*, dsb. Pada umumnya output dari perangkat tersebut berupa peta, citra dan tayangan gambar lainnya.
2. **Sub-sistem Database**, digitasi peta dasar pada berbagai wilayah/daerah cakupan dengan berbagai skala telah dan terus dilakukan dalam rangka membangun sistem *database* spasial yang mudah diperbaharui dan digunakan dengan data literal sebagai komponen utamanya.
3. **Sub-sistem Pengolahan Data**, pengolahan data baik yang berupa vektor maupun raster dapat dilakukan dengan berbagai software seperti AUTOCAD, ARC/INFO, ERDAS, MAPINFO, ILWIS. Untuk metode vektor biasanya disebut digitasi sedangkan raster dikenal dengan metode overlay. Salah satu karakteristik software SIG adalah adanya sistem Layer (pelapisan) dalam

menggabungkan beberapa unsur informasi (penduduk, tempat tinggal, jalan, persil tanah, dll). Seperti: Layer, Coverage (ArcInfo produk ESRI), *Theme* (ArcView produk ESRI), Layer (AutoCAD Map produk Autodesk), *Table* (MapInfo produk MapInfo Corp.), dan lain-lainya.

4. **Sub-sistem Penyajian Informasi**, Dilakukan dengan berbagai media agar mudah dimanfaatkan oleh pengguna.

2.4 Sistem Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung. Informasi dan data bergerak melalui kabel-kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama sama menggunakan *hardware/software* yang terhubung dengan jaringan. Tiap komputer, printer atau periferal yang terhubung dengan jaringan disebut **node**. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node.

Sebuah jaringan biasanya terdiri dari 2 atau lebih komputer yang saling berhubungan diantara satu dengan yang lain, dan saling berbagi sumber daya misalnya CDROM, Printer, pertukaran file, atau memungkinkan untuk saling berkomunikasi secara elektronik. Komputer yang terhubung tersebut, dimungkinkan berhubungan dengan media kabel, saluran telepon, gelombang radio, satelit, atau sinar infra merah.

2.4.1 Jenis Jaringan Komputer Yang Digunakan

Jenis jaringan komputer yang digunakan adalah LAN (*Local Area Connection*). LAN adalah jaringan yang dibatasi oleh *area* yang relatif kecil, umumnya dibatasi oleh *area* lingkungan seperti sebuah perkantoran di sebuah gedung atau sebuah sekolah dan biasanya tidak jauh dari sekitar 1 km persegi.

Beberapa model konfigurasi LAN, satu komputer biasanya di jadikan sebuah *file server*. Yang mana digunakan untuk menyimpan perangkat lunak (*software*) yang mengatur aktifitas jaringan, ataupun sebagai perangkat lunak

yang dapat digunakan oleh komputer-komputer yang terhubung ke dalam jaringan. Komputer-komputer yang terhubung ke dalam jaringan (*network*) itu biasanya disebut dengan *workstation*. Biasanya kemampuan *workstation* lebih di bawah dari *file server* dan mempunyai aplikasi lain di dalam *harddisknya* selain aplikasi untuk jaringan. Kebanyakan LAN menggunakan media kabel untuk menghubungkan antara satu komputer dengan komputer lainnya.

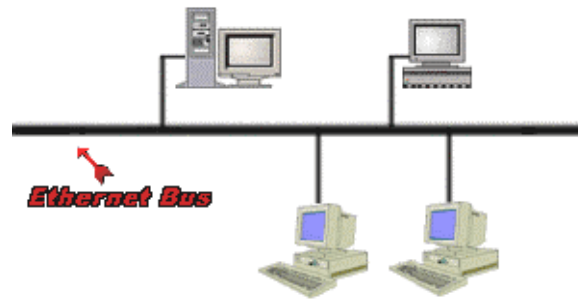
2.4.2 Protokol

Protokol adalah aturan-aturan main yang mengatur komunikasi diantara beberapa komputer di dalam sebuah jaringan, aturan itu termasuk di dalamnya petunjuk yang berlaku bagi cara-cara atau metode mengakses sebuah jaringan, topologi fisik, tipe-tipe kabel dan kecepatan transfer data. Protokol yang digunakan pada rancang bangun simulasi pengolahan data radar ini menggunakan Ethernet.

Ethernet

Protokol Ethernet sejauh ini adalah yang paling banyak digunakan, Ethernet menggunakan metode akses yang disebut CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*). Sistem ini menjelaskan bahwa setiap komputer memperhatikan ke dalam kabel dari jaringan sebelum mengirimkan sesuatu ke dalamnya. Jika dalam jaringan tidak ada aktifitas atau bersih, komputer akan mentransmisikan data, jika ada transmisi lain di dalam kabel, komputer akan menunggu dan akan mencoba kembali transmisi jika jaringan telah bersih. Kadangkala dua buah komputer melakukan transmisi pada saat yang sama, ketika hal ini terjadi, masing-masing komputer akan mundur dan akan menunggu kesempatan secara acak untuk mentransmisikan data kembali, metode ini dikenal dengan koalisi, dan tidak akan berpengaruh pada kecepatan transmisi dari jaringan.

Protokol Ethernet dapat digunakan pada model jaringan garis lurus, bintang, atau pohon. Data dapat ditransmisikan melalui kabel *twisted pair*, koaksial, ataupun kabel serat optik pada kecepatan 10 Mbps. Berikut pada gambar 2.5 adalah gambaran mengenai penggunaan protokol Ethernet.



Gambar 2.5 Penggunaan protokol ethernet

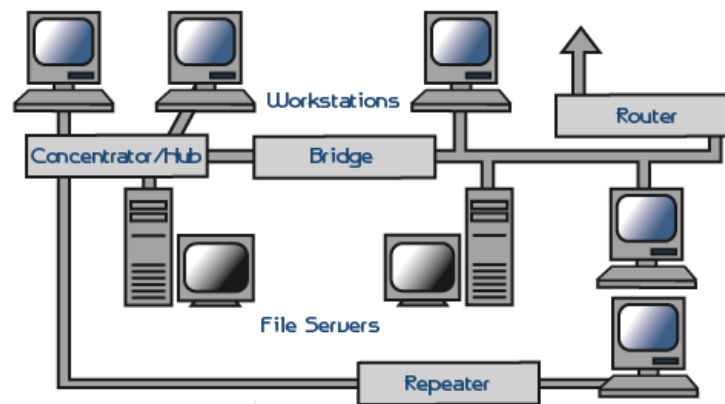
Berikut adalah sebuah tabel yang menjelaskan perbedaan diantara beberapa protokol yang ada dan melatar belakangi penggunaan protokol Ethernet dalam rancang bangun simulasi pengolahan data radar ini.

Tabel 2 Perbedaan setiap protokol dengan karakteristiknya

Protokol	Kabel Yang Digunakan	Kecepatan Transfer	Topology Fisik
Ethernet	<i>Twisted Pair, Coaxial, Fiber</i>	10 Mbps	<i>Linear Bus, Star, Tree</i>
Fast Ethernet	<i>Twisted Pair, Fiber</i>	100 Mbps	<i>Star</i>
Local Talk	<i>Twisted Pair</i>	0.23 Mbps	<i>Linear Bus or Star</i>
Token Ring	<i>Twisted Pair</i>	4-16 Mbps	<i>Star-Wired Ring</i>
FDDI	<i>Fiber</i>	100 Mbps	<i>Dual ring</i>
ATM	<i>Twisted Pair, Fiber</i>	155-2488 Mbps	<i>Linear Bus, Star, Tree</i>

2.4.3 Hardware yang Dibutuhkan

Perangkat keras yang dibutuhkan untuk membangun sebuah jaringan komputer untuk rancang bangun simulasi pengolahan data radar, yaitu *hub/switch*, dan segala sesuatu yang berhubungan dengan koneksi jaringan dan yang dibutuhkan untuk proses transformasi data didalam jaringan. Berikut pada gambar 2.6 adalah gambaran perangkat pada jaringan computer.



Gambar 2.6 Perangkat pada jaringan komputer

Perangkat-perangkat pada jaringan terdiri dari beberapa perangkat keras. Perangkat yang dipergunakan diantaranya adalah :

1. *File Servers*

Sebuah *file server* merupakan jantungnya sebuah jaringan. *File server* ini merupakan komputer yang umumnya memiliki spesifikasi sangat cepat, mempunyai memori yang besar, kapasitas *harddisk* yang besar, dan kartu jaringan yang cepat. Sistem operasi jaringan tersimpan disini, juga termasuk didalamnya beberapa aplikasi dan data yang dibutuhkan untuk jaringan.

Sebuah *file server* bertugas mengontrol komunikasi dan informasi diantara node/komponen dalam suatu jaringan. Sebagai contoh mengelola pengiriman file *database* atau pengolah kata dari *workstation* atau salah satu node, ke node yang lain, atau menerima *email* pada saat yang bersamaan dengan tugas yang lain, terlihat bahwa tugas *file server* sangat kompleks, dia juga harus menyimpan informasi dan membaginya secara cepat. Sehingga minimal sebuah *file server* mempunyai beberapa karakter seperti tersebut di bawah ini :

- *Processor* minimal 166 megahertz atau *processor* yang lebih cepat lagi (*Pentium Pro, Pentium II, PowerPC*).
- Sebuah *Harddisk* yang cepat dan berkapasitas besar atau kurang lebih 10 GB
- Sebuah RAID (*Redundant Array of Inexpensive Disks*).
- Sebuah tape untuk *back up* data (contohnya : DAT, JAZ, Zip, atau CDRW)
- Mempunyai banyak *port network*

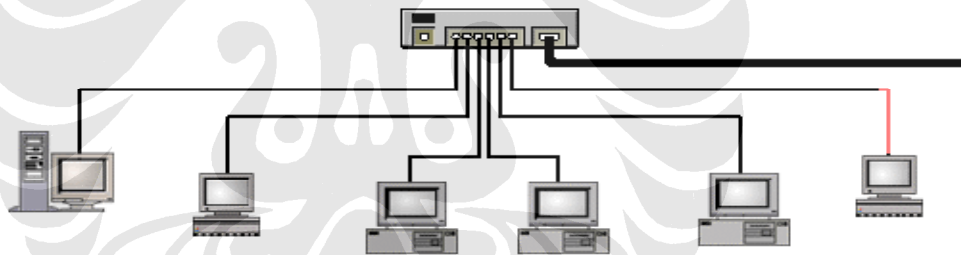
- Kartu jaringan yang cepat dan Reliabilitas
- Kurang lebih 32 MB memori

2. Workstations

Keseluruhan komputer yang terhubung ke *file server* dalam jaringan disebut sebagai *workstation*. Sebuah *workstation* minimal mempunyai kartu jaringan, aplikasi jaringan (*software* jaringan), kabel untuk menghubungkan ke jaringan, biasanya sebuah *workstation* tidak begitu membutuhkan *floppy disk* karena data yang ingin di simpan dapat diletakkan di *file server*.

3. Concentrators/Hubs

Sebuah Konsentrator/Hub adalah sebuah perangkat yang menyatukan kabel-kabel *network* dari tiap-tiap *workstation*, *server* atau perangkat lain. Dalam topologi bintang, kabel *twisted pair* datang dari sebuah *workstation* masuk kedalam hub. Pada gambar 2.7 adalah gambar perangkat *hub/switch*.



Gambar 2.7 Perangkat *Hub/Switch*

Hub mempunyai banyak *slot concentrator* yang mana dapat dipasang menurut nomor port dari card yang dituju

Ciri-ciri yang dimiliki Konsentrator (hub) adalah :

- Biasanya terdiri dari 8, 12, atau 24 port RJ-45
- Digunakan pada topologi bintang/Star
- Biasanya di jual dengan aplikasi khusus yaitu aplikasi yang mengatur manajemen port tersebut.
- Biasanya di pasang pada rak khusus, yang didalamnya ada Bridges, router

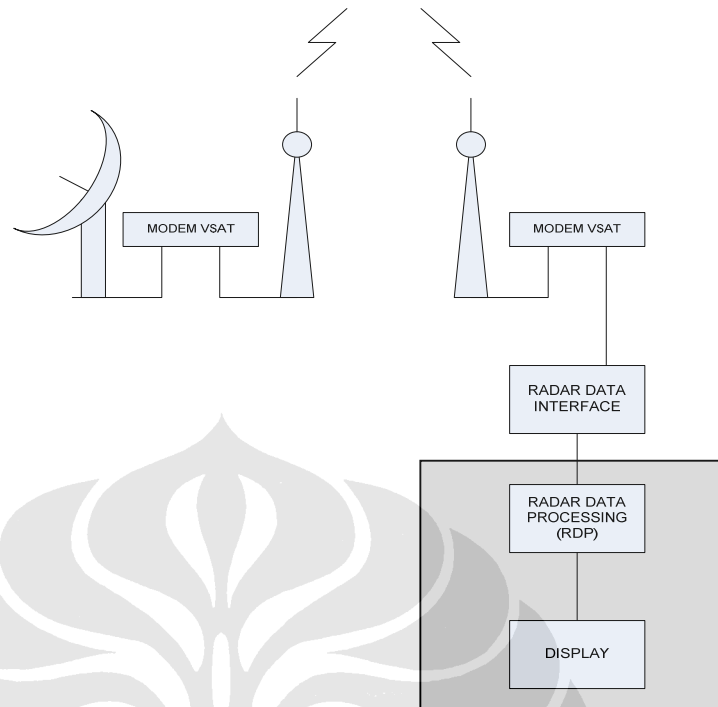
BAB III

PERANCANGAN SISTEM MONITORING RADAR MENGUNAKAN MAP SERVER

Pada bab ini akan dibahas mengenai perancangan pada sistem monitoring data radar dengan pemetaan menggunakan *Mapserver* untuk aplikasi monitoring lalu lintas udara. Perancangan ini merupakan perancangan untuk mengembangkan kondisi awal yang hanya menggunakan *PC desktop* untuk dapat memonitoring sistem lalu lintas udara, menjadi kondisi menggunakan jaringan komputer sehingga dimanapun keberadaannya selama terhubung dengan *server* pada jaringan komputer yang melayani aplikasi monitoring data radar tersebut maka akan mampu untuk mengakses atau memonitoring sistem lalu lintas udara.

3.1. Perancangan Sistem Radar Dalam Pengembangan

Perancangan ini dibuat untuk mengembangkan kondisi sesungguhnya, namun fokus pembahasan pada perancangan ini tidak membahas secara keseluruhan, melainkan pembahasan dibatasi dengan sistem *display* menggunakan *Mapserver*. Gambar 3.1 adalah sebuah skema bentuk perancangan radar yang perancangannya difokuskan pada sistem *display*.



Gambar 3.1 Skema pembahasan perancangan radar

3.2. Spesifikasi Radar Dalam Perencanaan

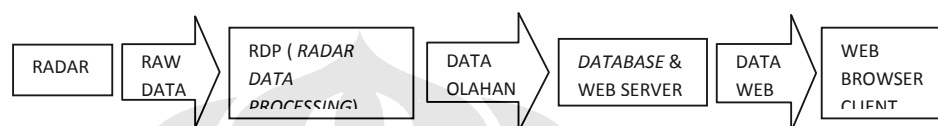
Radar yang digunakan dalam simulasi ini menggunakan radar jenis PR800. Radar ini memiliki spesifikasi yang memiliki format data yang mudah untuk dijadikan simulasi monitoring data radar dengan menggunakan pemetaan dengan *Map Server*.

Antar muka dan format data antara *Processor* Radar PR800 dan pengolahan data pada pusat pengolahan di bandara di definisikan dengan sebuah dokumen kontrol. Berikut adalah spesifikasi radar yang digunakan dalam perencanaan ini :

- a) *Output electric* dan standar fisik akan sesuai dengan spesifikasi CCITT V-24 (RS-232C). *Output* merupakan *one-way*, data yang sinkron, dengan hanya sebuah *clock* sinyal.
- b) Untuk *baud rate* pada *output* operasionalnya adalah 2400 atau 4800 *baud rate*.
- c) Untuk protokol komunikasinya menggunakan *half-duplex*, dengan ukuran *message/block* variabel, berorientasi pada karakter, dan bertipe BSC.

3.3. Perancangan Sistem *Hardware* dan *Software*

Pembuatan aplikasi monitoring ini menggunakan beberapa komponen perangkat keras dan perangkat lunak untuk mendukung proses alur data yang diinginkan. Pada gambar 3.2 adalah gambaran mengenai alur data secara garis besar.



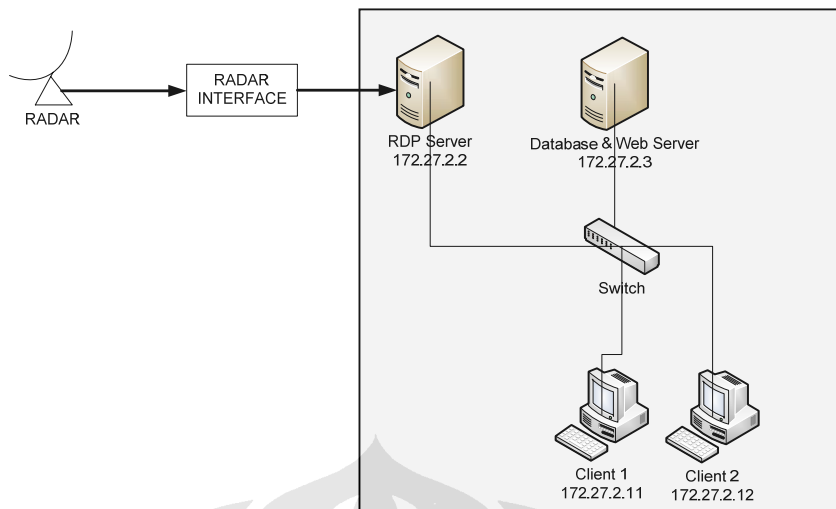
Gambar 3.2 Alur data aplikasi monitoring data Radar

Karena perancangan ini dispesifikan pada sistem *display*, maka pembahasan perancangan difokuskan pada sistem *display* yang menggunakan pemetaan dengan *Mapserver*.

Dalam keterangannya secara umum radar akan mengirimkan *raw data* ke RDP. Di RDP ini data diolah dan dihubungkan dengan sistem *database* yang ada pada *server* (dalam hal ini menggunakan *Mapserver*). *Database* ini diolah dan akan ditampilkan pada *server*. Dengan pengolahan sistem *database* pada *server* maka *client-client* yang terhubung dengan *server* akan dapat melihat tampilan *display* yang ditampilkan oleh *server*.

3.3.1. Aplikasi Kondisi *Real*

Perancangan ini dibuat pada kondisi sesungguhnya dan sistem *display* nya dirancang dengan menggunakan *server* yang bisa diakses oleh *client* yang terhubung *server*. Untuk lebih jelasnya dapat dilihat pada gambar 3.3.



Gambar 3.3 Sistem monitoring data radar kondisi *real*

Pembahasan disini hanya proses pada sistem *server* dan penyediaan layanan berupa tampilan data radar pada sistem pemetaan *mapserver*. Pada gambar 3.3 ada beberapa komponen pendukung dalam rancangan ini, diantaranya :

1) *RDP server*

Komponen ini yang akan menterjemahkan data mentah menjadi data olahan setengah matang yang nantinya akan dikirim pada *database server*. Komponen ini di *setting* menggunakan sistem operasi Linux Mandrake yang memiliki kelebihan dalam kemampuan pengelolaan program bahasa C maupun C++.

2) *Database server*

Komponen ini yang akan mengelola data masukan dari *RDP server* menjadi data yang benar-benar matang dan menghimpunnya dalam sebuah *database*. Data-data yang masuk ke *database* inilah yang akan ditampilkan melalui *web server*. Komponen ini menggunakan sistem operasi Microsoft Windows, sebagai sistem operasi *master/pusat*.

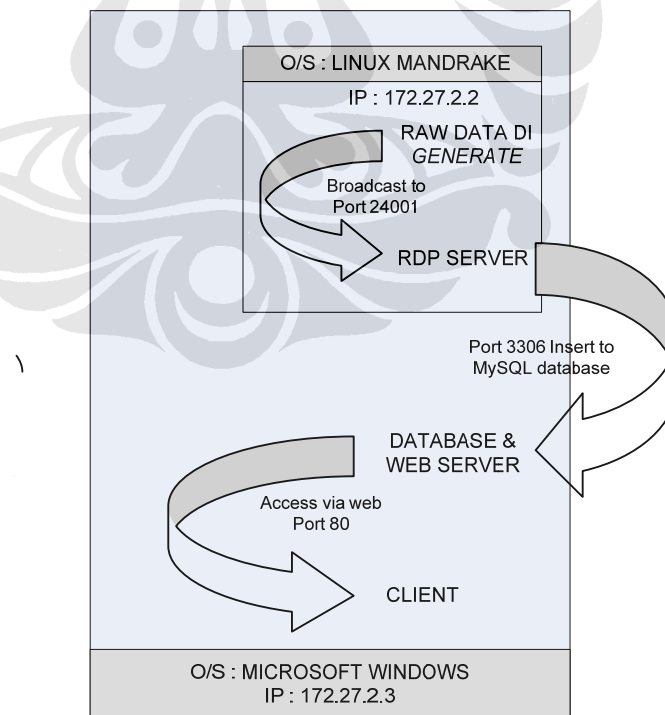
3) *Client*

Komponen ini bertujuan untuk mengakses layanan monitoring data radar melalui *server*. Komponen ini menggunakan sistem operasi Microsoft Windows untuk mempermudah proses *browsing* data radar melalui *web server*.

Pada prakteknya perancangan ini dibutuhkan tiga komponen penting, yaitu *RDP server*, *Database server*, serta *client*. Namun dalam perancangan ini untuk mempermudah pengelolaan data serta aplikasi secara langsung, maka hanya perlu membutuhkan satu komponen saja. Untuk merekayasa penggunaan tiga komponen penting tersebut menjadi satu komponen, maka dibuatlah sebuah simulasi yang mewakili tiga komponen penting tersebut.

3.3.2. Aplikasi Kondisi Simulasi

Perancangan dengan kondisi simulasi ini merupakan sebuah rekayasa dari perancangan dalam kondisi sesungguhnya. Namun rekayasa ini tidak mengurangi sifat dari masing-masing komponen penting yang digunakan pada perancangan dalam kondisi sesungguhnya. Tiga komponen penting yang ada akan dibuat dalam satu komponen yang dapat berfungsi sebagai tiga komponen sebelumnya. Untuk itu perlu di *setting* karakteristik dan spesifikasi komponen yang digunakan. Untuk lebih jelasnya dapat dilihat pada gambar 3.4 mengenai bagan penggunaan tiga komponen dalam satu komponen.



Gambar 3.4 Sistem monitoring data Radar dalam kondisi simulasi

Dalam satu komponen komputer dapat difungsikan menjadi tiga sistem kerja, diantaranya :

3.3.2.1. Komponen Sebagai *RDP Server*

Komponen ini menggunakan sistem operasi Linux, karena sistem operasi ini cukup mudah dalam mengolah program bahasa C dan C++, yang merupakan bahasa program yang digunakan pada *RDP Server* dalam mengolah *raw data*. Untuk memanipulasi agar dalam satu komponen komputer mampu memiliki lebih dari satu sistem operasi adalah dengan memanfaatkan perangkat lunak *virtual machine* (VMWare). Perangkat lunak inilah yang akan menciptakan sebuah windows baru yang mampu memiliki sistem operasi yang berbeda dari sistem operasi yang digunakan. Dalam hal ini sistem operasi Linux berada dalam VMWare, sedangkan sistem operasi yang digunakan pada komputer adalah Microsoft Windows.

Perangkat lunak yang digunakan pada komputer *RDP server* ini digunakan untuk mengolah data radar mentah dari *radar interface* dan dimatangkan kembali menjadi data-data matang yang lebih spesifik. Ada dua proses pada pengolahan data radar pada *RDP server*, yaitu membangkitkan *raw data* dari sumber data *radar interface* dan menspesifikkan data radar menjadi data matang yang lebih spesifik untuk ditampilkan.

a) *Proses membangkitkan raw data dari sumber data radar interface*

Ini merupakan proses pertama dari *RDP server*, dimana *raw data* yang diperoleh dari sumber data yang disimulasikan dengan file *cgk.dat* dari *radar interface*, dibangkitkan menjadi data setengah matang untuk diproses kembali menjadi data matang. Proses ini membutuhkan bantuan program untuk membangkitkan *raw data* yang bersumber dari *radar interface* dengan program bahasa C++ (nama filenya “*radar_data_cgk.cpp*”). File “*radar_data_cgk.cpp*” ini akan di *compile* sehingga menghasilkan file *executable* “*radar_data_cgk*”. File inilah yang akan dieksekusi sehingga *raw data* yang dibangkitkan akan ditampilkan seperti pada gambar 3.5.

```

Mandrake10 - VMware Workstation
File Edit View VM Power Snapshot Windows Help
Snapshot Revert
Home Mandrake10
total 4372
-rwxr-xr-x 1 root root 4400693 Dec  2 15:23 cgk.dat*
-rwxr-xr-x 1 root root      90 Dec  2 15:23 coba*
-rwxr-xr-x 1 root root  20514 Dec  2 15:23 data_radar_cgk*
-rwxr-xr-x 1 root root   3459 Dec  2 15:23 data_radar_cgk.cpp*
-rwxr-xr-x 1 root root  16532 Dec  2 15:23 RDP2MySQL*
-rwxr-xr-x 1 root root   5490 Dec  2 15:23 RDP2MySQL.c*
[root@localhost okabnyobal]# ./da
data radar_cgk      data_radar_cgk.cpp
[root@localhost okabnyobal]# ./data_radar_cgk
[ ff 14 82 70 40 c2 fd 90 0 10 e2 3 31 25 19 a1 0 0 3 41 ]
[ ff 14 82 70 40 d3 fb 48 2 1f e3 12 31 99 19 a0 0 0 3 89 ]
[ ff 14 82 70 1 13 f3 60 8 96 25 2 33 50 1e 5b 0 0 4 15 ]
[ ff 14 82 60 11 e0 f0 31 c 20 d5 9 32 60 19 a3 0 0 4 92 ]
[ ff 14 82 70 5 31 fb 40 8 dc f5 29 32 81 19 b9 0 0 4 78 ]
[ ff 14 82 70 41 a8 fc 9d 6 86 22 fa 33 30 1e 60 0 0 3 88 ]
[ ff 14 82 70 1 3e fc cb b 22 d3 9 33 40 1c e9 0 0 5 19 ]
[ ff 14 83 70 1 55 2 50 1f 17 4 0 10 0 0 0 0 0 0 40 ]
[ ff 14 82 70 40 9c 1 8f 6 1a 28 0 32 90 15 47 0 0 4 49 ]
[ ff 14 82 70 41 26 1 cb 5 97 28 f7 33 10 15 2f 0 0 4 67 ]
[ ff 14 82 70 1 29 b 3f f 5f 1e e4 33 50 10 78 0 0 4 62 ]
[ ff 14 82 70 1 f1 10 a3 10 b2 13 22 33 30 19 bf 0 0 4 36 ]
[ ff 14 82 70 1 d1 9 59 6 99 df c 31 85 1c ec 0 0 3 99 ]
[ ff 14 82 70 1 8a 10 25 8 bf e4 1e 32 24 1e 7 0 0 4 61 ]
[ ff 14 82 70 40 d3 fb 3b 2 26 e2 12 32 0 19 a0 0 0 3 90 ]
[ ff 14 82 70 1 13 f3 6e 8 91 25 f9 33 50 1e 5b 0 0 4 21 ]
[ ff 14 82 50 21 e0 f0 20 c 24 d5 9 32 60 19 a3 0 0 4 92 ]
[ ff 14 82 70 1 31 fb 2f 8 e1 e6 20 32 81 19 b9 0 0 4 62 ]
[ ff 14 82 70 41 a8 fc ac 6 85 23 fb 33 30 1e 60 0 0 3 95 ]
[ ff 14 82 70 1 3e fc bc b 23 d5 3 33 40 1c e9 0 0 4 84 ]

```

Gambar 3.5 Tampilan raw data dari RDP Server

- b) Proses menspesifikan data radar menjadi data matang yang lebih spesifik untuk ditampilkan

Ini merupakan proses kedua dari RDP server, dimana data radar sudah setengah matang sehingga perlu penterjemahan kembali agar data radar benar-benar matang dan lebih dispesifikan sehingga dapat ditampilkan kedalam sistem pemetaan melalui *mapserver*. Proses ini akan terus berlanjut hingga data yang diterjemahkan menjadi data matang dikirim ke *database server*. Proses ini membutuhkan sebuah program bahasa C yang mampu menterjemahkan data radar setengah matang tersebut dan proses pengiriman data matang radar ke

xxxviii

database server. File tersebut adalah “RDP2MySQL.c”. File ini akan di *compile* dan akan menghasilkan file *executable* yaitu “RDP2MySQL”. File inilah yang akan dieksekusi sehingga akan menampilkan tampilan data radar matang yang lebih spesifik seperti yang tergambar pada gambar 3.6.

```

Mandrake10 - VMware Workstation
Home Mandrake10
-----
TrackNumber : 497
posisi X : 248168
posisi Y : 246316
Flight Level : 330
SSR code : 4677
INSERT/UPDATE TrackNumber:497 succeeded: 1 row affected
-----
TrackNumber : 465
posisi X : 135196
posisi Y : 96304
Flight Level : 185
SSR code : 6354
INSERT/UPDATE TrackNumber:465 succeeded: 1 row affected
-----

```

Gambar 3.6 Tampilan data matang yang lebih spesifik dari *RDP Server*

3.3.2.2. Komponen Sebagai *Server*

Komponen komputer sebagai *server* ini menggunakan sistem operasi windows. Komponen ini menggunakan perangkat lunak untuk *server* dan *database server*. Hal ini bertujuan untuk mengolah data-data yang masuk dari filterisasi data ke dalam *database server*. *Database server* ini akan menghimpun data-data tersebut dan selanjutnya akan ditampilkan ke dalam sebuah *server* yang pengolahannya dilakukan oleh *Mapserver*. Perangkat lunak yang digunakan merupakan perangkat lunak *appserv* yang mengkombinasikan antara apache sebagai *web server*nya, MySQL sebagai *database server*nya, dan *script* penterjemah Php.

Proses ini membutuhkan sebuah *browser* untuk mengakses sistem *database* pada *appserv*. Terlebih dahulu dibuat sistem *databasenya* dengan fieldnya berjumlah lima (sesuai dengan data yang ingin ditampilkan), diantaranya *Track Number*, Posisi X (Pos X), Posisi Y (Pos Y), *Flight Level* (FL), dan *SSRCode*. Gambar 3.7 adalah gambaran dari sistem *database* yang telah dibuat dan data radar matang siap dikirim ke sistem *database* ini.



Server: localhost Database: dbmap Tabel: tdata

Struktur Browse SQL Cari Sisipkan Ekspor Operasi Hapus se

Field	Jenis	Pembandingan	Atribut Kosong	Default	Ekstra	Aksi
<input type="checkbox"/> TrackNumber	varchar(6)	latin1_swedish_ci	Tidak			
<input type="checkbox"/> PosX	varchar(10)	latin1_swedish_ci	Tidak			
<input type="checkbox"/> PosY	varchar(10)	latin1_swedish_ci	Tidak			
<input type="checkbox"/> FL	varchar(8)	latin1_swedish_ci	Tidak			
<input type="checkbox"/> SSRCode	varchar(6)	latin1_swedish_ci	Tidak			

↑ Tandakan semua / Uncheck semua Dengan pilihan:

Gambar 3.7 Tampilan *database server* sebagai pengolah data

Adapun sistem pemetaan dibuat dengan menggunakan program *Mapserver*. Pemetaan ini menggunakan pulau Jawa Barat, DKI Jakarta dan Banten. Masing-masing propinsi dibuat *class* dan *stylenya*. Berikut adalah cuplikan program *Mapserver* untuk membentuk ketiga propinsi tersebut dalam satu pulau.

Untuk propinsi Jawa Barat

```
// objek class & style jabar (comment)
$objClassJbr = ms_newClassObj ($objLayerJbr);
$objClassJbr->Set ("name", "Jabar");
$objClassJbr->SetExpression ("3"); // kodeprop=1
$objClassJbr->Set("template",
    "c:/buku_mapserver/html/query25_prop.html");
$objStyleJbr = ms_newStyleObj ($objClassJbr);
$objStyleJbr->color->setRGB (240,252,258);
$objStyleJbr->outlinecolor->SetRGB (0,0,0);
// objek label kelas (comment)
$objClassJbr->label->Set ("font", "arial");
```



```

$objClassJbr->label->Set ("type", MS_TRUETYPE);
$objClassJbr->label->Set ("size", 8);
$objClassJbr->label->Set ("position", MS_CC);
$objClassJbr->label->color->SetRGB (0, 0, 254);

```

Untuk propinsi DKI Jakarta

```

// objek class & style DKI (comment)
$objClassDki = ms_newClassObj ($objLayerJbr);
$objClassDki->Set ("name", "DKI");
$objClassDki->SetExpression ("2"); // kodeprop=2
$objClassDki->Set("template",
    "c:/buku_mapserver/html/query25_prop.html");
$objStyleDki = ms_newStyleObj ($objClassDki);
$objStyleDki->color->setRGB (237, 177, 203);
$objStyleDki->outlinecolor->SetRGB (0,0,0);
// objek label kelas (comment)
$objClassDki->label->Set ("font", "arialbold");
$objClassDki->label->Set ("type", MS_TRUETYPE);
$objClassDki->label->Set ("size", 8);
$objClassDki->label->Set ("position", MS_CC);
$objClassDki->label->color->SetRGB (0, 0, 0);

```

Untuk propinsi Banten

```

// objek class & style Banten (comment)
$objClassBtn = ms_newClassObj ($objLayerJbr);
$objClassBtn->Set ("name", "Banten");
$objClassBtn->SetExpression ("1"); // kodeprop=1
$objClassBtn->Set ("template",
    "c:/buku_mapserver/html/query25_prop.html");
$objStyleBtn = ms_newStyleObj ($objClassBtn);
$objStyleBtn->color->setRGB (172, 247, 60);
$objStyleBtn->outlinecolor->SetRGB (0,0,0);
// objek label kelas (comment)
$objClassBtn->label->Set ("font", "comicsans");
$objClassBtn->label->Set ("type", MS_TRUETYPE);
$objClassBtn->label->Set ("size", 8);
$objClassBtn->label->Set ("position", MS_CC);
$objClassBtn->label->color->SetRGB (254, 32, 40);

```

Begitupun juga dengan sistem *plot* yang dilakukan pada peta. *Plot* ini adalah penandaan posisi objek yang terdeteksi oleh radar, sesuai dengan keberadaan objek tersebut ketika radar bekerja. *Plot* ini dibuat *layer* sendiri yang berisikan *class* dan *stylenya*. Karakteristik *plot* atau *point* dirancang melalui program *Mapserver*, berikut adalah rancangannya untuk karakteristik *point* atau *plot* :

```
// objek layer Point
$objLayerPoint[ ] = ms_newLayerObj($objMap);
$objLayerPoint[$i]->set ("name", "Point");
$objLayerPoint[$i]->set ("type", MS_LAYER_POINT);
$objLayerPoint[$i]->set ("status", MS_DEFAULT);

$objCoordList[ ] = ms_newLineObj();
$objPointShape[ ] = ms_newShapeObj(MS_SHAPE_POINT);

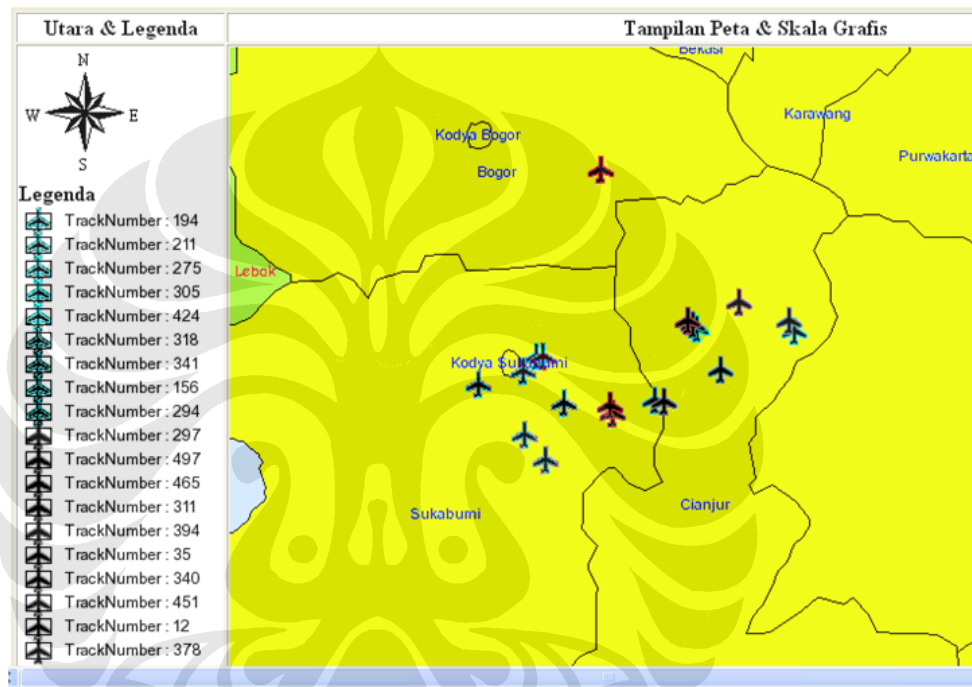
$posTargetX[ ] = $Koordinat_Radar_X+$posX[$i];
$posTargetY[ ] = $Koordinat_Radar_Y+$posY[$i];

$objCoordList[$i]->addXY($posTargetX[$i], $posTargetY[$i]);
$objPointShape[$i]->add($objCoordList[$i]);
$objLayerPoint[$i]->addFeature($objPointShape[$i]);

// objek class & style Point
$a = 10*$i;
$objClassPoint[ ] = ms_newClassObj ($objLayerPoint[$i]);
$objClassPoint[$i]->Set ("name", "TrackNumber : $TrackNumber[$i];
    Koordinat:($posTargetX[$i], $posTargetY[$i]);
    FlightLevel:$FL[$i]; SSR:$SSR[$i]");
$objStylePoint[ ] = ms_newStyleObj($objClassPoint[$i]);
$objStylePoint[$i]->color->setRGB(0,0,0);
$objStylePoint[$i]->outlinecolor->setRGB(2*$a, 255-2*$a, 255-2*$a);
$objStylePoint[$i]->set ("symbolname", "pesawat");
$objStylePoint[$i]->set( "size", "15");
```

3.3.2.3. Komponen Sebagai Client

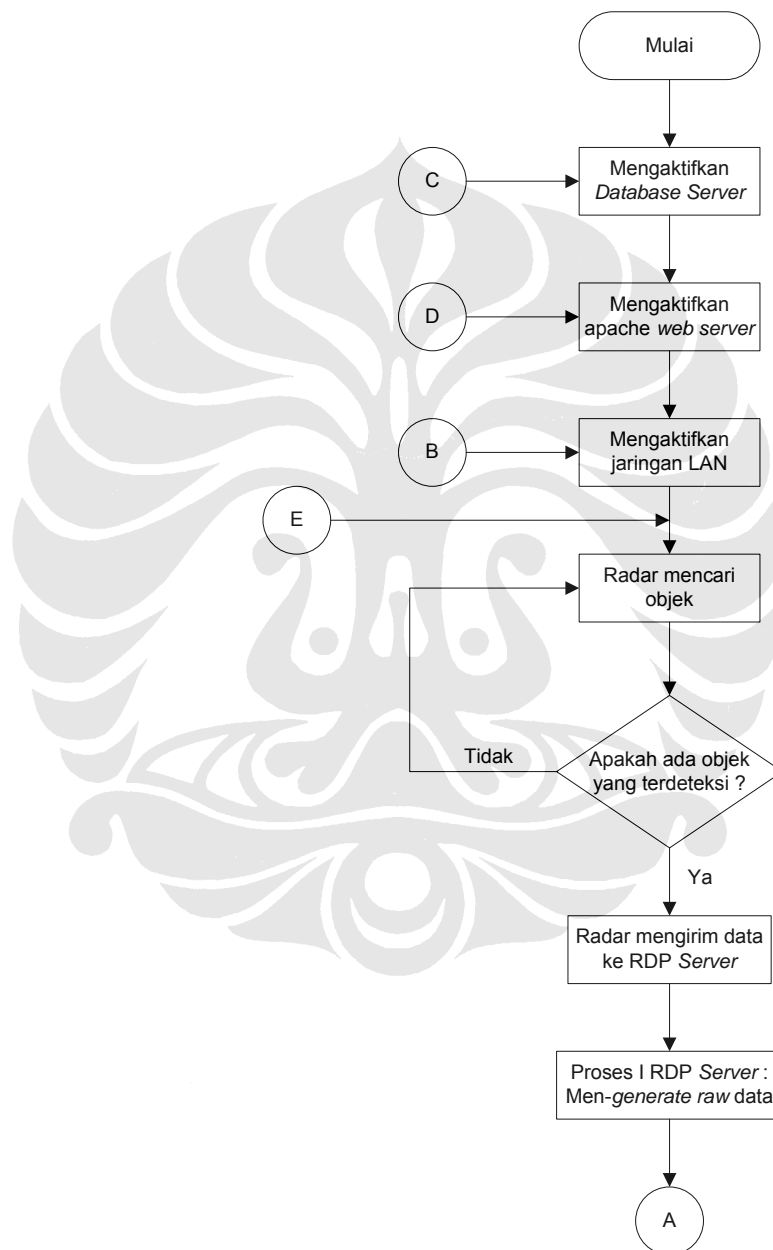
Komponen ini menggunakan sistem operasi windows. Ada beberapa persyaratan untuk dapat berfungsi menjadi *client*, yaitu terhubung dengan jaringan *server* dan memiliki *browser internet* yang nantinya akan digunakan untuk mengakses data radar yang dimonitoring melalui *server*. Gambar 3.8 dibawah ini merupakan gambaran dari sistem monitoring pada *client*.

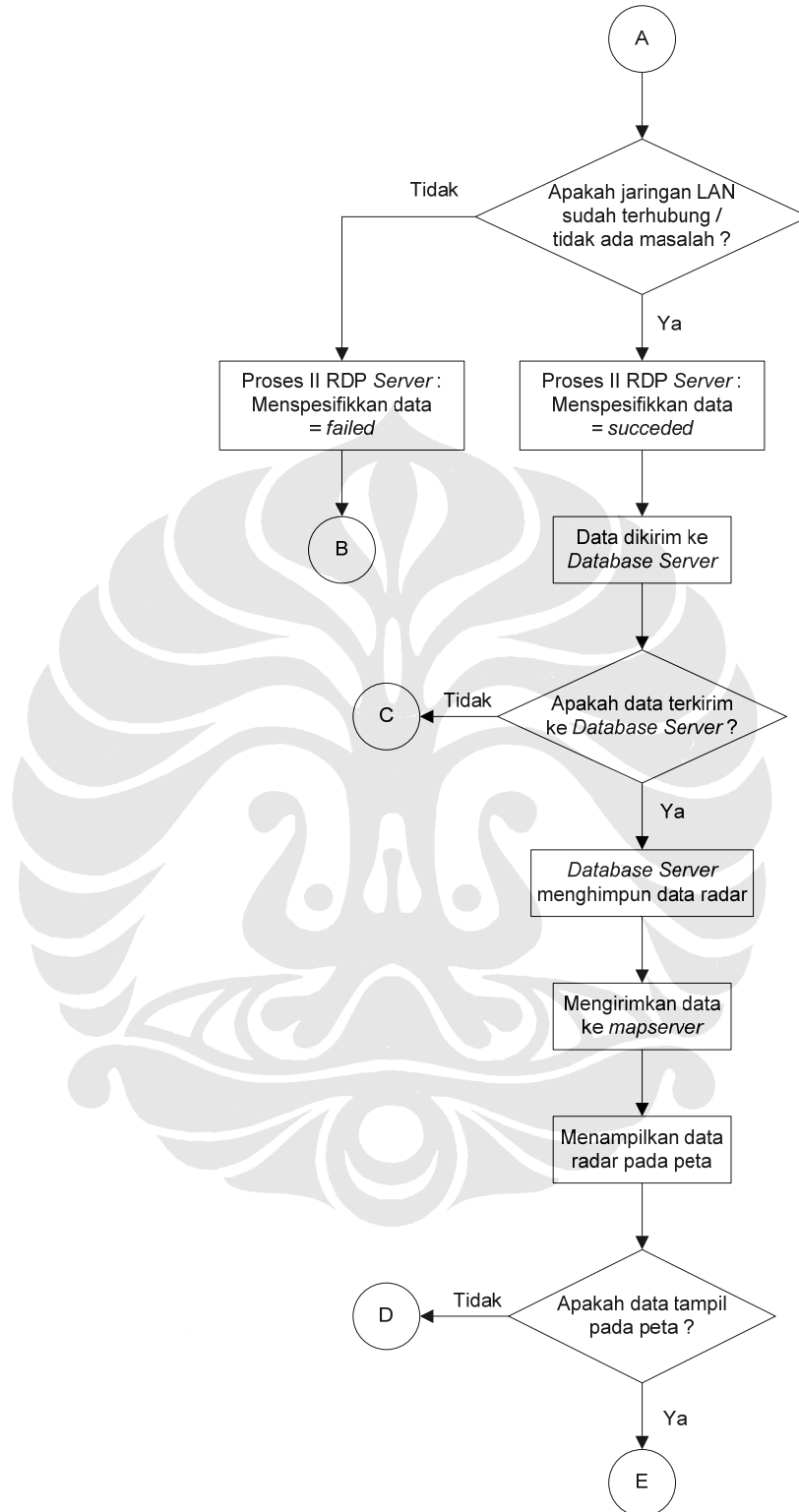


Gambar 3.8 Tampilan monitoring data radar pada *client*

3.4. Diagram Alir (*Flowchart*) Sistem Monitoring Data Radar Menggunakan *Mapserver*

Berikut adalah diagram alir (flowchart) yang menggambarkan aliran penggunaan dalam sistem monitoring data radar.





BAB IV ANALISIS PENGOLAHAN DATA RADAR

Pada bab ini akan dibahas mengenai pembahasan analisis data yang diperoleh. Data radar yang diperoleh adalah data yang sebenarnya sudah matang, namun masih belum dimengerti oleh *user*. Untuk itu bab ini akan membahas proses penterjemahan data radar.

Data radar ini diperoleh dari radar yang digunakan untuk memantau pesawat (objek) umum yang biasanya menggunakan radar sekunder. Penggunaan radar sekunder inilah yang menentukan format data radar apa yang akan digunakan. Data radar yang diperoleh dari lapangan ini berada pada sistem pemrosesan data (*RDP server*). Data sumber ini merupakan data berformat “*dat*”. Pada perencanaan ini disimulasikan dengan sebuah *file* yang merupakan rekaman dari data radar sesungguhnya dalam jangka waktu beberapa menit.

Sumber data radar “*cgk.dat*” ini lalu diolah dengan bantuan sebuah program yang dibuat menggunakan program bahasa C++. File program bahasa C++ ini disimpan dengan nama file “*data_radar_cgk.cpp*”. File inilah yang akan membangkitkan *raw* data menjadi data matang yang menampilkan data radar secara terus menerus sesuai dengan data yang masuk pada saat pengambilan data. Namun dalam pembahasan analisis ini tidak membahas proses keluarnya *raw* data. Pembahasan akan dilakukan proses penterjemahn data radar yang telah dibangkitkan menjadi sebuah data yang lebih spesifik. Data-data ini dispesifikkan lagi karena dalam skripsi ini hanya beberapa data saja yang akan ditampilkan. Data-data yang akan ditampilkan diantaranya :

1. *Track Number*,
2. Posisi X (Pos X),
3. Posisi Y (Pos Y),
4. *Flight Level (FL)*,
5. *SSR Code*.

4.1. Format Data Radar yang Dipergunakan

Untuk lebih memahami *raw data*, maka dibutuhkan sebuah format data yang dimiliki oleh radar tersebut. Karena data radar diperoleh dari radar sekunder maka yang digunakan adalah format data radar sekunder. Untuk lebih jelasnya format data sekunder akan ditampilkan dalam sebuah tabel berikut.

Tabel 4.1 Format *Secondary Message*

F		F						12 or 14							
P	F	0	0	0	IP	1/2	T	QUAL							
0	SA	Q ₁	Q ₀	RF	0	0	N ₈	N ₇	N ₆	N ₅	N ₄	N ₃	N ₂	N ₁	N ₀
X ₁₅							X ₈	X ₇	X ₀						
Y ₁₅							Y ₈	Y ₇	Y ₀						
DX							DY								
C	M ₂	M ₁	M ₀	Code 1											
C	M ₂	M ₁	M ₀	Code 2											
C	M ₂	M ₁	M ₀	Code 3											
SPEED MODULE															

Decoding Secondary Messages

- FF : *Secondary Message*
 14 : *Number of Message Byte (20, in decimal) presence of SPEED MODULE field*
 12 : *Number of Message Byte (18, in decimal) absence of SPEED MODULE field*
 P=0, F=1 : *Case of Civil or Military Distress*
 P=1, F=0 : *All Other Cases*
 IP=1 : *SPI*
 1/2 : *PR800 Number*
 T=1 : *Dummy Plot*
 QUAL : *Track Quality [max. value = 6E (110 in decimal)]*
 SA=1 : *Associated*
 Q₁, Q₀ : *Number of Misses*
 RF=1 : *Reflection*
 N₀ to N₆ : *Track Number*
 X₀ to X₁₅ : *X Position (X₀ = 1/32 nautical)*
 Y₀ to Y₁₅ : *Y Position (Y₀ = 1/32 nautical)*
 DX : *Defines the direction of speed vector (LSB – 1/32 nautical)*
 DY : *Defines the direction of speed vector (LSB – 1/32 nautical)*
 C=1 : *Garbled code or code message, where*

Tabel 4.2 Format Kode *Message*

M ₀	M ₁	M ₂	Indication
0	0	0	No Code
0	0	1	Mode A
0	1	0	Mode B
0	1	1	Mode C
1	0	0	Mode D
1	0	1	Mode 1
1	1	0	Mode 2
1	1	1	Note Used

Mode C adalah kode dalam BCD dengan 3 digit setiap 4 bits (*with the least significant bit (LSI) = 100 feet*). Untuk semua mode yang lain kodenya adalah : A4, A2, A1, B4, B2, B1, C4, C2, C1, D4, D2, D1.

SPEED MODULE : Module of Speed vector present if heading message : FF14

4.2. Pengujian *Sample Data*

Untuk lebih memudahkan dalam penterjemahan radar, maka perlu diambil sebuah contoh dari *raw* data untuk dijelaskan secara rinci. Pada awalnya *raw* data radar dibangkitkan, sehingga membentuk *raw* data berbaris. Bentuk *raw* data dapat dilihat pada gambar 4.1 berikut ini.

```
[root@localhost okabnyoba]# ./data_radar_cgk
[ ff 14 82 70 40 c2 fd 90 0 10 e2 3 31 25 19 a1 0 0 3 41 ]
[ ff 14 82 70 40 d3 fb 48 2 1f e3 12 31 99 19 a0 0 0 3 89 ]
[ ff 14 82 70 1 13 f3 60 8 96 25 2 33 50 1e 5b 0 0 4 15 ]
[ ff 14 82 60 11 e0 f0 31 c 20 d5 9 32 60 19 a3 0 0 4 92 ]
[ ff 14 82 70 5 31 fb 40 8 dc f5 29 32 81 19 b9 0 0 4 78 ]
[ ff 14 82 70 41 a8 fc 9d 6 86 22 fa 33 30 1e 60 0 0 3 88 ]
[ ff 14 82 70 1 3e fc cb b 22 d3 9 33 40 1c e9 0 0 5 19 ]
[ ff 14 83 70 1 55 2 50 1f 17 4 0 10 0 0 0 0 0 40 ]
[ ff 14 82 70 40 9c 1 8f 6 1a 28 0 32 90 15 47 0 0 4 49 ]
[ ff 14 82 70 41 26 1 cb 5 97 28 f7 33 10 15 2f 0 0 4 67 ]
[ ff 14 82 70 1 29 b 3f f 5f 1e e4 33 50 10 78 0 0 4 62 ]
[ ff 14 82 70 1 f1 10 a3 10 b2 13 22 33 30 19 bf 0 0 4 36 ]
```

Gambar 4.1 Raw data radar

dari raw data diatas diambil sebuah *sample* untuk dilakukan perhitungan.

Misalnya diambil sebuah *sample* pada data pertama, yaitu :

xlvi

[Ff 14 82 70 40 c2 fd 90 0 10 e2 3 31 25 19 a1 0 0 3 41]

Dari *sample* data diatas, informasi-informasi yang ada pada data tersebut dipisah satu sama lain dengan spasi. Sesuai format data radar yang digunakan yaitu format data sekunder, maka ada 20 *point* informasi yang terdapat pada sebuah data. *Point-point* informasi itu dapat dijelaskan pada penjelasan berikut ini :

1) **ff** = *secondary message*

Point ini maksudnya adalah sebuah data yang diperoleh dari radar sekunder

2) **14** = *Number of Message Byte (20, in decimal) presence of SPEE MODULE field*

Point ini maksudnya adalah jumlah informasi yang ditampilkan pada data ini adalah 20 (dalam desimal) dan menampilkan SPEED MODULE

3) **82** = P F 0 0 0 IP $\frac{1}{2}$ T

Dalam bentuk binernya adalah **82** = 1000 0010, jadi

- P = 1
 - F = 0
 - 0 = 0
 - 0 = 0
 - 0 = 0
 - 0 = 0
 - IP = 0
 - $\frac{1}{2}$ = 1
 - T = 0
- } artinya all other cases.
} maksudnya adalah semua jenis penerbangan kecuali militer
- artinya adalah menunjukkan penggunaan radar PR800 bukan "*dummy plot*" artinya data ini diplot dalam peta

4) **70** = QUAL

5) **40** = 0 SA Q1 Q0 RF 0 0 N8

Dalam bentuk binernya adalah **40** = 0100 0000

- 0 = 0
- SA = 1 artinya *associated*
- Q = 0
- Q0 = 0
- RF = 0

- 0 = 0
- 0 = 0
- N8 = 0

6) $C2 = N7 N6 N5 N4 N3 N2 N1 N0$

Dalam bentuk binernya adalah $C2 = 1100 0010$

- N7 = 1
- N6 = 1
- N5 = 0
- N4 = 0
- N3 = 0
- N2 = 0
- N1 = 1
- N0 = 0

Karena *track number* membutuhkan nilai N8, maka binernya didapat

$$0 1100 0010 = 0 + 128 + 64 + 0 + 0 + 0 + 0 + 2 + 0 = 194$$

Jadi track numbernya adalah 194

- 7) **FD** } FD = 1111 1101
 8) **90** } 90 = 1001 0000

Penjabarannya dapat dilihat pada table 4.3 dibawah ini

Tabel 4.3 Penjelasan Posisi X

X ₁₅	X ₁₄	X ₁₃	X ₁₂	X ₁₁	X ₁₀	X ₉	X ₈	X ₇	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀
1	1	1	1	1	1	0	1	1	0	0	1	0	0	0	0

Jadi binernya 1111 1101 1001 0000 = 64912 (desimal)

Karena $X_0 = 1/32$ nautical, dan

1 Nautical Mile (nmi) = 1852 m, maka :

$$\frac{64912}{32} \times 1,852 \text{ Km} = 3756,782 \text{ Km}$$

Selanjutnya dikonversi kedalam bentuk derajat.

1 derajat = 1113619 km

1 derajat ~ 1113620 km

$$\text{Jadi } \frac{3756,782 \text{ Km}}{1113620 \text{ Km}} = 0,00337 \text{ derajat}$$

- 9) **0** } 00 = 0000 0000
 10) **10** } 10 = 0001 0000

Penjabarannya dapat dilihat pada table 4.4 dibawah ini

Tabel 4.4 Penjelasan Posisi Y

Y ₁₅	Y ₁₄	Y ₁₃	Y ₁₂	Y ₁₁	Y ₁₀	Y ₉	Y ₈	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Jadi binernya 0000 0000 0001 0000 = 16 (desimal)

Karena $X_0 = 1/32$ nautical, dan

1 Nautical Mile (nmi) = 1852 m, maka :

$$\frac{16}{32} = 1,852 \text{ km} = 0,926 \text{ km}$$

Selanjutnya dikonversi kedalam bentuk derajat.

1 derajat = 1113619 km

1 derajat ~ 1113620 km

$$\text{Jadi } \frac{0,926 \text{ km}}{1113620 \text{ km}} \sim 0 \text{ derajat}$$

- 11) **E2** = DX = 1110 0010 = 226

DX = mendefinisikan arah kecepatan X

- 12) **3** = 03 = DY = 0000 0011 = 3

DY = mendefinisikan arah kecepatan Y

- 13) **31** } 31 = 0011 0001

- 14) **25** } 25 = 0010 1001

Penjelasannya dapat dilihat pada table 4.5 dibawah ini

Tabel 4.5 Penjelasan Mode *Flight Level*

C	M2	M1	M0	Code 1											
0	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1

Dari penjabaran tabel diatas, didapat keterangan bahwa,

C = 0, hal ini berarti kode *message* tidak dirusak (*garbled*)

dan jika dilihat urutan M2, M1, M0 pada tabel didapat mode C (lihat tabel 4.1 format data radar sekunder)

jika dilihat dari modenya, maka *flight level* = *code* 1, dimana *flight level* dalam bentuk heksadecimal yaitu 0001 0010 1001 = 125

FL = 125

15) **19** } 0001 1001
 16) **A1** } 1010 0001

Penjelasannya dapat dilihat pada tabel 4.6 dibawah ini

Tabel 4.6 Penjelasan Mode *SSR Code*

C	M2	M1	M0	Code 2												
0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	0	1

Dari penjabaran tabel diatas bahwa didapat keterangan bahwa,

C = 0, hal ini berarti kode *message* tidak dirusak (*garbled*)

dan jika dilihat urutan M2, M1, M0 = 001, pada tabel didapat mode A (lihat table 4.1 format data radar sekunder)

jika dilihat dari modenya, maka *SSR Code* dapat dilihat pada *code* 2 dalam bentuk desimal dengan ketentuan 3 dijit setiap 4 bit, yaitu :

Code 2 : 1001 1010 0001
 $\underbrace{\quad\quad\quad\quad}_{4} \quad \underbrace{\quad\quad\quad\quad}_{6} \quad \underbrace{\quad\quad\quad\quad}_{4} \quad \underbrace{\quad\quad\quad\quad}_{1}$

Jadi didapat *SSR Codenya* 4641

17) 0

18) 0

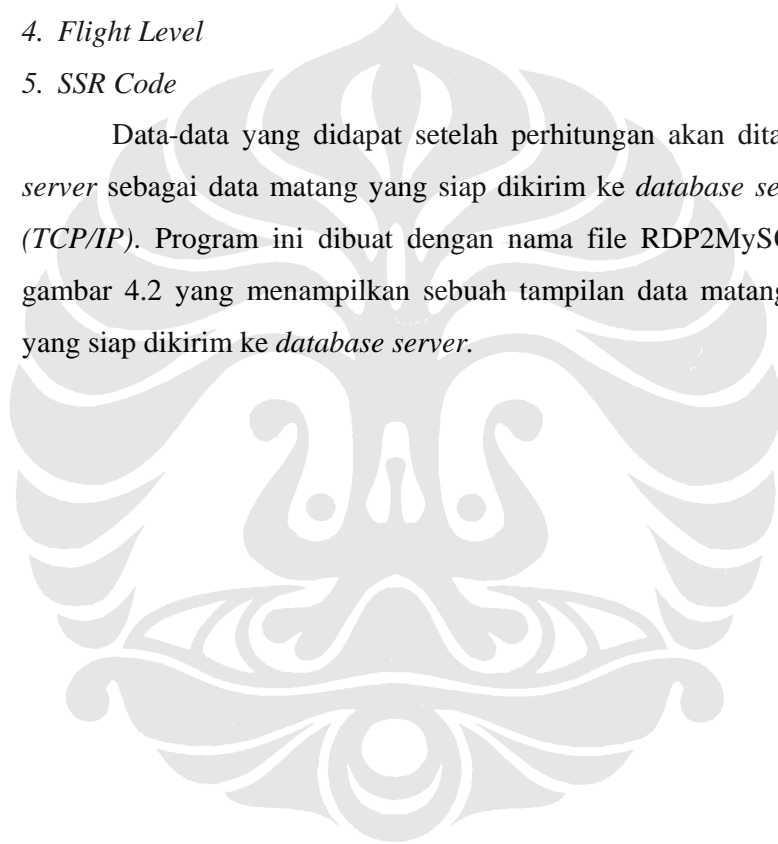
19) 3

20) 41

Dari perhitungan pada *raw* data diatas, dibuatlah sebuah program yang mampu menterjemahkan *raw* data menjadi data yang lebih matang. Program ini dibuat dengan menggunakan program bantuan yang ditujukan untuk melakukan proses perhitungan dan filterisasi data mana saja yang dipergunakan untuk ditampilkan, sehingga didapat data-data yang lebih spesifik, diantaranya

1. *Track Number*
2. Posisi X
3. Posisi Y
4. *Flight Level*
5. *SSR Code*

Data-data yang didapat setelah perhitungan akan ditampilkan oleh *RDP server* sebagai data matang yang siap dikirim ke *database server* melalui *socket (TCP/IP)*. Program ini dibuat dengan nama file *RDP2MySQL*. Berikut adalah gambar 4.2 yang menampilkan sebuah tampilan data matang pada *RDP server* yang siap dikirim ke *database server*.



```

TrackNumber : 194
posisi X : -35188
posisi Y : 0
Flight Level : 125
SSR code : 4641
Data (byte) : 0 c2 ff ff 76 8c 0 0 0 0 4 6 4 1 0 7d 0
INSERT/UPDATE TrackNumber:194 succeeded: 1 row affected
-----

TrackNumber : 211
posisi X : -68524
posisi Y : 29632
Flight Level : 199
SSR code : 4640
Data (byte) : 0 d3 ff fe f4 54 0 0 73 c0 4 6 4 0 0 c7 0
INSERT/UPDATE TrackNumber:211 succeeded: 1 row affected
-----

TrackNumber : 275
posisi X : -187052
posisi Y : 125936
Flight Level : 350
SSR code : 7133
Data (byte) : 1 13 ff fd 25 54 0 1 eb f0 7 1 3 3 1 5e 0
INSERT/UPDATE TrackNumber:275 succeeded: 1 row affected
-----

TrackNumber : 480
posisi X : -233352
posisi Y : 179644
Flight Level : 260
SSR code : 4643
Data (byte) : 1 e0 ff fc 70 78 0 2 bd bc 4 6 4 3 1 4 0
INSERT/UPDATE TrackNumber:480 succeeded: 1 row affected

```

Gambar 4.2 data matang setelah diterjemahkan

Setelah data terkirim maka *database server* akan menghimpun data-data yang masuk, sehingga *database server* dibuat dengan *field* yang sesuai dengan aspek-aspek yang dikirim oleh *RDP server*.

4.3. Analisis Program

Dari *database server* maka dibuat sebuah program *mapserver* untuk melakukan pemetaan pada data-data yang telah terhimpun dalam *database*.

Ada beberapa aspek dalam sistem pemetaan yang ditampilkan, yaitu :

1. Peta yang digunakan serta perangkat-perangkatnya

Peta yang dipergunakan adalah peta Jawa Barat, DKI Jakarta dan Banten. Hal ini digunakan karena data yang diperoleh dari lapangan didapatkan bahwa objek yang terdeteksi berada pada wilayah pulau Jawa bagian barat.

Adapun masing-masing propinsi *dibuat class* dan *stylenya* masing-masing, dan diperoleh dari *library* yang dimiliki oleh program *mapserver* itu sendiri.

```
// objek layer => ini mendefinisikan layer
$objLayerJbr = ms_newLayerObj($objMap);
$objLayerJbr->set ("name", "Kabupaten");
$objLayerJbr->set ("type", MS_LAYER_POLYGON);
$objLayerJbr->set ("status", MS_DEFAULT); // template html
$objLayerJbr->set ("data", "jabar.shp");
$objLayerJbr->set ("labelitem", "name2"); // label layer
$objLayerJbr->set ("classitem", "kodeprop"); // field kelas
$objLayerJbr->set
    ("header", "c:/buku_mapserver/html/header26_prop.htm");
$objLayerJbr->set
    ("footer", "c:/buku_mapserver/html/footer26_prop.htm");
//
```

Dari program diatas dibuat untuk membuat layer baru yang berisikan peta jawa barat dengan menggunakan sumber file “jabar.shp”, yang dibatasi dengan kabupaten di seluruh pulau jawa barat. Untuk pengaturan class dan stylenya umumnya ketiga propinsi (jawa barat, DKI Jakarta, Banten) adalah sama dalam hal memberikan label, set warna peta, dan lain-lain.

```
// objek class & style jabar
$objClassJbr = ms_newClassObj ($objLayerJbr);
$objClassJbr->Set ("name", "Jabar");
$objClassJbr->SetExpression ("3"); // kodeprop=1
$objClassJbr->Set("template",
c:/buku_mapserver/html/query25_prop.html");
$objStyleJbr = ms_newStyleObj ($objClassJbr);
$objStyleJbr->color->setRGB (240,252,258);
$objStyleJbr->outlinecolor->SetRGB (0,0,0);
// objek label kelas
$objClassJbr->label->Set ("font", "arial");
$objClassJbr->label->Set ("type", MS_TRUEETYPE);
$objClassJbr->label->Set ("size", 8);
$objClassJbr->label->Set ("position", MS_CC);
$objClassJbr->label->color->SetRGB (0, 0, 254);
```

```

// objek class & style DKI
$objClassDki = ms_newClassObj ($objLayerJbr);
$objClassDki->Set ("name", "DKI");
$objClassDki->SetExpression ("2"); // kodeprop=2
$objClassDki->Set("template",
"c:/buku_mapserver/html/query25_prop.html");
$objStyleDki = ms_newStyleObj ($objClassDki);
$objStyleDki->color->setRGB (237, 177, 203);
$objStyleDki->outlinecolor->SetRGB (0,0,0);
// objek label kelas
$objClassDki->label->Set ("font", "arialbold");
$objClassDki->label->Set ("type", MS_TRUEETYPE);
$objClassDki->label->Set ("size", 8);
$objClassDki->label->Set ("position", MS_CC);
$objClassDki->label->color->SetRGB (0, 0, 0);

// objek class & style Banten
$objClassBtn = ms_newClassObj ($objLayerJbr);
$objClassBtn->Set ("name", "Banten");
$objClassBtn->SetExpression ("1"); // kodeprop=1
$objClassBtn->Set("template", "c:/buku_mapserver/html/query25_prop.html");
$objStyleBtn = ms_newStyleObj ($objClassBtn);
$objStyleBtn->color->setRGB (172, 247, 60);
$objStyleBtn->outlinecolor->SetRGB (0,0,0);
// objek label kelas
$objClassBtn->label->Set ("font", "comicsans");
$objClassBtn->label->Set ("type", MS_TRUEETYPE);
$objClassBtn->label->Set ("size", 8);
$objClassBtn->label->Set ("position", MS_CC);
$objClassBtn->label->color->SetRGB (254, 32, 40);

```

2. Plotting data objek

Plot disini adalah plot data objek yang terdeteksi oleh radar kedalam pemetaan yang telah tersedia. Adapun programnya dapat dijelaskan sebagai berikut :

```

// objek layer Point
$objLayerPoint[ ] = ms_newLayerObj($objMap);

```



```

$objLayerPoint[$i]->set ("name", "Point");
$objLayerPoint[$i]->set ("type", MS_LAYER_POINT);
$objLayerPoint[$i]->set ("status", MS_DEFAULT);

```

Program diatas merupakan pembuatan layer untuk point (plot). Untuk menentukan koordinat pada objek yang terdeteksi, perlu dilakukan perhitungan awal jarak antara titik pusat radar dengan objek. Seperti dijelaskan sebelumnya bahwa Raw data radar akan menunjukkan posisi X dan Y. posisi X dan Y ini adalah posisi kedudukan objek terhadap radar.

Misalnya pada posisi X

binernya 1111 1101 1001 0000 = 64912 (desimal)

Karena $X_0 = 1/32$ nautical, dan

1 Nautical Mile (nmi) = 1852 m, maka :

$$\frac{64912}{32} \times 1,852 \text{ Km} = 3756,782 \text{ Km}$$

Selanjutnya dikonversi kedalam bentuk derajat.

1 derajat = 1113619 km

1 derajat ~ 1113620 km

$$\text{Jadi } \frac{3756,782 \text{ Km}}{1113620 \text{ Km}} = 0,00337 \text{ derajat}$$

Dan pada posisi Y

binernya 0000 0000 0001 0000 = 16 (decimal)

Karena $X_0 = 1/32$ nautical, dan

1 Nautical Mile (nmi) = 1852 m, maka :

$$\frac{16}{32} = 1,852 \text{ km} = 0,926 \text{ km}$$

Selanjutnya dikonversi kedalam bentuk derajat.

1 derajat = 1113619 km

1 derajat ~ 1113620 km

$$\text{Jadi } \frac{0,926 \text{ km}}{1113620 \text{ km}} \sim 0 \text{ derajat}$$

Ini adalah posisi X dan Y dalam derajat terhadap radar. Untuk mengetahui letak koordinat objek radar, maka dapat dilakukan dengan menjumlahkan koordinat radar dengan derajat objek yang diperoleh melalui perhitungan. Sehingga program *Mapserver* yang dibuat adalah :

```
$posTargetX[ ] = $Koordinat_Radar_X+$posX[$i];
$posTargetY[ ] = $Koordinat_Radar_Y+$posY[$i];
```

Dimana koordinat radar ditentukan sebelumnya yaitu :

```
// Koordinat Radar
$Koordinat_Radar_X = 107;
$Koordinat_Radar_Y = -7;
```

Sehingga didapat :

Derajat X = 0,00337 => koordinat X = $107 + 0,00337 = 107,00337$

Derajat Y = 0 => koordinat Y = $-7 + 0 = -7$

Jadi yang diplot kedalam peta adalah posisi target X dan Y, yaitu :

Koordinat (X,Y) = (107,00337, -7)

3. Keterangan penunjang

Untuk keterangan penunjang hanya sebagai informasi keterangan objek, dimana informasi itu adalah tracknumber, posisi x, posisi y, Flight Level. Keempat informasi penting ini harus ditampilkan pada sistem pemetaan sehingga user akan mudah membaca kondisi peta objek pesawat. Untuk itu dibuat program sebagai berikut :

```
$objClassPoint[$i]->Set("name", "TrackNumber:$TrackNumber[$i];
Koordinat:($posTargetX[$i], $posTargetY[$i]);FlightLevel:$FL[$i]
;");
```

4. Sistem navigasi

Sistem navigasi merupakan standarisasi sebuah peta, dimana navigasi, arah adalah suatu keterangan penunjang setiap peta. Pada skripsi ini mencoba

mengkombinasikan teknik pemetaan pada umumnya dengan sistem monitoring data radar.

5. Koneksi dengan *database server*

Koneksi ini sangat diperlukan mengingat sumber data dari map server ini adalah *databaseny*a. Untuk itu dibuat sebuah program sebagai berikut :

```
//Query Database  
mysql_connect("localhost","root","");  
mysql_select_db ("dbmap");
```

koneksi ini diawali dengan data di RDP server yang lokasinya di localhost/root lalu dihubungkan ke *database* server dengan nama “dbmap”. Sehingga data yang masuk ke “dbmap” dari localhost/root akan langsung di respon oleh sistem pemetaan dengan mem-plot data-data yang masuk.

BAB IV KESIMPULAN

Dari uraian teori dan pengembangan dalam perancangan pada skripsi ini, maka dengan ini dapat diambil beberapa kesimpulan, diantaranya :

1. Radar merupakan salah satu komponen sangat penting sebagai pengintai objek pada sistem monitoring lalu lintas udara
2. Sistem monitoring data radar menggunakan *Mapserver* merupakan sebuah pengembangan teknologi sistem monitoring data radar pada aplikasi lalu lintas udara sebelumnya
3. Kelebihan pengembangan ini terletak pada luasnya wilayah dalam mengakses monitoring data radar dengan persyaratan memiliki jaringan yang terhubung terhadap web server pusat pengolah basis data radar
4. *Mapserver* merupakan salah satu perangkat lunak berbasis SIG yang mampu menampilkan sistem pemetaan dengan data-data yang berkaitan pada peta tersebut

Adapun pada perancangan ini terdapat beberapa kelemahan yang mungkin akan dikembangkan oleh pembahasan selanjutnya. Kelemahan-kelemahan pada perancangan ini diantaranya :

1. Perancangan sistem ini masih berupa simulasi, belum dapat diaplikasikan pada kondisi sesungguhnya.
2. Keandalan *server* masih belum standar dalam fungsinya sebagai *web server* yang bertugas sebagai penyedia informasi.
3. Tampilan peta pada *web server* masih jauh dari kesempurnaan, sehingga dapat dikembangkan dengan tampilan yang lebih menarik dan fleksibel.

DAFTAR ACUAN

- [1] Ir. Hj. Ny. Rochmah N, Sukardi M. Eng Sc, “*Diktat Perencanaan Sistem Transmisi*”, Teknik Elektro, Universitas Indonesia, Depok, 2008
- [2] <http://www.radartutorial.eu>
- [3] <http://mapserver.gis.umn.edu/> 2 September 2008
- [4] Map Server For Windows, <http://www.maptools.org/ms4w/index/phtml/> 29 September 2008
- [5] Dana Indra Sensuse, “*Desain Sistem Informasi Sinkronisasi Ruang Berbasis Web*”, Magister Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Indonesia
- [6] Yomi Agung Susanto “*Aplikasi Sistem Informasi Geografis Berbasis Web (WEBGIS) Untuk Pengembangan Sektor Industri Di Kabupaten Pacitan*”, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Trunojoyo
- [7] Gatot H. Pramono, “*Development of Coastal And Marine Spatial Catalog*”, Proceeding Geo-Marine Research Forum 2007
- [8] Dirjen Penataan Ruang, Departemen Pekerjaan Umum, “*Operasionalisasi WEB Penataan Ruang dan GIS Penataan Ruang*”
- [9] Firman H, Soni D, Ketut W, “*Pengolahan Data Penginderaan Jauh dan Pembangunan SIG Berbasis Open Source*”, Kelompok Keahlian Penginderaan Jauh dan Sains Informasi Geografis, Departemen Teknik Geodesi, Institute Teknologi Bandung, Bandung 2005

- [10]Iwan Setiawan, Harris Noor Rabbasa, “*Aplikasi Open Source untuk Pemetaan Online*”, Program Studi Teknologi Informasi untuk Pengelolaan SDA (MIT), Institut Pertanian Bogor, Bogor,
- [11]Scott Davis, “*GIS for Web Developers Adding Where to Your Web Applications*”, The Pragmatic Programmers LLC, 2007
- [12] vITraining.com, “*Automatic Vehicle Location System Utilization of GIS, GPS, Wireless, and Internet Technology*”, Bandung
- [13]Arif darmawan, “*Sekilas tentang Sistem Informasi Geografis (Geographic Information System)*”, IlmuKomputer.com, 2006
- [14]Nur Meita Indah Mufidah, “*Pengantar GIS (Gographical Information System)*”, IlmuKomputer.com, 2006
- [15]Dian Ardiyansah, “*Teknologi Jaringan Komputer*”, IlmuKomputer.com, 2003
- [16]Harindra Wisnu Pradhana, “*Dasar-dasar web-design dengan PHP dan MySQL*”, Teknik Elektro Universitas Diponegoro, Semarang, 2006
- [17]Anon Kuncoro Widigdo, “*Dasar Pemrograman PHP dan MySQL*”, IlmuKomputer.com, 2003

DAFTAR PUSTAKA

- Eddy Prahasta, “*Sistem Informasi Geografis : Membangun Aplikasi Web-Based GIS dengan MapServer*” Penerbit Informatika, Bandung, 2006.
- Eddy Prahasta, “*Sistem Informasi Geografis : Aplikasi Pemrograman Map Info*” Penerbit Informatika, Bandung, 2005
- Eddy Prahasta, “*Sistem Informasi Geografis : Belajar dan Memahami Map Info*” Penerbit Informatika, Bandung, 2004
- Arif darmawan, “*Sekilas tentang Sistem Informasi Geografis (Geographic Information System)*”, IlmuKomputer.com, 2006
- Nur Meita Indah Mufidah, “*Pengantar GIS (Gographical Information System)*”, IlmuKomputer.com, 2006
- Ir. Hj. Ny. Rochmah N, Sukardi M. Eng Sc, “*Diktat Perencanaan Sistem Transmisi*”, Teknik Elektro, Universitas Indonesia, Depok, 2008
- <http://www.radartutorial.eu>
- Sidik B, “*Pemrograman Web dengan PHP*” Penerbit Informatika, Bandung, 2004
- Widigdo Anon Kuncoro, “*Dasar Pemrograman PHP dan MySQL*”, IlmuKomputer.com, 2003

Lampiran

Listing Program Map Server

```

<?php
//Query Database
mysql_connect("localhost","root","");
mysql_select_db ("dbmap");

//Delete file temporary gif c:\temp
system('del c:\temp\*.gif');

// Koordinat Radar
$Koordinat_Radar_X = 107;
$Koordinat_Radar_Y = -7;

if (!extension_loaded ("MapScript"))
dl ('php_mapscript_44.'.PHP_SHLIB_SUFFIX);
// objek map di mapfile
$GLOBALS["objMap"] = ms_NewMapObj ("");
$GLOBALS["LokasiKlik"] = ms_newPointObj();
$objMap->Set ("name", "Kab");
//$objMap->setSize (384,204);
$objMap->setSize (800,600);
//$objMap->setExtent (105.06, -7.86, 108.90, -5.82);
$objMap->setExtent (106.50, -7.00, 108.00, -6.95); // modifikasi
$objMap->Set("units", MS_DD); // derajat
$objMap->imagecolor->SetRGB ( 210, 233, 255);
$objMap->Set ("shapepath", "c:/buku_mapserver/shp/");
$objMap->SetSymbolSet ("c:/buku_mapserver/simbol/simbol.sym");
$objMap->SetFontSet ("c:/buku_mapserver/font/font.dat");
//
// objek web di mapfile
$objMap->web->set ("imagepath", "c:/temp/");
$objMap->web->set ("imageurl", "c:/temp/");
//
// objek layer
$objLayerJbr = ms_newLayerObj($objMap);
$objLayerJbr->set ("name", "Kabupaten");
$objLayerJbr->set ("type", MS_LAYER_POLYGON);
$objLayerJbr->set ("status", MS_DEFAULT); // template html
$objLayerJbr->set ("data","jabar.shp");
$objLayerJbr->set ("labelitem","name2"); // label layer
$objLayerJbr->set ("classitem","kodeprop"); // field kelas
$objLayerJbr->set ("header","c:/buku_mapserver/html/header26_prop.htm");
$objLayerJbr->set ("footer","c:/buku_mapserver/html/footer26_prop.htm");
//
// objek class & style jabar
$objClassJbr = ms_newClassObj ($objLayerJbr);

```


(lanjutan)

```

$objClassJbr->Set ("name", "Jabar");
$objClassJbr->SetExpression ("3"); // kodeprop=1
$objClassJbr->Set ("template", "c:/buku_mapserver/html/query25_prop.html");
$objStyleJbr = ms_newStyleObj ($objClassJbr);
$objStyleJbr->color->setRGB (240,252,258);
$objStyleJbr->outlinecolor->SetRGB (0,0,0);
//
// objek label kelas
$objClassJbr->label->Set ("font", "arial");
$objClassJbr->label->Set ("type", MS_TRUETYPE);
$objClassJbr->label->Set ("size", 8);
$objClassJbr->label->Set ("position", MS_CC);
$objClassJbr->label->color->SetRGB (0, 0, 254);
//
// objek class & style DKI
$objClassDki = ms_newClassObj ($objLayerJbr);
$objClassDki->Set ("name", "DKI");
$objClassDki->SetExpression ("2"); // kodeprop=2
$objClassDki->Set ("template", "c:/buku_mapserver/html/query25_prop.html");
$objStyleDki = ms_newStyleObj ($objClassDki);
$objStyleDki->color->setRGB (237, 177, 203);
$objStyleDki->outlinecolor->SetRGB (0,0,0);
//
// objek label kelas
$objClassDki->label->Set ("font", "arialbold");
$objClassDki->label->Set ("type", MS_TRUETYPE);
$objClassDki->label->Set ("size", 8);
$objClassDki->label->Set ("position", MS_CC);
$objClassDki->label->color->SetRGB (0, 0, 0);
//
// objek class & style Banten
$objClassBtn = ms_newClassObj ($objLayerJbr);
$objClassBtn->Set ("name", "Banten");
$objClassBtn->SetExpression ("1"); // kodeprop=1
$objClassBtn->Set ("template", "c:/buku_mapserver/html/query25_prop.html");
$objStyleBtn = ms_newStyleObj ($objClassBtn);
$objStyleBtn->color->setRGB (172, 247, 60);
$objStyleBtn->outlinecolor->SetRGB (0,0,0);
//
// objek label kelas
$objClassBtn->label->Set ("font", "comicans");
$objClassBtn->label->Set ("type", MS_TRUETYPE);
$objClassBtn->label->Set ("size", 8);
$objClassBtn->label->Set ("position", MS_CC);
$objClassBtn->label->color->SetRGB (254, 32, 40);

//      Query data Mysql

```

(lanjutan)

```

$query=mysql_query("select * from tdata");
$jumlah=mysql_num_rows($query);
if ($jumlah==0) { echo "Belum ada data";}
else {
$i=0;
while($row=mysql_fetch_array($query)){
$TrackNumber[] = $row[TrackNumber];
$posX[] = $row[PosX]/1113620; // konversi, 1 derajat = 111.3619 km
$posY[] = $row[PosY]/1113620; // konversi, 1 derajat = 111.3619 km
$FL[] = $row[FL];
$SSR[] = $row[SSRCode];
// objek layer Point
$objLayerPoint[] = ms_newLayerObj($objMap);
$objLayerPoint[$i]->set ("name", "Point");
$objLayerPoint[$i]->set ("type", MS_LAYER_POINT);
$objLayerPoint[$i]->set ("status", MS_DEFAULT);

$objCoordList[] = ms_newLineObj();
$objPointShape[] = ms_newShapeObj(MS_SHAPE_POINT);

$posTargetX[] = $Koordinat_Radar_X+$posX[$i];
$posTargetY[] = $Koordinat_Radar_Y+$posY[$i];
//$posTargetX[] = 107;
//$posTargetY[] = -7;

$objCoordList[$i]->addXY($posTargetX[$i],$posTargetY[$i]);
$objPointShape[$i]->add($objCoordList[$i]);
$objLayerPoint[$i]->addFeature($objPointShape[$i]);

// objek class & style Point
$a = 10*$i;
$objClassPoint[] = ms_newClassObj ($objLayerPoint[$i]);
$objClassPoint[$i]->Set ("name", "TrackNumber : $TrackNumber[$i];
Koordinat: ($posTargetX[$i],$posTargetY[$i]); FlightLevel:$FL[$i];");
$objStylePoint[] = ms_newStyleObj($objClassPoint[$i]);
$objStylePoint[$i]->color->setRGB(0,0,0);
//$objStylePoint[$i]->outlinecolor->setRGB(0,255,255);
$objStylePoint[$i]->outlinecolor->setRGB(2*$a,255-2*$a,255-2*$a);
$objStylePoint[$i]->set("symbolname", "pesawat");
$objStylePoint[$i]->set( "size", "15");
// continue Query Mysql
$i++;
}
}
// end

//

```

(lanjutan)

```

// scalebar, skala
$objMap->scalebar->Set ("status", MS_ON); // selalu muncul
$objMap->scalebar->Set ("style", 0);
$objMap->scalebar->Set ("intervals", 3);
$objMap->scalebar->Set ("height", 4);
$objMap->scalebar->Set ("width", 150);
$objMap->scalebar->color->SetRGB(0,0,0);
$objMap->scalebar->backgroundcolor->SetRGB (255,255,255);
$objMap->scalebar->outlinecolor->SetRGB (0,0,0);
$objMap->scalebar->Set ("units", 4); // 1 feet, 2 mil, 3 meter, 4 km
$objMap->scalebar->Set ("position", MS_LL); // bawah-kiri
$objMap->scalebar->Set ("transparent", 1); // 1 true, 0 false
$objMap->scalebar->label->Set ("font", "arial");
$objMap->scalebar->label->Set ("size", MS_MEDIUM);
$objMap->scalebar->label->color->SetRGB (0,0,0);
//
// proyeksi peta
$strProj = "proj=latlong,ellps=WGS84,datum=WGS84";
$objMap->SetProjection ($strProj, MS_FALSE);
//
// format output
$objMap->outputformat->Set ("name", "Format GIF");
$objMap->outputformat->Set ("extension", "gif");
//
// legenda
$objMap->legend->Set ("status", MS_DEFAULT); // selalu muncul
$objMap->legend->Set ("keysize", 20);
$objMap->legend->Set ("keysizey", 13);
$objMap->legend->Set ("keyspacingx", 20);
$objMap->legend->Set ("keyspacingy", 5);
$objMap->legend->Set ("postlabelcache", 1); // true
$objMap->legend->Set ("transparent", 0); // 0 false, 1 true
$objMap->legend->outlinecolor->SetRGB (0,0,0);
$objMap->legend->label->Set ("font", "arial");
$objMap->legend->label->Set ("position", 1); // rata kiri
$objMap->legend->label->Set ("size", 9);
$objMap->legend->label->Set ("offsetx", -10);
$objMap->legend->label->Set ("offsety", -13);
$objMap->legend->label->Set ("antialias", 50);
$objMap->legend->label->Set ("type", MS_TRUETYPE);
$objMap->legend->label->color->SetRGB (0,0,0);
//
// objek layer grid
$objLayerGrid = ms_newLayerObj($objMap);
$objLayerGrid->Set ("name", "Grid");
$objLayerGrid->set ( "type", MS_LAYER_LINE);
$objLayerGrid->Set ("status", MS_ON); // bisa dirubah, on/off

```

(lanjutan)

```

ms_newGridObj ($objLayerGrid); // layer menjadi layer grid
$objLayerGrid->grid->Set ("labelformat", "DDMM");
$objLayerGrid->grid->Set ("minsubdivide", 16);
$objLayerGrid->grid->Set ("maxsubdivide", 64);
$objLayerGrid->grid->Set ("mininterval", 1.0);
$objLayerGrid->grid->Set ("maxinterval", 2.0);
$objLayerGrid->grid->Set ("minarcs", 2);
$objLayerGrid->grid->Set ("maxarcs", 64);
$objClassGrid = ms_newClassObj ($objLayerGrid);
$objClassGrid->Set ("name", "Grid"); // graticule
$objStyleGrid = ms_newStyleObj ($objClassGrid);
$objStyleGrid->color->setRGB (0, 0, 0);
$objClassGrid->label->Set ("font", "arial");
$objClassGrid->label->Set ("type", MS_TRUEATYPE);
$objClassGrid->label->Set ("size", 8);
$objClassGrid->label->Set ("position", MS_UR);
$objClassGrid->label->color->setRGB (0, 0, 0);
//
$FaktorSkala = 2; // default, bisa dirubah
if (isset ($HTTP_POST_VARS["mapa_x"]) && isset ($HTTP_POST_VARS["mapa_y"])
&&
!isset ($HTTP_POST_VARS["ZoomFull"]))
{
$check_pan = "CHECKED";
$objExtent = explode (" ", $HTTP_POST_VARS["extent"]);
$GLOBALS["PosisiKlik"] = ms_newPointObj();
$PosisiKlikX = $HTTP_POST_VARS["mapa_x"];
$PosisiKlikY = $HTTP_POST_VARS["mapa_y"];
$PosisiKlik->SetXY ($PosisiKlikX, $PosisiKlikY);
$MinX = $objExtent[0];
$MinY = $objExtent[1];
$MaxX = $objExtent[2];
$MaxY = $objExtent[3];
$objKotak = ms_newRectObj();
$objKotak->SetExtent ($MinX, $MinY, $MaxX, $MaxY);
$IndeksZoom = $HTTP_POST_VARS["zoom"];
$NilaiPerbesaran = $HTTP_POST_VARS["Perbesaran"];
$SkalaZoom = $IndeksZoom * $NilaiPerbesaran;
$StatQuery = $HTTP_POST_VARS["rQuery"];
IF ($StatQuery == "on")
{
$StatQuery == "on";
$check_pan = "";
$check_zoomout = "";
$check_zoomin = "";
$check_query = "CHECKED";
$StatusFungsi = "ON";
}
}

```

(lanjutan)

```

} // akhir query
else
{
$StatusFungsi = "OFF";
if ($SkalaZoom == 0)
{
$SkalaZoom = 1;// maksa pan
$check_pan = "CHECKED";
$check_zoomout = "";
$check_zoomin = "";
$check_query = "";
}
else if ($SkalaZoom < 0)
{
$check_pan = "";
$check_zoomout = "CHECKED";
$check_zoomin = "";
$check_query = "";
}
else if ($SkalaZoom > 0)
{
$check_pan = "";
$check_zoomout = "";
$check_zoomin = "CHECKED";
$check_query = "";
}
$objMap->ZoomPoint ($SkalaZoom, $PosisiKlik, $objMap->width, $objMap-
>height, $objKotak);
} // else
} // isset
//
>NamaLayer = 'LayerGrid'; //nama chekbox
if (isset ($NamaLayer) && $_POST['LayerGrid'] == "on")
{
$objLayerGrid->Set ('status', MS_ON);
} // if
else
{
$objLayerGrid->Set ('status', MS_OFF);
} // else
//
IF ($StatQuery == "on") { $objImagePeta = $objMap->DrawQuery(); }
ELSE {$objImagePeta = $objMap->Draw(); }
$objUrlPeta = $objImagePeta->SaveWebImage();
$objRect = $objMap->extent->minx." ".$objMap->extent->miny." "
.$objMap->extent->maxx." ".$objMap->extent->maxy;
$objImageSkala = $objMap->DrawScalebar();

```

(lanjutan)

```

$objUrlSkala = $objImageSkala-> SaveWebImage();
$objImageLegenda = $objMap->DrawLegend();
$objUrlLegenda = $objImageLegenda-> SaveWebImage();
//
// kumpulan fungsi
//
function KonvPixToLB ($xPix, $yPix)
{
    $xmi = $GLOBALS["objMap"]->extent->minx;
    $ymi = $GLOBALS["objMap"]->extent->miny;
    $xma = $GLOBALS["objMap"]->extent->maxx;
    $yma = $GLOBALS["objMap"]->extent->maxy;
    $PanjangPeta = $xma - $xmi;
    $LebarPeta = $yma - $ymi;
    $PanjangImage = $GLOBALS["objMap"]->width;
    $LebarImage = $GLOBALS["objMap"]->height;
    $RasioX = $PanjangPeta / $PanjangImage;
    $RasioY = $LebarPeta / $LebarImage;
    $x1Pix = $xPix;
    $y1Pix = $LebarImage - $yPix;
    $AbMap = $xmi + $x1Pix * $RasioX;
    $OrMap = $ymi + $y1Pix * $RasioY;
    $GLOBALS["LokasiKlik"]->SetXY ($AbMap,$OrMap);
}

function LakukanQuery()
{
    KonvPixToLB ($GLOBALS["PosisiKlik"]->x, $GLOBALS["PosisiKlik"]->y);
    $oLayerProp = $GLOBALS["objMap"]->GetLayerByName("Kabupaten");
    $oLayerBts = $GLOBALS["objMap"]->GetLayerByName("Antena");
    $NamaField = "Name2";
    $NamaDataValue = "Bandung";
    if (@$oLayerProp->QueryByAttributes($NamaField, $NamaDataValue,
    MS_SINGLE)==MS_SUCCESS)
    {
        $hasilProp = $oLayerProp->GetResult(0);
        if ($hasilProp > 0)
        {
            $oLayerProp->Open();
            $oShapeProp = $oLayerProp->GetShape ($hasilProp->tileindex, $hasilProp->
            >shapeindex);
            $DVProp = $oShapeProp->values["NAME1"];
            echo $DVProp; printf ("<br>");
            $oLayerProp->Close;
        } // if
    } // if
    if (@$oLayerBts->QueryByFeatures(0)==MS_SUCCESS)

```

(lanjutan)

```

{
$jmlHasil = $oLayerBts->GetNumResults();
for ($i=0; $i<$jmlHasil; $i++)
{
$hasilBts = $oLayerBts->GetResult($i);
$oLayerBts->Open();
$oShapeBts = $oLayerBts->GetShape ($hasilBts->tileindex, $hasilBts-
>shapeindex);
$DvId = $oShapeBts->values["ID"];
$DvOp = $oShapeBts->values["OPERATOR"];
$DvUm = $oShapeBts->values["UMURPER"];
$DvWak = $oShapeBts->values["WAKTUINS"];
$DvTing = $oShapeBts->values["TINGGIMEN"];
$DvKe = $oShapeBts->values["KETINGGIAN"];
echo "Op:"; echo $DvOp; printf (" , ");
echo "Pasang:"; echo $DvWak; printf (" , ");
echo "Antena:"; echo $DvTing; printf (" m <br>");
$oLayerBts->Close;
} // for
} // if
} // akhir fungsi LakukanQuery()
?>

<html>
<head><title>aplikasi peta interaktif</title></head>
<meta http-equiv='refresh' content='10'; url='file_name or URL' />
<link type="text/css" rel="stylesheet" href="c:\buku_mapserver\script\ms35.css"
/></head>
<body>
<form method=POST action=<?php echo $HTTP_SERVER_VARS['PHP_SELF']; ?> >
<table style="text-align: left; width: 100%;" border="3" cellpadding="1"
cellspacing="1">
<tbody>
<tr>
<td align="center" bgcolor="#0099FF"><b>Utara & Legenda</b></td>
<td align="center" bgcolor="#0099FF"><b>Tampilan Peta & Skala
Grafis</b></td>
<td align="center" bgcolor="#0099FF"><b>Fungsi</b></td>
</tr>
<tr>
<td valign="top" bgcolor="#66FFFF">
<img src= "c:/buku_mapserver/simbol/sim_utara.gif"> <br>
<b>Legenda</b><br>
<img src=<?php echo $objUrlLegenda ?>> </td>
<td bgcolor="#FFFF33"> <input type=image name="mapa" src="<?php echo
$objUrlPeta ?>" ></td>
<td valign="top" bgcolor="#99FF99">

```

(lanjutan)

```

<b>Navigasi</b><br>
<input type=radio name="zoom" value=0 <?php echo $check_pan ?> >
<img src= "c:/buku_mapserver/simbol/kursor_pan.gif"> Pan/Grab <br>
<input type=radio name="zoom" value=1 <?php echo $check_zoomin ?> >
<img src= "c:/buku_mapserver/simbol/kursor_zoomin.gif"> Zoom in <br>
<input type=radio name="zoom" value=-1 <?php echo $check_zoomout ?> >
<img src= "c:/buku_mapserver/simbol/kursor_zoomout.gif"> Zoom out<br>
<input type=submit name="ZoomFull" value="Zoom Full"><br><p>
<input type="checkbox" name="rQuery" value="on" <?php echo $check_query
?>>

<b>Query</b><br>
</td>
</tr>
<tr>
<td><input type="checkbox" name="LayerGrid" value="on"
<?php echo ($objLayerGrid->status == MS_ON) ? 'checked="checked"' :
''; ?> />
<br><?php echo $objClassGrid->name; ?>&nbsp;<br>
<input type="submit" value= "Update"/><br>
</td>
<td> <b>Kons.</b>
<?php
$bSkala = (int)$GLOBALS["objMap"]->scale;
$sSkala = (string)$bSkala;
$aSkala = "1:.".sSkala; // gabung 2 string
?>
<input type=text name="Perbesaran" value="<?php echo $FaktorSkala ?>"
size=3>
Skala<input type=text value="<?php echo $aSkala ?>" size=9>
<img src=<?php echo $objUrlSkala ?>> </td>
<td> <?php echo "Posisi klik: "; echo $PosisiKlikX; echo ","; echo
$PosisiKlikY;?> </td>
</tr>
<tr><td><b>Hasil Query</b></td>
<td><?php if ($StatusFungsi == "ON") { LakukanQuery();} else echo ""; ?></td>
<td></td>
</tr>
</tbody>
</table>
<input type=hidden name="extent" value="<?php echo $objRect ?>" >
</form></body>
</html>

```