

PEMANFAATAN APPLET JAVA
PADA PROTOTYPE
MULTISOURCES CLIENT SERVER INFORMATION RETRIEVAL
(MCSIR)
DAN APLIKASI
SISTEM INFORMASI PIMPINAN UNIVERSITAS
(SIPINTAS)

Skripsi diajukan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Ilmu Komputer

Oleh :

FARIAN UJONG

1291000216



UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER
DEPOK
1997

**SKRIPSI : PEMANFAATAN JAVA APPLET PADA PROTOTIPE
MULTISOURCES CLIENT SERVER INFORMATION
RETRIEVAL (MCSIR) DAN APLIKASI SISTEM INFORMASI
PIMPINAN UNIVERSITAS (SIPINTAS).**

NAMA : FARIAN UJONG

NPM : 1291000216

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI

DEPOK, 19 FEBRUARI 1997



SJARIF ABDAT, M.Sc.

PEMBIMBING I

BETTY PURWANDARI, S.Kom

PEMBIMBING II

KATA PENGANTAR

Pertama-tama penulis memanjatkan puji syukur kehadirat Tuhan Yang Maha Esa, berkat terselesaikannya tugas akhir ini. Tugas akhir yang berjudul **PEMANFAATAN JAVA APPLET PADA PROTOTIPE MCSIR DAN APLIKASI SIPINTAS** dapat dikerjakan dengan usaha yang cukup keras dan gigih. Penulis tak lupa juga mengucapkan rasa terima kasih yang sebesar-besarnya kepada :

1. Bapak dan ibu yang telah mendidik dan membesarkan saya dengan penuh perjuangan dan pengorbanan.
2. Bapak Sjarif Abdat, M.Sc. dan Betty Purwandari, S.Kom yang telah membimbing saya selama mengerjakan tugas akhir ini.
3. Ibu Fanny Santosa, M.Sc. selaku pembimbing akademik yang telah banyak membantu saya dalam menyelesaikan studi di Fakultas Ilmu Komputer.
4. Segenap staf, dosen, pihak Sarak dan perpustakaan yang tak dapat saya sebutkan satu persatu, terima kasih atas segala dukungan dan bantuannya.
5. Saudara Hendra Widjaja, Adhimas, Sufatrio dan Dicky Suryadi beserta rekan-rekan mahasiswa sekalian yang tak dapat saya sebutkan satu persatu, terima kasih atas kerjasama dan perhatiannya.

Semoga tugas akhir ini bermanfaat bagi berbagai pihak yang berkepentingan.

Penulis,

1997

ABSTRAK

Kebutuhan akan program yang dapat berjalan di klien semakin terasa penting sejalan dengan penggunaan jaringan Internet yang semakin luas. Pada saat ini program yang dapat berjalan di klien melalui jaringan Internet adalah Applet Java. Applet Java akan berjalan di masing-masing klien tanpa tergantung lagi dengan jenis pelayan yang digunakan oleh pemakai.

Tugas akhir ini bertujuan untuk merancang dan mengimplementasikan Applet Java untuk prototipe *Multisources Client Server Information Retrieval* (MCSIR) dan aplikasi Sistem Informasi Pimpinan Universitas (SIPINTAS). Aplikasi SIPINTAS ini akan memanfaatkan layanan-layanan yang ada pada prototipe MCSIR. Pemanfaatan Applet Java untuk aplikasi SIPINTAS adalah spesifik untuk aplikasi yang bersangkutan. Pemanfaatan Applet Java untuk prototipe MCSIR dibuat umum agar dapat dipakai oleh aplikasi yang memanfaatkan layanan MCSIR.

Aplikasi SIPINTAS yang akan diuji-cobakan adalah aplikasi Mahasiswa Tidak Tepat Waktu (MTTW). Aplikasi ini masih dalam bentuk prototipe dan dikembangkan untuk Ketua Jurusan. Aplikasi SIPINTAS diterapkan dalam bentuk *page Hypertext Markup Language* (HTML) dan data yang diperoleh berasal dari pelayan basis data Sybase.

ix+ 69 hlm.; Lamp.

Bibliografi : 12 (1989-1996)

DAFTAR ISI

LEMBAR PERSETUJUAN	i
KATA PENGANTAR	ii
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Tujuan	2
1.3 Pembatasan Masalah	3
1.4 Metode Penelitian	3
1.5 Sistematika Penulisan	4
BAB II SISTEM INFORMASI EKSEKUTIF	6
2.1 Konsep Sistem Informasi Eksekutif (SIE)	6
2.2 Sistem Informasi Pimpinan Universitas	9
2.2.1 Arsitektur SIPINTAS	10
2.2.2 Model MCSIR	12
2.2.2.a Komponen Presentasi	15
2.2.2.b Komponen Kendali	15
2.2.2.c Komponen Gerbang	15
2.3 Konsep Applet Java	16
2.3.1 Internet	16
2.3.2 Java	17

2.3.2.a Aplikasi Java	19
2.3.2.b Applet Java	20
2.3.2.b.i Parameter Applet Java	22
2.3.2.b.ii Keterbatasan Applet Java	23
BAB III ANALISA KEBUTUHAN DAN RANCANGAN APPLET JAVA	25
3.1 Analisa Kebutuhan	25
3.2 Komunikasi Antar <i>Applet</i>	26
3.2.1 Komunikasi Antar <i>Applet</i> Pada <i>Page</i> Yang Sama	26
3.2.2 Komunikasi Antar <i>Applet</i> Pada <i>Page</i> Yang Berbeda	27
3.3 Rancangan Applet Java	29
3.3.1 Pemanfaatan Applet Java untuk aplikasi	29
3.3.2 Pemanfaatan Applet Java untuk presentasi	31
3.3.3 Pemanfaatan Applet Java untuk kendali	32
3.3.3.a Previous Level dan Next Level	32
3.3.3.a.i Perpindahan List ke Bawah	35
3.3.3.a.ii Perpindahan List ke Atas	37
3.3.3.a.iii Perpindahan List Sejajar	38
3.3.3.b Home Page	39
3.3.3.c Back To Table	39
3.3.3.d Back To Sipintas	41
3.3.3.e Table Map	41
3.3.4 Komunikasi Antar <i>Applet</i> Aplikasi	43
3.3.5 Komunikasi Antar <i>Applet</i> Kendali	44
BAB IV PENERAPAN DAN UJI COBA PROTOTIPE	45
4.1 Penerapan Prototipe Aplikasi Mahasiswa Tidak Tepat Waktu (MTTW) untuk Ketua Jurusan	45
4.1.1 Perhitungan Bobot Angkatan	47

4.1.2 Perhitungan Bobot Tahun Ajaran	47
4.1.3 Peta Page MTTW	48
4.2 Implementasi Applet Java dalam Aplikasi MTTW	51
4.2 Implementasi Applet Java Kendali	51
4.3 Implementasi Applet Java Aplikasi dan Presentasi	53
4.3 Algoritma Modul-Modul Applet Java	56
4.3.1 Modul Applet Java untuk aplikasi	56
4.3.2 Modul Applet Java untuk presentasi	57
4.3.3 Modul Applet Java untuk kendali	59
4.4 Uji Coba	61
BAB V KESIMPULAN DAN SARAN	68
5.1 Kesimpulan	68
5.2 Saran	69
DAFTAR PUSTAKA	
LAMPIRAN	
LAMPIRAN-A PEMAKAIAN PROGRAM APPLET JAVA	
LAMPIRAN-B FILE STRUKTUR.MTTW	
LAMPIRAN-C STORE PROCEDURE APLIKASI MTTW	

DAFTAR GAMBAR

Gambar 2.1 Model sebuah SIE [MCL93, 587]	6
Gambar 2.2 Tingkatan pemakai SIPINTAS [ABD95a, 470]	9
Gambar 2.3 Arsitektur SIPINTAS [ABD95a, 471]	10
Gambar 2.4 Model MCSIR [ABD95b,7]	12
Gambar 2.5 Proses masing-masing komponen MCSIR	13
Gambar 2.6 Model klien pelayan	14
Gambar 2.7 Perbandingan kompilasi program Java dan tradisional [LEM96,8]	18
Gambar 2.8 Applet Java, HTML dan <i>browser</i>	20
Gambar 2.9 Tahapan Applet Java di-load oleh <i>browser</i>	21
Gambar 2.10 <i>Bytecode verifier</i> [GOS95, 58]	22
Gambar 2.11 Pemberian parameter ke Applet Java	23
Gambar 3.1 Komunikasi antar <i>applet</i> pada <i>page</i> yang sama	27
Gambar 3.2 Komunikasi antar <i>applet</i> pada <i>page</i> yang berbeda	28
Gambar 3.3 Komunikasi Applet Swith Table dan Applet Change	30
Gambar 3.4 Data yang diperoleh Applet Chart beserta parameter URL-nya	31
Gambar 3.5 Hierarki <i>level</i> untuk tiap <i>page</i>	32
Gambar 3.6 Perbedaan Previous dan Next Level dengan <i>backward</i> dan <i>forward</i>	34
Gambar 3.7a <i>Page</i> pindah ke <i>level</i> bawah dan <i>list</i> bertambah	35
Gambar 3.7b <i>Page</i> pindah ke <i>level</i> bawah dan <i>list</i> tetap	36
Gambar 3.7c <i>Page</i> pindah ke <i>level</i> bawah dan <i>list</i> berkurang	36
Gambar 3.8a Perpindahan <i>page</i> ke <i>level</i> atas dan <i>list</i> tetap	37
Gambar 3.8b Perpindahan <i>page</i> ke <i>level</i> atas dan <i>list</i> berkurang	37
Gambar 3.9a Perpindahan <i>page</i> pada <i>level</i> sejajar dan <i>list</i> tetap	38
Gambar 3.9b Perpindahan <i>page</i> pada <i>level</i> sejajar dan <i>list</i> berkurang	38
Gambar 3.10 Hierarki beberapa <i>page</i> acuan utama dan daerahnya masing-masing ..	40
Gambar 3.11 Perpindahan <i>page</i> acuan utama dalam Table Map	42

Gambar 3.12 Komunikasi antar <i>applet</i> aplikasi	43
Gambar 3.13 Komunikasi antar <i>applet</i> kendali	44
Gambar 4.1 Perhitungan mahasiswa yang belum dan sudah lulus	46
Gambar 4.2 Peta <i>page</i> MTTW	49
Gambar 4.3 <i>Frame</i> dalam aplikasi MTTW	51
Gambar 4.4 Applet Java Kendali	52
Gambar 4.5 Applet Java Aplikasi dan Presentasi	53
Gambar 4.6a <i>Bar Chart</i>	54
Gambar 4.6b <i>Line Chart</i>	54
Gambar 4.6c <i>Pie Chart</i>	55
Gambar 4.7 <i>Home page</i> MTTW	61
Gambar 4.8 <i>Page</i> peta mahasiswa	62
Gambar 4.9a mahasiswa yang sudah lulus	63
Gambar 4.9b Grafik <i>Bar Chart</i>	63
Gambar 4.10 Kendali Back To Table	64
Gambar 4.11 Kendali Previous dan Next Level	65
Gambar 4.12 Kendali Back To Sipintas	66
Gambar 4.13 Kendali Table map	67

DAFTAR TABEL

Tabel 3.1 Perbandingan Previous dan Next Level dengan <i>backward</i> dan <i>forward</i>	33
Tabel 4.1 Data mahasiswa yang sudah lulus	47
Tabel 4.2 Bobot angkatan	48



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG MASALAH

Kebutuhan manusia akan informasi pada saat ini semakin terasa penting. Informasi yang beragam jenis tersebut dapat diperoleh dari berbagai sumber melalui suatu sistem jaringan komputer yang mencakup seluruh dunia. Sistem jaringan ini disebut Internet. Sumber informasi yang disediakan oleh Internet dapat diperoleh dari berbagai tempat baik lokal, nasional maupun internasional.

Penggunaan Internet yang semakin luas dewasa ini menyebabkan semakin dibutuhkannya suatu program yang dapat berjalan di atas jaringan Internet. Program ini diharapkan dapat berjalan di masing-masing klien pemakai Internet sehingga eksekusi yang dilakukan tidak tergantung lagi dari jenis pelayan yang digunakan oleh pemakai Internet tersebut. Program yang mampu melaksanakan tugas ini adalah Applet Java.

Pemanfaatan Applet Java saat ini semakin luas khususnya untuk aplikasi-aplikasi yang digunakan oleh para eksekutif. Eksekutif tersebut menggunakan jaringan Internet yang ada untuk keperluan pengambilan berbagai sumber informasi. Sumber informasi tersebut terbagi atas sumber informasi yang terstruktur dan tak terstruktur. Sumber informasi yang terstruktur adalah sumber informasi yang berasal dari suatu pelayan basis data relasional. Sumber informasi yang tak terstruktur adalah sumber informasi

yang berasal dari selain pelayan basis data relasional misalnya : pelayan HTTP, pelayan Lotus Notes, serta pelayan NNTP.

Masalah-masalah seperti waktu yang terlalu sedikit, sumber informasi yang diterima terlalu banyak dan keputusan yang diambil harus cepat dilakukan menyebabkan para eksekutif membutuhkan suatu program yang dapat mengatasi masalah ini. Program tersebut diharapkan akan mempermudah para eksekutif dalam mengolah kembali data dan memilah data yang penting saja. Applet Java dapat mengatasi masalah ini karena Applet Java dapat berjalan di klien tempat para eksekutif berada. Pemanfaatan Applet Java ini tidak hanya terbatas pada aplikasi khusus yang dibuat untuk para eksekutif, tetapi dapat juga dibuat untuk modul-modul yang akan dipakai oleh aplikasi-aplikasi para eksekutif secara umum. Dengan dimanfaatkannya Applet Java ini, permasalahan-permasalahan yang dihadapi oleh para eksekutif tersebut diharapkan dapat dipecahkan.

1.2 TUJUAN

Tugas akhir ini bertujuan untuk merancang dan mengimplementasikan pemanfaatan Applet Java untuk prototipe *Multisources Client Server Information Retrieval* (MCSIR) dan aplikasi Sistem Informasi Pimpinan Universitas (SIPINTAS). Aplikasi SIPINTAS adalah aplikasi yang menggunakan layanan-layanan yang ada pada MCSIR. Pemanfaatan Applet Java untuk aplikasi SIPINTAS adalah spesifik untuk aplikasi yang bersangkutan. Pemanfaatan Applet Java untuk prototipe MCSIR dibuat umum agar dapat dipergunakan oleh aplikasi yang memanfaatkan layanan MCSIR.

1.3 PEMBATASAN MASALAH

Arsitektur SIPINTAS memiliki berbagai macam komponen yang salah satunya adalah komponen MCSIR. Komponen MCSIR terdiri atas tiga bagian yaitu bagian presentasi, kendali dan gerbang (*gateway*). Pemanfaatan Applet Java yang dikembangkan untuk prototipe MCSIR adalah pada bagian presentasi dan kendali.

Prototipe aplikasi SIPINTAS yang akan diuji cobakan adalah aplikasi Mahasiswa Tidak Tepat Waktu (MTTW) untuk Ketua Jurusan. Aplikasi ini diterapkan dalam bentuk *page Hypertext Markup Language (HTML)*. Data yang diperoleh berasal dari basis data Sybase. Pemanfaatan Applet Java untuk aplikasi SIPINTAS dibuat khusus untuk aplikasi MTTW.

1.4 METODE PENELITIAN

Metode-metode penelitian yang dipakai adalah sebagai berikut :

A. Mencari sumber informasi bahan-bahan awal tugas akhir.

Kegiatan ini mencakup mempelajari konsep tentang Applet Java, apa yang dapat dilakukan oleh *applet* dan keterbatasannya. Mempelajari konsep tentang *Hypertext Transfer Protocol (HTTP)*, serta mengetahui bagaimana cara kerja *browser* beserta *Hypertext Markup Language (HTML)*.

B. Mempelajari konsep Sistem Informasi Pimpinan Universitas (SIPINTAS).

Metode ini mencakup mempelajari konsep SIE dan SIPINTAS, bagaimana model Arsitektur SIPINTAS, mempelajari konsep MCSIR (*Multisources*

Client-Server Information Retrieval), dan bagian-bagian dalam MCSIR yaitu bagian presentasi, kendali dan gerbang.

C. Analisis dan Perancangan Applet Java.

Pada tahap ini merancang pemanfaatan Applet Java untuk aplikasi MTTW, pemanfaatan Applet Java untuk bagian presentasi dan kendali dalam MCSIR. Applet Java untuk aplikasi MTTW adalah Applet Java yang dapat mengolah kembali sumber informasi di klien. Applet Java pada bidang presentasi adalah Applet Java yang merepresentasikan kembali data dalam bentuk grafik dan Applet Java kendali adalah Applet Java yang mengendalikan *page* HTML aplikasi MTTW.

D. Implementasi dan Uji Coba

Aplikasi SIPINTAS yang digunakan adalah aplikasi (MTTW) untuk Ketua Jurusan. Tahap ini merancang MTTW dalam bentuk *page* HTML.

1.5 SISTEMATIKA PENULISAN

Bab I menjelaskan tentang latar belakang permasalahan yang ada, tujuan penulisan tugas akhir, pembatasan permasalahan, metode penelitian yang dipakai serta sistematika penulisan tugas akhir.

Bab II menjelaskan tentang konsep Sistem Informasi Eksekutif yang akan diterapkan dalam arsitektur SIPINTAS. Menerangkan konsep Sistem Informasi Pimpinan Universitas (SIPINTAS) serta komponen arsitekturnya. Bab ini diakhiri dengan menjelaskan konsep tentang Applet Java.

Bab III Menjelaskan analisa kebutuhan dan rancangan Applet Java untuk aplikasi SIPINTAS dan prototipe MCSIR. Pada bab ini diterangkan tentang *applet* untuk aplikasi, *applet* untuk presentasi dan *applet* untuk kendali.

Bab IV menerangkan tentang penerapan dan uji coba prototipe aplikasi MTTW untuk Ketua Jurusan. Dijelaskan juga tentang implementasi Applet Java dalam aplikasi MTTW. Pada bab ini dipaparkan algoritma modul-modul utama yaitu algoritma modul Applet Java untuk aplikasi MTTW, algoritma modul Applet Java untuk bidang presentasi, dan algoritma modul Applet Java untuk kendali. Bab ini ditutup dengan uji coba Applet Java aplikasi, presentasi dan kendali.

Bab V menerangkan tentang kesimpulan dan saran. Kesimpulan-kesimpulan berdasarkan dari apa yang telah dilakukan selama pembuatan tugas akhir ini serta saran-saran yang dianjurkan untuk melengkapi tugas akhir ini agar lebih sempurna.

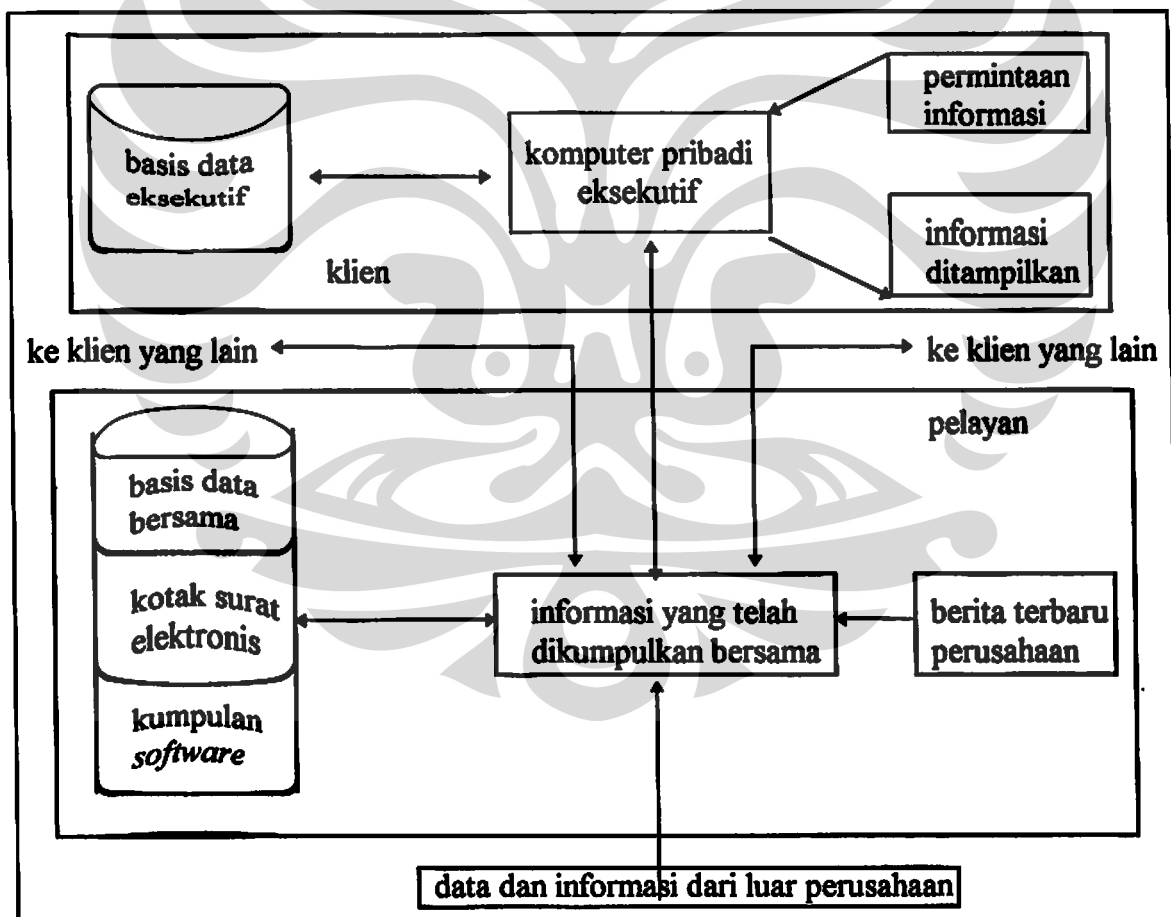
BAB II

SISTEM INFORMASI EKSEKUTIF

2.1 KONSEP SISTEM INFORMASI EKSEKUTIF

Sistem Informasi Eksekutif (SIE) adalah sebuah sistem yang menyediakan layanan informasi kepada eksekutif tentang unjuk kerja keseluruhan perusahaan [MCL93, 586]. Informasi tersebut harus mudah diperoleh dan dapat disediakan untuk seluruh lapisan struktur perusahaan secara terinci.

Konfigurasi komputer yang dipakai oleh SIE adalah sebagai berikut :



Gambar 2.1 Model sebuah SIE [MCL93, 587]

Pada gambar 2.1, komputer pribadi eksekutif dianggap sebagai klien. Konfigurasi komputer pribadi eksekutif terdiri dari tempat penyimpanan basis data eksekutif itu sendiri. Basis data yang telah didapat berisi data dan informasi yang telah diproses oleh pelayan. Pelayan juga menyediakan kotak surat elektronik dan informasi yang berasal dari lingkungan sekitar perusahaan, serta berita-berita aktual dalam perusahaan yang terdapat saat itu.

Pada kenyataannya para eksekutif membangun sebuah SIE berdasarkan beberapa konsep manajemen yang ada sehingga SIE yang dibuat dapat mempermudah para eksekutif dalam mengambil keputusan. Tiga konsep management yang akan dibahas disini yaitu *critical success factors*, *management by exception*, dan *mental models* [MCL93, 588]. Berikut akan dijelaskan ketiga konsep manajemen tersebut :

1. *Critical Success Factors*.

Critical Success Factors (CSF) adalah salah satu aktifitas perusahaan yang sangat kuat pengaruhnya dalam menentukan kemampuan perusahaan dalam mencapai tujuannya [MCL93, 90]. D. Ronald Daniel dari McKinsey & Company mengatakan, "Untuk beberapa aktifitas kunci perusahaan (CSF), menentukan sukses atau gagalnya berbagai macam perusahaan untuk mencapai tujuannya dan aktifitas itu bervariasi dari satu organisasi yang satu dengan yang lainnya." Misalnya perusahaan mobil menentukan CSF-nya dari rancangannya, distribusi penjualan yang efektif dan pengetatan harga industrinya. Perusahaan asuransi melihat CSF-nya dari pengembangan manajemen personal untuk para agennya, serta inovasi untuk produk-

produk asuransi yang terbaru. CSF ini yang nantinya akan dimonitor oleh para eksekutif untuk menentukan apakah perusahaan sudah berjalan baik dan sesuai dengan tujuannya.

2. *Management By Exception*

Management By Exception (MBE) adalah suatu cara dimana para eksekutif melihat suatu kejadian yang diluar kontrol aktifitas yang semestinya sudah ditetapkan [MCL93, 89]. SIE yang dibangun ini diharapkan dapat menampilkan MBE kepada para eksekutif sehingga mempermudah para eksekutif dalam mengambil keputusannya.

3. *Mental Models*

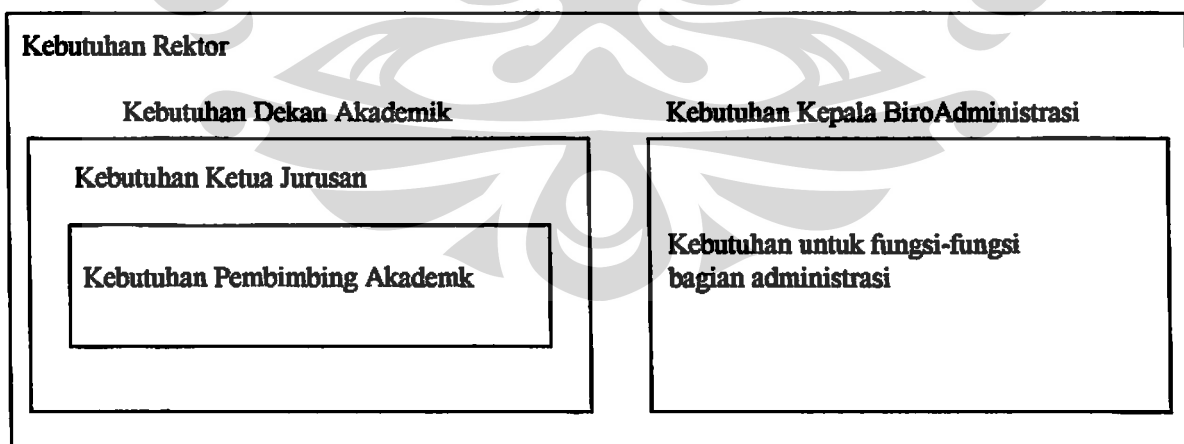
Peraturan yang utama dalam SIE adalah menampilkan data yang telah disaring dan dipadatkan sehingga data dan informasi yang ditampilkan benar-benar data dan informasi yang sangat bermanfaat bagi para eksekutif. Data yang telah disaring dan dipadatkan, kemudian ditampilkan dinamakan sebuah *Mental Model* [MCL93, 588]. John Rockart dan David De long dari MIT mengatakan bahwa, "*Mental Models* membuat para eksekutif dapat mengambil kesimpulan dan prediksi, mengerti fenomena yang terjadi, memutuskan aksi apa yang akan diambil dan mengontrol eksekusinya, serta berpengalaman terhadap kejadian-kejadian yang diwakilkan oleh *Mental Models* itu."

2.2 SISTEM INFORMASI PIMPINAN UNIVERSITAS

Sistem Informasi Pimpinan Universitas adalah suatu sistem SIE yang dikembangkan untuk lingkungan universitas. Tujuan dikembangkannya proyek Sistem Informasi Pimpinan Universitas (SIPINTAS) adalah membangun suatu Sistem Informasi Eksekutif untuk Universitas Indonesia [ABD95a, 467]. Sistem ini nantinya diharapkan tidak hanya dikembangkan di Universitas Indonesia, tetapi juga bisa dijadikan acuan untuk dipakai pada universitas-universitas yang lainnya di Indonesia.

SIPINTAS mengintegrasikan beberapa konsep Teknologi Informasi dan interaksi antara manusia dengan komputer, kelompok pengambilan keputusan, pengambilan beberapa sumber informasi, representasi data dan untuk lingkungan multimedia.

Pemakai SIPINTAS adalah Rektor dengan Pembantu Rektor, Dekan, Ketua Jurusan, Pembimbing Akademik dan Kepala biro bagian administrasi yang digambarkan sebagai tingkatan-tingkatan berikut :

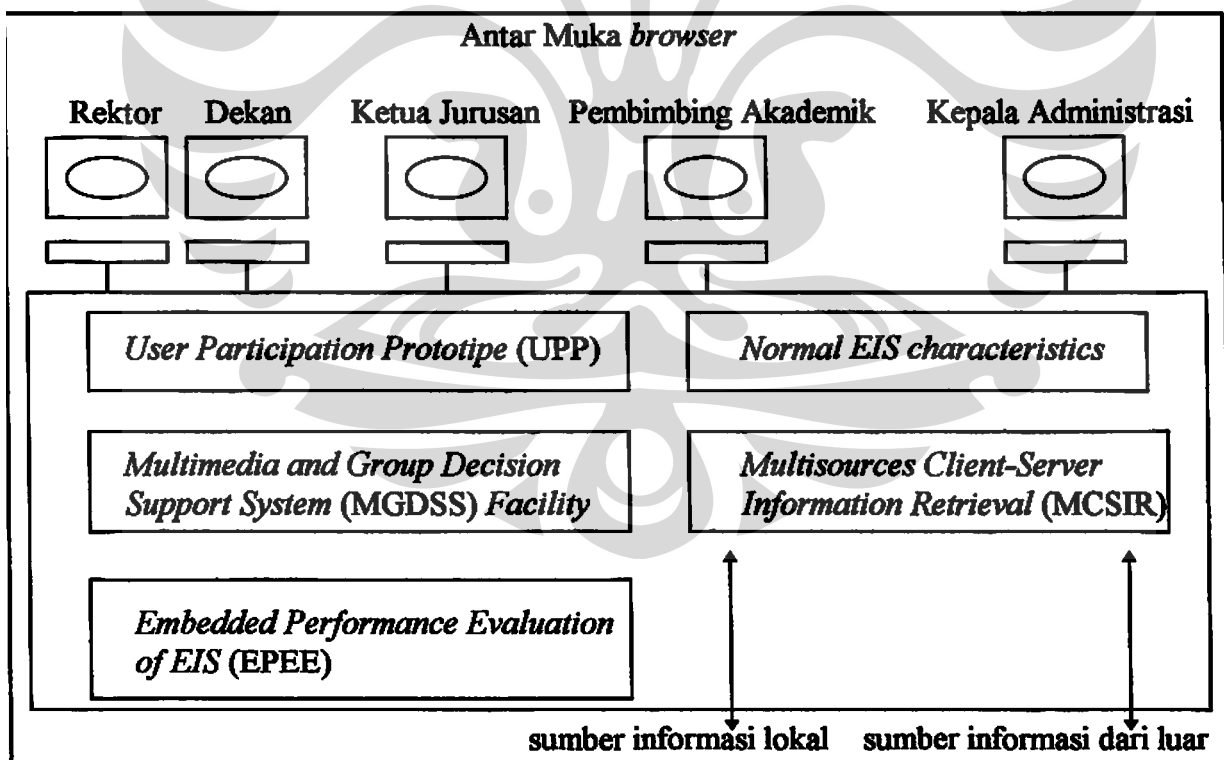


Gambar 2.2 Tingkatan pemakai SIPINTAS [ABD95a, 470]

Informasi yang dibutuhkan oleh Pembimbing Akademik merupakan informasi yang sangat terinci. Pemakai dengan tingkatan yang lebih tinggi dalam struktur organisasi membutuhkan informasi yang semakin global. Pemakai dengan tingkatan yang lebih tinggi ini dapat memperoleh informasi yang lebih terinci melalui teknik *drill down investigation* (penelusuran lebih lanjut). Teknik ini dimungkinkan karena setiap bagian dari struktur/tingkatan pemakai SIPINTAS merupakan sub bagian dari struktur/tingkatan pemakai SIPINTAS di atasnya.

2.2.1 Arsitektur SIPINTAS

Arsitektur SIPINTAS terdiri atas beberapa komponen yang digambarkan sebagai berikut :



Gambar 2.3 Arsitektur SIPINTAS [ABD95a, 471]

Keterangan gambar :

1. *User Participation Prototipe (UPP)*

Dalam perkembangan SIPINTAS, kita perlu merancang prototipenya terlebih dahulu. Prototipe ini lalu diuji cobakan kepada beberapa pemakai untuk menentukan sesuai atau tidaknya rancangan prototipe yang telah jadi. Partisipasi pemakai untuk sistem ini sangat penting mengingat rancangan yang dibuat harus mendefinisikan kebutuhan-kebutuhan pemakai.

2. *Multimedia and Group Decision Support System (MGDSS) Facility.*

Kebutuhan akan pemakaian multimedia semakin terasa penting untuk pembuatan berbagai aplikasi khususnya aplikasi Sistem Informasi Eksekutif. SIPINTAS akan mengimplementasikan unsur multimedia dan sistem pengambilan keputusan bersama ini untuk masa yang akan datang.

3. *Embedded Performance Evaluation of EIS (EPEE)*

Dalam merancang prototipe SIPINTAS, pemakai tidak hanya berpartisipasi dalam pembuatannya saja, tetapi pemakai juga memberikan evaluasi untuk hasil unjuk kerja. Dalam memberikan hasil evaluasi ini, pemakai diijinkan untuk memberikan komentarnya langsung dalam sistem dan membuat kebutuhan pemakai yang baru. Hal ini akan mempermudah pengembangan aplikasi yang sedang dibuat.

4. *Normal EIS Characteristic*

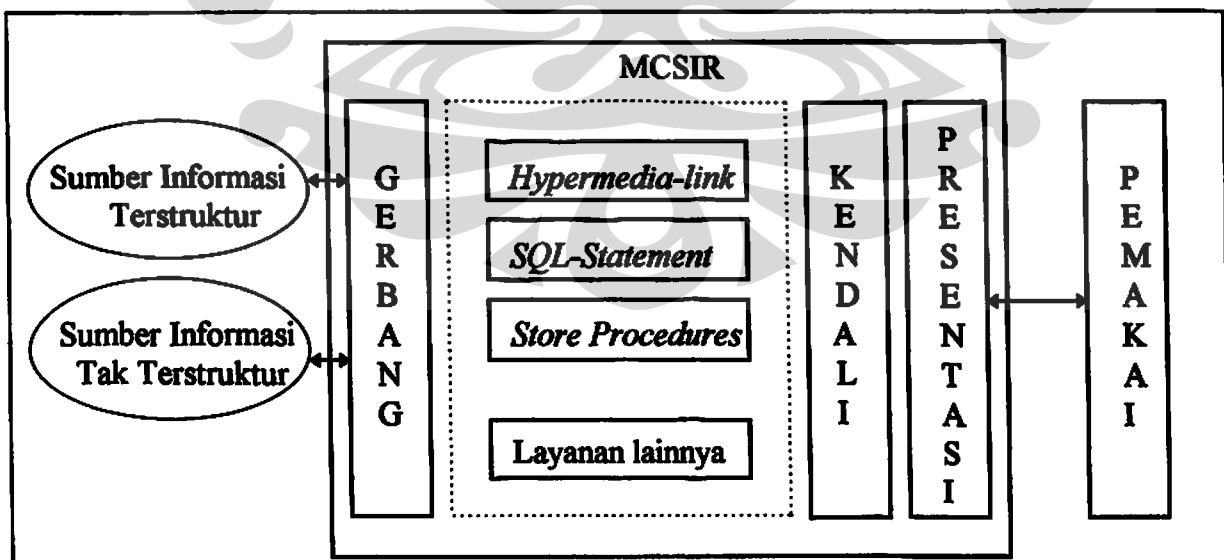
Bagian ini menyediakan berbagai fasilitas karakteristik SIE, seperti penyediaan surat elektronik, kamus elektronik dan sebagainya.

5. *Multisources Client-Server Information Retrieval (MCSIR)*

Pada saat sekarang ini dibutuhkan suatu sistem informasi yang dapat mengintegrasikan berbagai macam sumber informasi. Sumber Informasi yang didapat berasal dari sumber informasi yang terstruktur (berasal dari suatu basis data) dan sumber informasi yang tak terstruktur (bukan berasal dari suatu basis data). Sistem ini akan bekerja dan berkomunikasi lewat jaringan komputer yang telah ada. Jaringan komputer yang dipakai tidak hanya bertingkat lokal dan nasional, tetapi juga bersifat internasional. Sistem ini disebut sebagai *Multisources Client Server Information Retrieval (MCSIR)* . Informasi yang diambil tidak hanya bersifat lokal, tetapi juga berasal dari luar seperti *World Wide Web (WWW)*.

2.2.2 Model MCSIR

Berikut akan digambarkan model MCSIR :

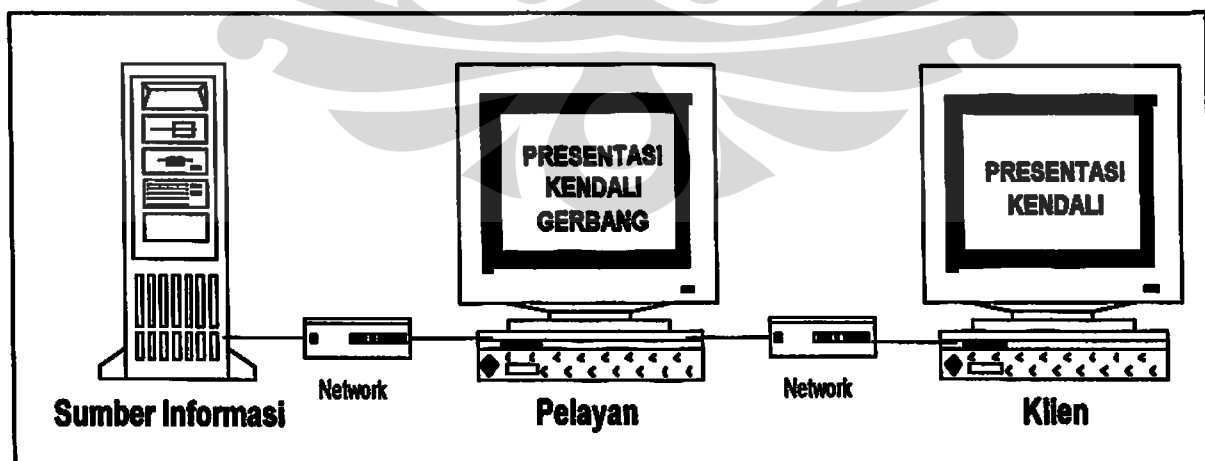


Gambar 2.4 Model MCSIR [ABD95b, 7]

MCSIR terdiri atas layanan-layanan yang tersedia seperti *SQL statement*, *Store Procedure*, *Hypermedia link* dan layanan lainnya yang nanti akan diintegrasikan ke dalam MCSIR.

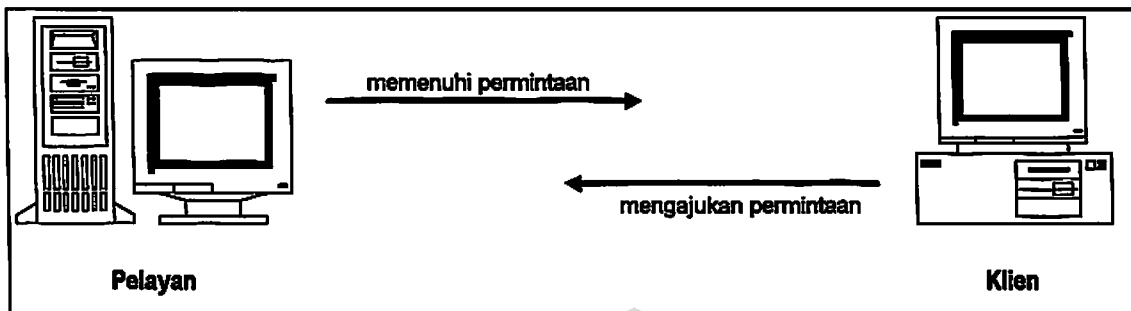
Model MCSIR disebut sebagai *Toaster Model*. *Toaster Model* adalah sebuah model di mana suatu modul dapat dipasang atau diambil dari model tersebut tanpa mempengaruhi modul lainnya. Penerapan konsep ini berlaku pada MCSIR, yaitu bila suatu layanan ditambahkan atau dikurangi pada model MCSIR, tidak akan mempengaruhi layanan yang lainnya.

Komponen-komponen yang ada pada MCSIR yaitu komponen presentasi, komponen kendali dan komponen gerbang (*gateway*). Komponen-komponen ini berada di pelayan. Ketiga komponen ini akan saling bekerja sama sesuai dengan fungsinya masing-masing. Sebagian proses komponen kendali dan presentasi akan berjalan di klien dan sebagian lainnya berjalan di pelayan, sedangkan proses komponen gerbang akan berjalan di pelayan sepenuhnya. Berikut adalah gambaran proses masing-masing komponen MCSIR :



Gambar 2.5 Proses masing-masing komponen MCSIR

Adapun model dari klien dan pelayan digambarkan sebagai berikut :



Gambar 2.6 Model klien pelayan

Model tersebut diilustrasikan sebagai berikut : program pada sisi yang satu mengajukan permintaan ke program pada sisi yang lain dan permintaan tersebut dipenuhi oleh program pada sisi yang lain. Program yang mengajukan permintaan disebut sebagai klien dan program yang memenuhi/membalas permintaan tersebut disebut sebagai pelayan [COM91, 483]. Komunikasi selalu dimulai dari sisi klien dan tak pernah dimulai dari sisi pelayan.

Berikut adalah skenario antara klien dan pelayan :

1. Pelayan diaktifkan pada suatu komputer dan menunggu permintaan yang datang dari klien.
2. Permintaan klien biasanya adalah hasil dari masukan pemakai dan dikirim ke pelayan melalui jaringan komputer yang menghubungkan antara klien dan pelayan.
3. Pelayan melayani permintaan yang dikirim oleh klien.
4. Ketika permintaannya telah selesai dan pelayan mengirim balik hasilnya ke klien, maka pelayan menunggu lagi permintaan berikutnya yang akan dikirim oleh klien.

2.2.2.a Komponen presentasi

Komponen yang merepresentasikan data kepada pemakai disebut komponen presentasi. Komponen inilah yang akan menampilkan data yang didapat dari layanan-layanan yang ada seperti *SQL statement*, *Store Procedure*, dan *Hypermedia link* ke pemakai dalam bentuk format yang diinginkan. Tugas akhir ini mengimplementasikan komponen presentasi ke dalam bentuk grafik sehingga komponen ini dapat dipergunakan oleh aplikasi yang memanfaatkan layanan MCSIR.

2.2.2.b Komponen kendali

Komponen yang mengatur dan mengendalikan aplikasi SIPINTAS sehingga mempermudah pemakai untuk memilah sumber informasi yang didapatnya disebut komponen kendali. Tugas akhir ini mengimplementasikan juga komponen kendali. Komponen kendali akan mengatur *page HTML* aplikasi yang memanfaatkan layanan MCSIR.

2.3.2.c Komponen gerbang (*gateway*)

Komponen yang bertugas untuk mengambil sumber informasi adalah komponen gerbang. Komponen gerbang akan mengambil sumber informasi yang terstruktur (yang berasal dari basis data) dan sumber informasi yang tak terstruktur (yang berasal dari WWW).

2.3 KONSEP APPLLET JAVA

Berikut akan diterangkan beberapa konsep yang menyangkut perancangan Applet Java yang akan dibuat.

2.3.1 Internet

Pengertian *internetting* atau *internetworking* adalah hubungan antara beberapa komputer dalam suatu jaringan yang ada tanpa mempedulikan berbagai macam perangkat keras yang digunakan, sehingga komputer yang satu dengan yang lainnya dapat saling berkomunikasi [COM91, 1]. Pengertian Internet dengan huruf besar berarti hubungan beberapa jaringan yang ada beserta gerbangnya (*gateways*) dengan menggunakan protokol *Transmission Control Protocol (TCP) / Internet Protocol (IP)* [COM91, 493].

Komunikasi antara beberapa jaringan yang ada beserta perangkat kerasnya dapat terjalin karena adanya alamat yang diberikan dalam Internet. Alamat yang diberikan dalam Internet mengacu pada alamat jaringan dan alamat perangkat keras yang berada pada jaringan tersebut. Pengalamatan ini memungkinkan untuk komunikasi antara komputer pada jaringan yang satu dengan komputer pada jaringan yang lainnya.

Pengalamatan dengan menggunakan *Hypertext Transfer Protocol (HTTP)* adalah sebagai berikut :

`http://[nama host/alamat IP]/[direktori yang dituju]/[nama program]`

2.3.2 Java

Java adalah bahasa pemrograman yang dirancang khusus untuk pemakaian Internet. Berikut akan dibahas beberapa karakteristik yang dimiliki oleh bahasa Java [GOS95,12].

1. Berorientasi objek

Pemrograman dengan bahasa Java menyerupai pemrograman dengan bahasa C/C++. *Object Oriented* yang dipakai dalam Java hampir sama dengan *Object Oriented C++* dengan perbedaannya di Java tidak mempunyai *multiple inheritance* (banyak *parent*). Penggunaan *Object Oriented* di Java disebabkan pemakaian sistemnya yang terdistribusi dan seringnya dilakukan *message passing* antar objek.

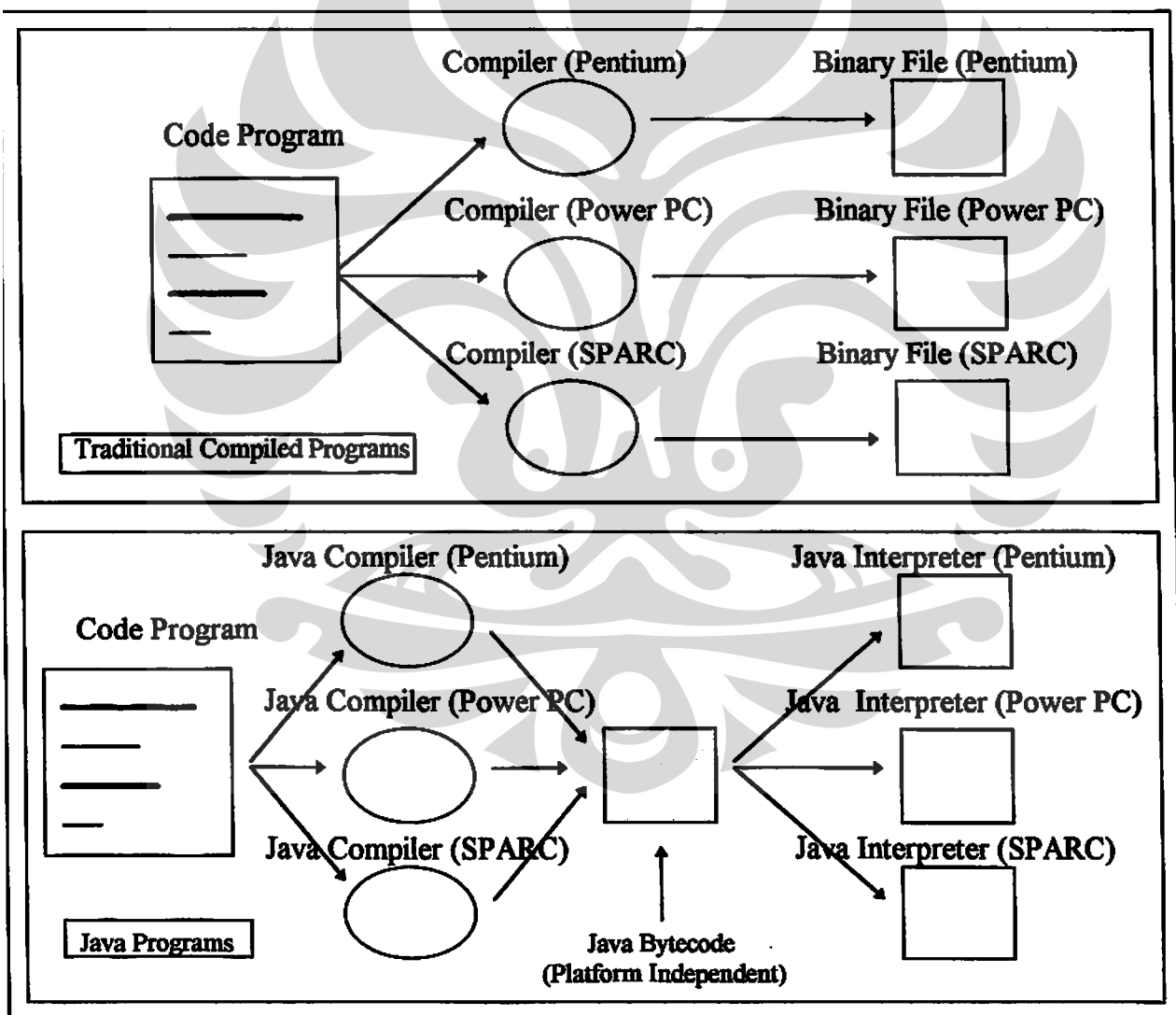
2. Handal dan aman

Bahasa Java dirancang sebagai bahasa yang handal. Keandalan ini dibuktikan antara lain dengan manajemen memorinya yang otomatis dan tidak adanya *pointers arithmetic* lagi (dalam C/C++ masih ada).

Bahasa Java digunakan dalam sistem yang terdistribusi di mana masalah keamanan merupakan masalah yang penting. Aplikasi-aplikasi yang dibuat dalam bahasa Java dijamin aman bila melewati jaringan komputer yang ada karena adanya *verifier* sebelum dilakukan interpretasi bahasa tersebut.

3. Dapat dijalankan dimana saja (*platform independent*)

Salah satu karakteristik bahasa Java adalah *platform independent*. Hal ini dimungkinkan dengan dibentuknya suatu *bytecode* terlebih dahulu. *Bytecode* adalah bahasa mesin yang tak tergantung jenis perangkat kerasnya pada saat *code* ini dibentuk. *Bytecode* inilah yang nantinya dapat dipindahkan ke berbagai macam perangkat keras. Berikut akan digambarkan perbedaan *coding* antara bahasa biasa dan Java :



Gambar 2.7 Perbandingan kompilasi program Java dan tradisional [LEM96,8]

4. Unjuk kerja yang tinggi

Bytecode dapat diinterpretasikan langsung (*on the fly*) ke mesin yang bersangkutan. Hal ini akan menyebabkan unjuk kerja yang tinggi pada bahasa Java.

5. *Interpreted*

Hasil dari kompilasi yang berbentuk *bytecode* harus dijalankan dengan menggunakan Interpreter Java. Interpreter Java berbeda-beda dari mesin yang satu dengan mesin yang lainnya.

6. *Multithreading*

Multithreading adalah suatu teknik yang digunakan agar *thread-thread* dapat berjalan secara bersamaan (*concurrent threads*). Konsep ini diterapkan dalam bahasa Java yang sering melakukan *thread* secara bersamaan dalam jaringan Internet.

Program Java terbagi atas dua yaitu Aplikasi Java (*stand alone application*) dan Applet Java. Berikut akan diterangkan bagian tersebut.

2.3.2.a Aplikasi Java

Aplikasi Java adalah program yang berdiri sendiri (*stand alone application*). Aplikasi Java dapat dijalankan langsung seperti program yang lainnya. Perbedaan Aplikasi Java dengan program lainnya yaitu Aplikasi Java dapat dijalankan pada mesin yang mana saja (*platform independent*), tetapi dengan syarat Aplikasi Java ini harus dikompilasi

terlebih dahulu ke dalam bentuk *bytecode*. *Bytecode* inilah yang nantinya akan diinterpretasikan ke mesin yang bersangkutan.

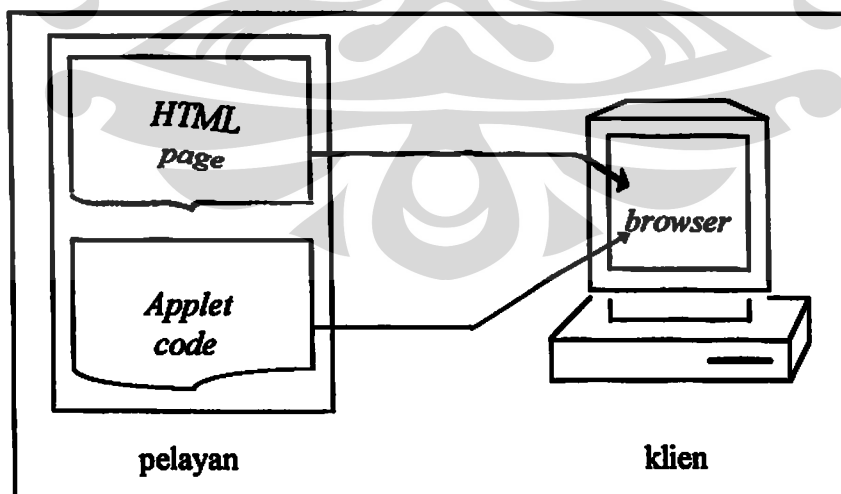
2.3.2.b Applet Java

Applet Java adalah program yang berjalan di *browser*. Applet Java harus terlebih dahulu di kompilasi ke dalam bentuk *bytecode*. *Bytecode* inilah yang nantinya akan dijalankan oleh *browser*. Applet Java tersebut dapat diperoleh dari lokal maupun dari pelayan.

Applet Java yang dijalankan di *browser* harus terlebih dahulu didefinisikan di dalam *tag HTML*. *Tag HTML* ini berbentuk :

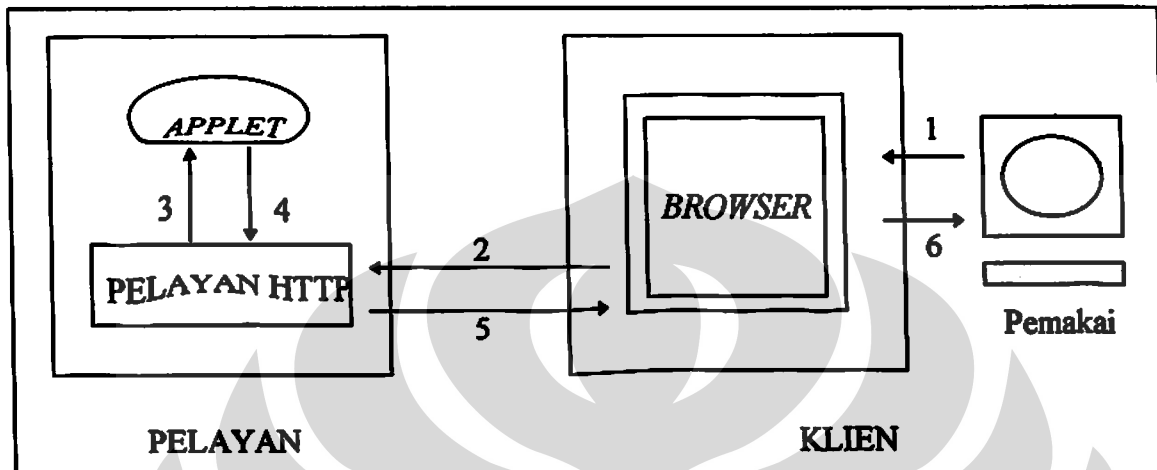
```
<applet codebase=[URL bytecode] code=[bytecode]
width=[area lebar applet] height=[area panjang applet]>
</applet>
```

Berikut akan digambarkan bagaimana Applet Java dan HTML di *loading* oleh *Browser* :



Gambar 2.8 Applet Java, HTML dan *browser*

Applet Java yang berasal dari pelayan didapat oleh *browser* melalui tahapan-tahapan sebagai berikut :

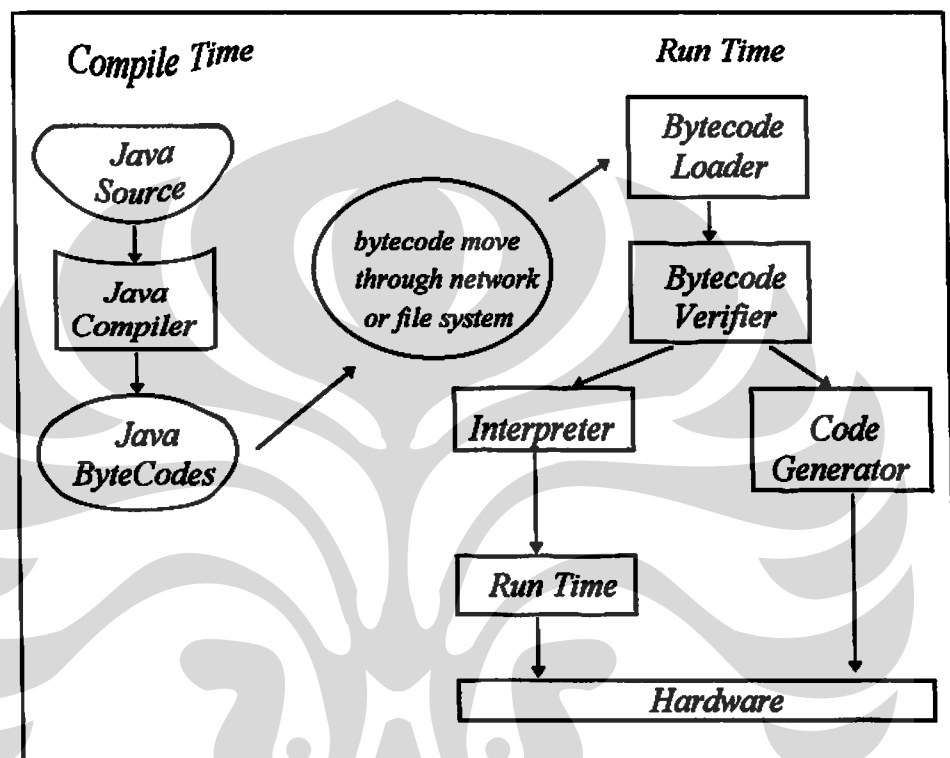


Gambar 2.9 Tahapan Applet Java di-load oleh *browser*

Keterangan Gambar :

1. Pemakai meminta *browser* untuk menampilkan *applet* dengan mengirimkan pesan ke alamat pelayan HTTP yang dituju.
2. *Browser* meneruskan pesan itu ke alamat pelayan HTTP.
3. Pelayan HTTP mencari *applet* yang dituju.
4. Pelayan HTTP mengambil *applet* tersebut.
5. Pelayan HTTP mengirim *applet* ke *browser* di klien.
6. *Browser* lalu memverifikasi dan menterjemahkan *applet* tersebut .

Applet Java yang akan dijalankan ini harus terlebih dahulu diverifikasi *bytecode*-nya oleh *bytecode verifier*. Berikut akan digambarkan bagaimana proses jalannya *bytecode verifier* tersebut :



Gambar 2.10 *Bytecode verifier* [GOS95, 58]

2.3.2.b.i Parameter Applet Java

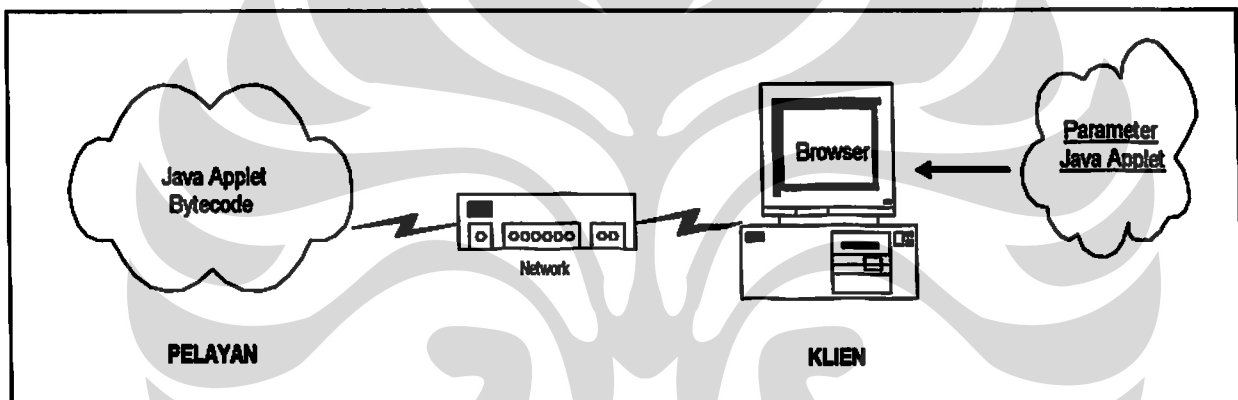
Applet Java juga bisa diberi parameter di mana parameter ini akan ditangkap oleh Applet Java di klien. Parameter yang diberi ke Applet Java harus didefinisikan terlebih dahulu ke dalam *tag* HTML. Pendefinisian parameter ini harus berada dalam pendefinisian *tag* Applet Java. *Tag*

pendefinisian parameter yang diberi ke Applet Java adalah sebagai berikut :

```
<applet ..... >
<param name={variabel yang didefinisikan}
value={harga yang diberi ke variabel tsb} >
. . .
</applet>
```

Gambar ini melukiskan Applet Java menerima parameternya di

klien :



Gambar 2.11 Pemberian parameter ke Applet Java

2.3.2.b.ii Keterbatasan Applet Java

Kekhususan sifat Applet Java yang diambil *bytecode*-nya dari pelayan dan dijalankan di klien, mengakibatkan faktor keamanan menjadi semakin penting. Untuk alasan ini maka terdapat keterbatasan-keterbatasan terhadap Applet Java [COR96,320]. Keterbatasan-keterbatasan tersebut yaitu :

1. Applet Java tak dapat menjalankan program lokal yang *executable*.
2. Applet Java tak dapat berkomunikasi dengan pelayan yang lain, selain tempat Applet Java itu berasal.
3. Applet Java tak dapat membaca ataupun menulis *file system* di mana ia berjalan.
4. Applet Java tak dapat mengetahui informasi lokal komputer tempat ia berjalan, kecuali untuk mengetahui versi Java yang digunakan serta nama dan versi sistem operasi lokal komputer. Applet Java tak dapat mengetahui nama pemakai komputer lokal dan alamat surat elektronisnya.

BAB III

ANALISA KEBUTUHAN DAN RANCANGAN APPLLET JAVA

3.1 ANALISA KEBUTUHAN

Model MCSIR memiliki arsitektur jaringan klien-pelayan. Klien digunakan untuk interaksi antar muka dengan pemakai dan pelayan digunakan sebagai tempat pengambilan berbagai sumber informasi.

Jaringan klien-pelayan yang digunakan oleh MCSIR adalah jaringan Internet. MCSIR membutuhkan pelayan HTTP dan *browser* agar dapat berjalan di atas jaringan Internet. Pelayan HTTP digunakan sebagai penghubung antara klien dan pelayan. *Browser* digunakan sebagai perangkat lunak yang berjalan di klien untuk menampilkan aplikasi.

Penggunaan akan jaringan Internet ini mengakibatkan dibutuhkannya suatu program yang dapat berjalan di atas jaringan Internet. Program yang dapat berjalan di atas jaringan Internet saat ini adalah Applet Java. Applet Java akan berjalan di klien melalui pelayan HTTP dan akan ditampilkan oleh *browser* di klien. Applet Java ini diharapkan dapat berinteraksi dengan pemakai di klien.

Applet Java yang dibuat dalam tugas akhir ini mencakup komunikasi antar *applet* pada *page* yang sama maupun pada *page* yang berbeda. Komunikasi antar *applet* pada *page* yang sama dibutuhkan karena dalam melakukan tugasnya diperlukan kerja sama antara *applet* yang satu dengan *applet* lainnya. *Applet* yang satu bertugas

untuk menangkap data dan menampilkannya kembali ke *browser* sementara *applet* yang lainnya bertugas untuk mengolah data tersebut.

Komunikasi antar *applet* pada *page* yang berbeda dibutuhkan karena *applet* pada *page* yang satu tidak akan terpengaruh oleh *applet* pada *page* yang lainnya. *Page* yang satu akan terus berpindah-pindah *page*, sementara *page* yang lainnya akan tetap. *Applet* pada *page* yang satu akan mengirim informasi ke *applet* lainnya pada *page* yang berbeda.

3.2 KOMUNIKASI ANTAR APPLLET

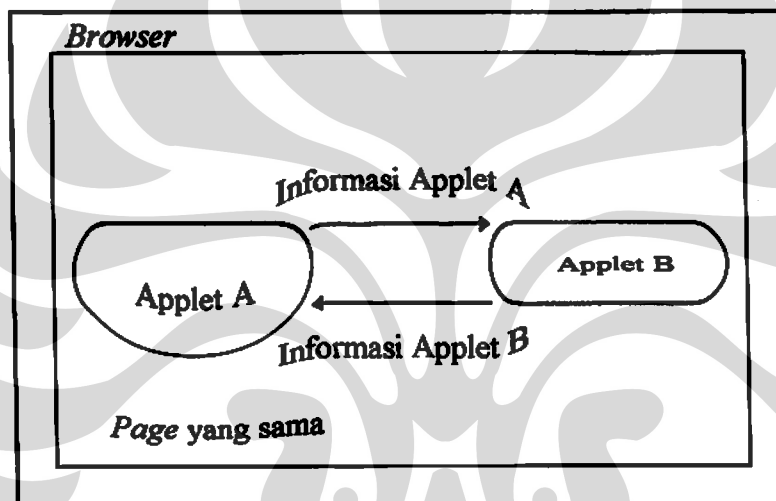
Komunikasi antar *applet* adalah hubungan antara *applet* yang satu dengan *applet* lainnya. Dalam komunikasi ini akan terjadi proses pertukaran informasi. Komunikasi antar *applet* tersebut dapat terjadi pada *page* yang sama maupun pada *page* yang berbeda. Berikut ini akan diterangkan komunikasi antar *applet* pada *page* yang sama dan pada *page* yang berbeda.

3.2.1 Komunikasi antar *applet* pada *page* yang sama

Komunikasi antar *applet* pada *page* yang sama dapat dilakukan dengan cara *applet* yang satu mencari *applet* lainnya pada *page* yang sama. Jika proses pencarian ini berhasil, maka proses pertukaran informasi dapat dimulai. *Applet* tersebut dapat dicari dengan cara menggunakan perintah sebagai berikut :

```
getAppletContext().getApplets().
```

Setelah *applet* tersebut diperoleh, maka dapat dilakukan pertukaran informasi (*message passing*) antar *applet*. Pertukaran informasi ini dapat dilakukan untuk variabel maupun metode yang ada pada *applet* tersebut. Variabel dan metode yang dipertukarkan harus didefinisikan terlebih dahulu ke dalam bentuk *public* agar dapat diakses oleh *applet* lainnya. Berikut akan digambarkan bagaimana *applet* yang satu dapat berkomunikasi dengan *applet* yang lain pada *page* yang sama.



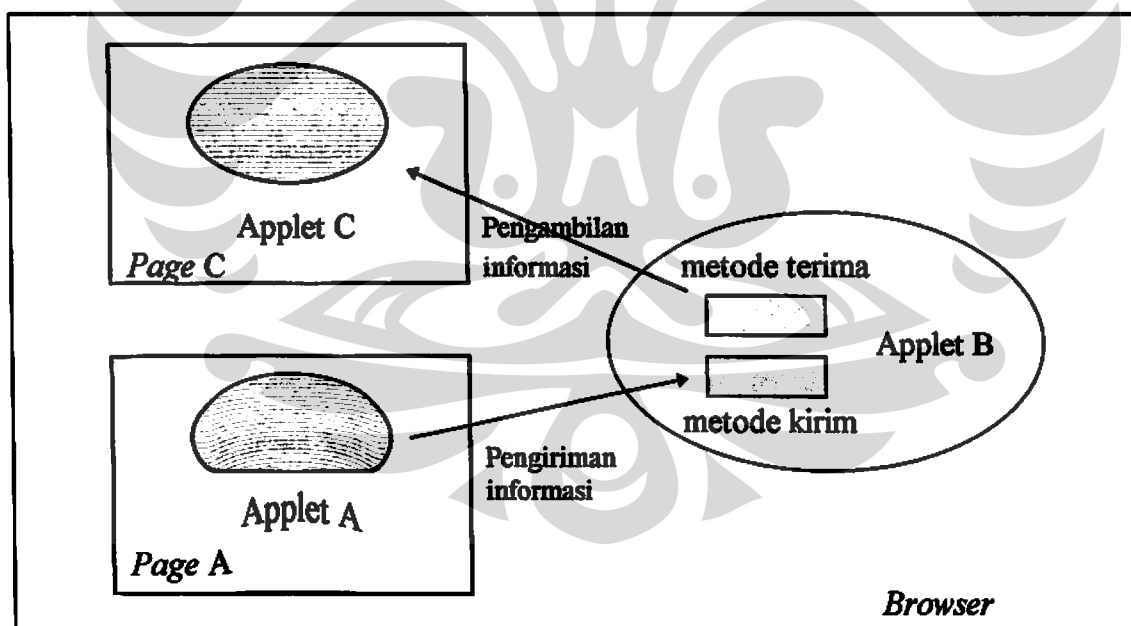
Gambar 3.1 Komunikasi antar *applet* pada *page* yang sama

3.2.2 Komunikasi antar *applet* pada *page* yang berbeda

Komunikasi antar *applet* pada *page* yang berbeda dapat dilakukan dengan cara menggunakan *applet* perantara. *Applet* pada *page* yang satu mengirim informasi ke *applet* perantara dan *applet* yang lainnya pada *page* yang berbeda akan mengambil informasi ini dari *applet* perantara. Pengiriman dan pengambilan informasi dari *applet* perantara ini dilakukan dengan cara

memanggil *applet* perantara beserta variabel atau metode yang terdapat di dalamnya.

Misalkan *applet* perantara tersebut bernama Applet B dengan metode yang digunakannya yaitu metode kirim(argumen yang dikirim) untuk pengiriman informasi dan metode terima() untuk penerimaan informasi. Applet A dan Applet C saling berkomunikasi dimana Applet A berada pada *page* A dan Applet C berada pada *page* C. Informasi yang dikirim oleh Applet A ke Applet C adalah informasi XXX. Applet A mengirim informasi tersebut dengan cara B.kirim(XXX) dan Applet C mengambil informasi ini dengan cara B.terima(). Berikut akan digambarkan bagaimana *applet* tersebut saling berkomunikasi :



Gambar 3.2 Komunikasi antar *applet* pada *page* yang berbeda

3.3 RANCANGAN APPLLET JAVA

Berikut akan diterangkan pemanfaatan Applet Java yang akan dirancang untuk kebutuhan aplikasi dan prototipe MCSIR.

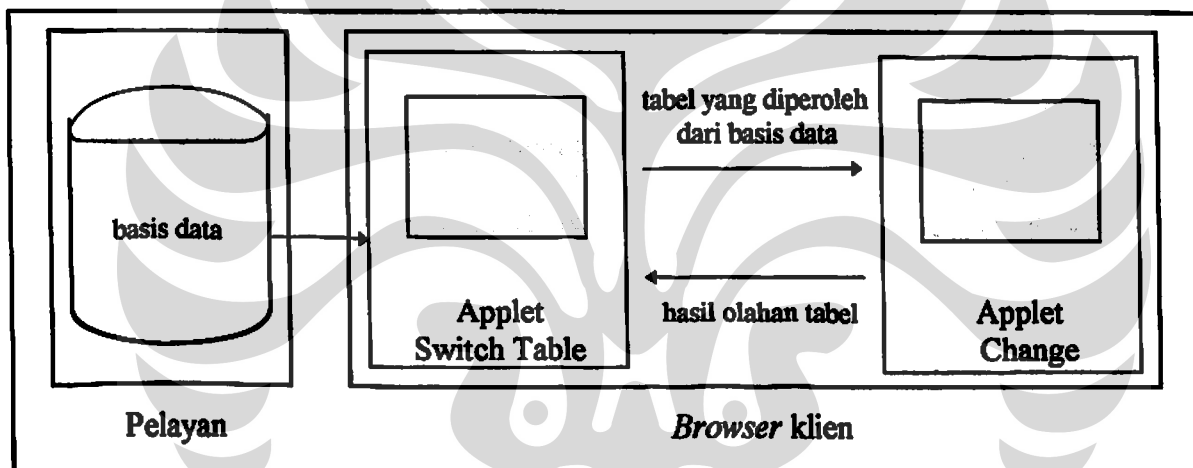
3.3.1 Pemanfaatan Applet Java untuk aplikasi

Pemanfaatan Applet Java untuk aplikasi adalah pemanfaatan Applet Java untuk aplikasi SIPINTAS. Tujuan dibuatnya Applet Java untuk aplikasi ini adalah untuk mengolah kembali tabel yang diperoleh dari basis data ke dalam bentuk olahan tabel lainnya langsung di klien. Tabel yang diperoleh dari basis data tersebut adalah tabel mahasiswa yang belum lulus yang akan diolah menjadi tabel mahasiswa yang sudah lulus oleh Applet Java aplikasi. Alasan pengolahan tabel langsung di klien adalah waktu eksekusi yang lebih cepat dibandingkan dengan proses pengambilan tabel lagi pada pelayan basis data.

Permasalahan yang timbul di sini adalah keterbatasan Applet Java dalam mengambil data tabel tersebut langsung di klien. Applet Java tak dapat menangkap data tabel tersebut dan juga tak dapat menampilkannya langsung di *browser*. Karena keterbatasan ini maka Applet Java yang dibuat terbagi atas dua yaitu Applet Java yang bertugas dalam menangkap dan menampilkan data tabel yang berasal dari basis data melalui argumen yang diberikan dan Applet Java yang mengolah tabel tersebut. Applet Java yang bertugas dalam

menangkap dan menampilkan data tabel adalah Applet Switch Table. Applet Java yang bertugas untuk mengolah tabel tersebut adalah Applet Change.

Saat pertama kali dijalankan, Applet Switch Table akan menerima data tabel yang berasal dari basis data. Data tabel ini akan dikirim oleh Applet Switch Table ke Applet Change untuk diolah kembali. Hasil olahan ini kemudian akan dikirim balik oleh Applet Change ke Applet Switch Table untuk ditampilkan. Berikut akan digambarkan komunikasi antar Applet Switch Table dan Applet Change :



Gambar 3.3 Komunikasi Applet Swith Table dan Applet Change

Data tabel yang diperoleh dari basis data tersebut harus ditangkap satu persatu oleh Applet Switch Table, sampai Applet Switch Table akan dimuat sebanyak data tabel yang akan ditampilkan.

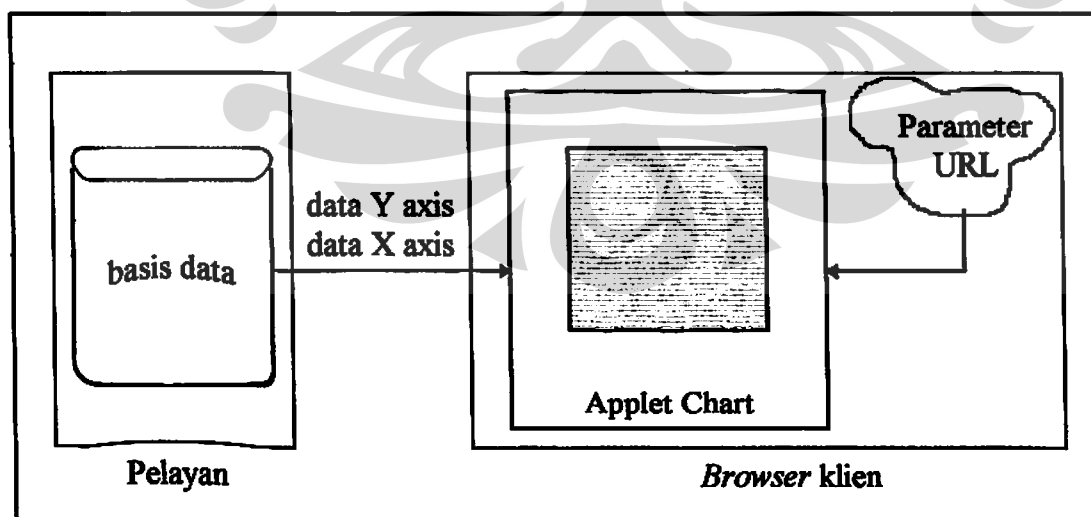
Komunikasi yang dilakukan antar Applet Switch Table dengan Applet Change adalah komunikasi antar *applet* pada *page* yang sama.

3.3.2 Pemanfaatan Applet Java untuk presentasi

Komponen MCSIR yang bertugas untuk menampilkan data yang telah diperoleh dari suatu sumber informasi ke pemakai adalah komponen presentasi. Pemanfaatan Applet Java untuk komponen presentasi MCSIR bertujuan untuk menampilkan data dalam bentuk grafik ke pemakai.

Applet Java yang bertugas dalam menampilkan data ke dalam bentuk grafik adalah Applet Chart. Applet Chart membutuhkan koordinat X axis dan Y axis untuk dapat menampilkan data tersebut ke dalam bentuk grafik. Koordinat X axis dan Y axis ini didapat langsung dari pelayan basis data.

Applet Chart juga membutuhkan *Uniform Resource Locator* (URL) grafik yang akan ditampilkan. URL ini bertujuan agar masing-masing grafik yang akan ditampilkan dapat di-link ke alamat yang diinginkan. Berikut adalah gambaran data yang akan didapat oleh Applet Chart dari pelayan basis data beserta parameter URL yang diberikan ke Applet Chart :



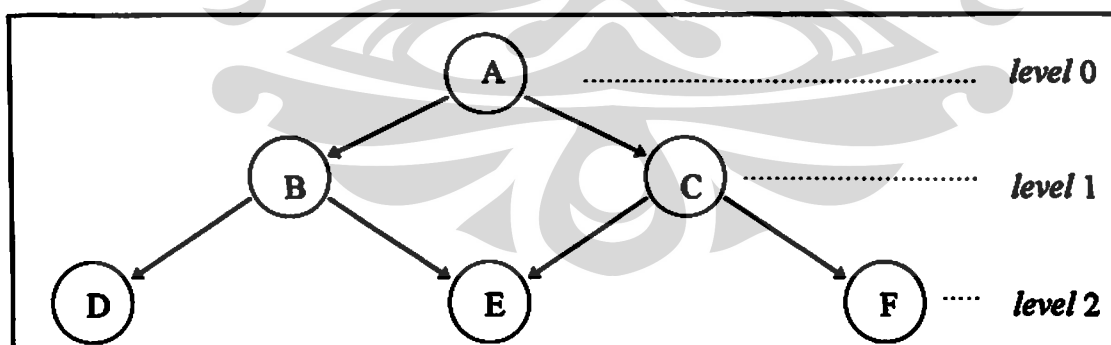
Gambar 3.4 Data yang diperoleh Applet Chart beserta parameter URL-nya

3.3.3 Pemanfaatan Applet Java untuk kendali

Pemanfaatan Applet Java untuk komponen kendali MCSIR dibuat dengan tujuan mengendalikan *page* aplikasi SIPINTAS. Pemanfaatan Applet Java untuk komponen kendali memakai struktur hierarki *page* aplikasi yang digunakan. Berikut akan dijelaskan kendali Previous dan Next Level, Home Page, Back To Table, Back To Sipintas, dan Table Map.

3.3.3.a Previous Level dan Next Level

Kendali Previous Level dan Next Level adalah salah satu kendali yang bertugas untuk mengendalikan *page* aplikasi berdasarkan hierarki *level*. Hierarki *level* adalah hierarki *page* yang ada pada aplikasi yang digunakan. Hierarki *page* tersebut adalah susunan *page* yang dibuat sehingga *page* pada *level* di atas mengacu pada *page level* di bawahnya. Berikut adalah gambaran hierarki *page* tersebut :



Gambar 3.5 Hierarki *level* untuk tiap *page*

Kendali ini bertujuan untuk mempermudah pemakai dapat berpindah-pindah *page* ke *level* di atasnya (Previous Level) atau ke *page level* di bawahnya (Next Level). Syarat utama penggunaan kendali ini yaitu *page* tersebut sudah pernah dikunjungi.

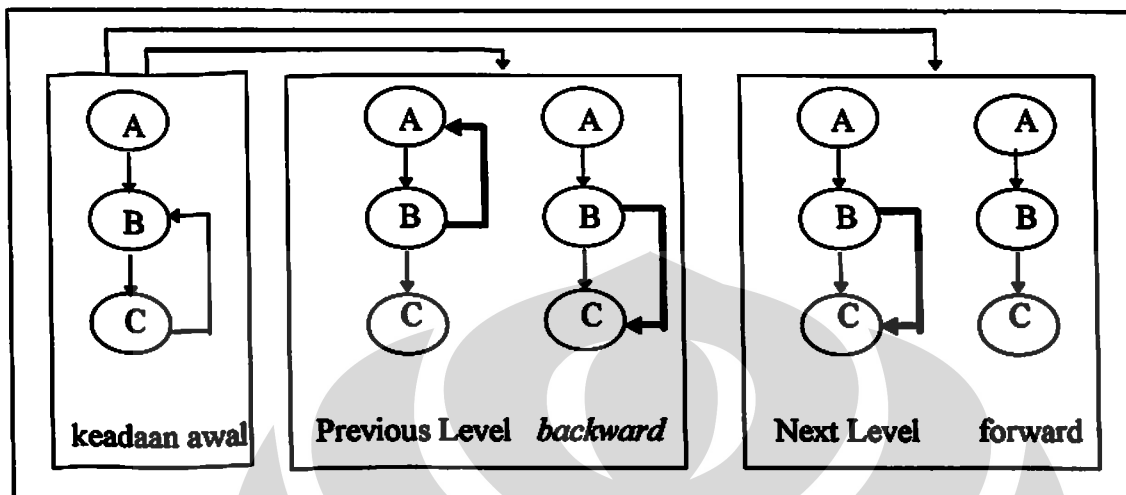
Applet yang digunakan untuk kendali ini adalah Applet Sender, Applet Receiver dan Applet Media. Applet Sender bertugas untuk mengirim URL ke Applet Receiver melalui Applet Media. Applet Receiver yang akan mengambil informasi URL ini dari Applet Media. Applet Media bertugas sebagai media perantara antara Applet Sender dan Applet Receiver. Penggunaan Applet Media disebabkan komunikasi ketiga *applet* ini menggunakan teknik komunikasi antar *applet* pada *page* yang berbeda.

Penggunaan kendali Previous Level dan Next Level berbeda dengan penggunaan *backward* dan *forward* yang ada pada tiap-tiap *browser*. Penjelasan berikut memberikan gambaran perbedaan-perbedaan yang utama antara Previous Level dan Next Level dengan *backward* dan *forward* :

<u>Prototipe Kendali</u>		<u>Browser</u>	
Previous Level	ke <i>level</i> di atasnya (<i>up level</i>)	<i>backward</i>	<i>page</i> sebelumnya
Next Level	ke <i>level</i> di bawahnya (<i>down level</i>)	<i>forward</i>	<i>page</i> sesudahnya

Tabel 3.1 Perbandingan Previous dan Next Level dengan *backward* dan *forward*

Gambar berikut adalah contoh perbedaan antara Previous Level dengan *backward* dan Next Level dengan *forward*.



Gambar 3.6 Perbedaan Previous dan Next Level dengan *backward* dan *forward*

Keterangan gambar :

Keadaan awal adalah mengunjungi *page* A, B, C lalu ke B lagi.

Pemakaian Previous Level akan membawa perpindahan ke *page* A (ke *level* di atasnya) sedangkan pemakaian *backward* menyebabkan perpindahan ke *page* C (*page* sebelumnya). Demikian juga pemakaian Next Level akan membawa perpindahan ke *page* C (ke *level* di bawahnya) sedangkan pemakaian *forward* tidak dapat dilakukan karena *page* B adalah *page* yang terakhir dikunjunginya.

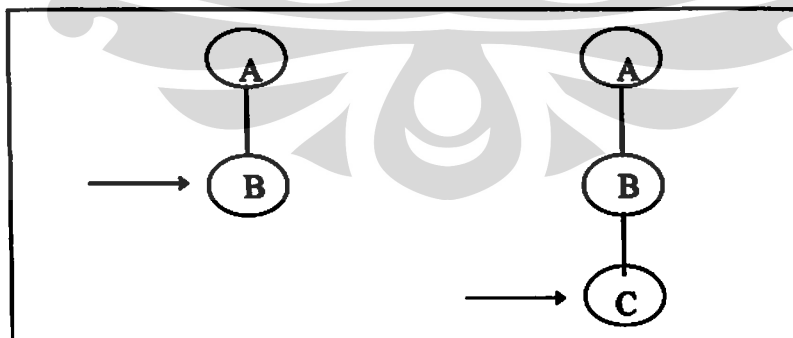
Hierarki *page* ini disimpan pada suatu *file* yang bernama *struktur.mttw*. *File* ini berada pada sisi pelayan. Isi *file* yang dibaca untuk kendali ini adalah pada pendefinisian kata *level*. Pada kata *level* di *file* *struktur.mttw* terdapat pendefinisian *level* tiap-tiap *page* aplikasi

sehingga Applet Media dapat mengetahui posisi *page* yang aktif saat itu berada pada *level* ke berapa dan akan dimasukkan ke dalam *list page* yang sudah ada. *Applet* yang bertugas untuk mentransfer *file struktur.mttw* dari sisi pelayan ke klien adalah Applet Media. Applet Media juga bertugas menjaga *list page* yang sudah dikunjungi. *List page* yang sudah dikunjungi akan berubah-ubah isinya sesuai dengan perpindahan *page* yang satu ke *page* yang lainnya. Berikut akan dijelaskan analisa yang akan terjadi pada suatu *list* bila *page* berpindah dari suatu *page* ke *page* lainnya.

3.3.3.a.i Perpindahan *page* ke *level* di bawahnya

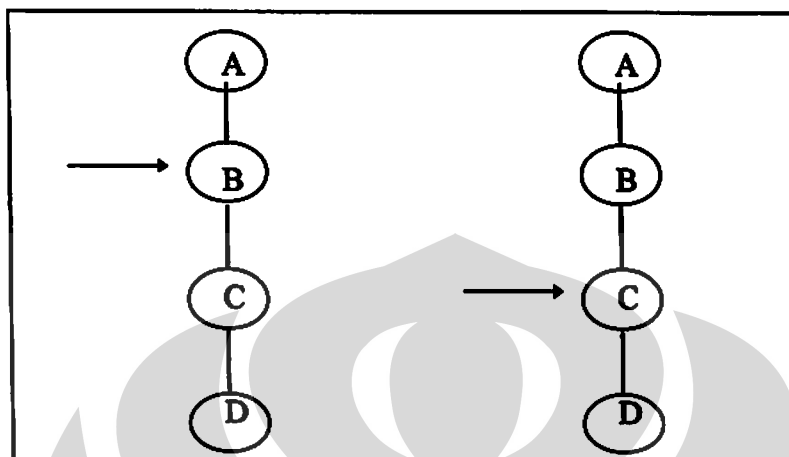
Perpindahan *page* ke *level* di bawahnya akan menyebabkan perubahan *list page* sebagai berikut :

1. Penambahan *list* yang disebabkan *page* tersebut baru dikunjungi pertama kalinya.



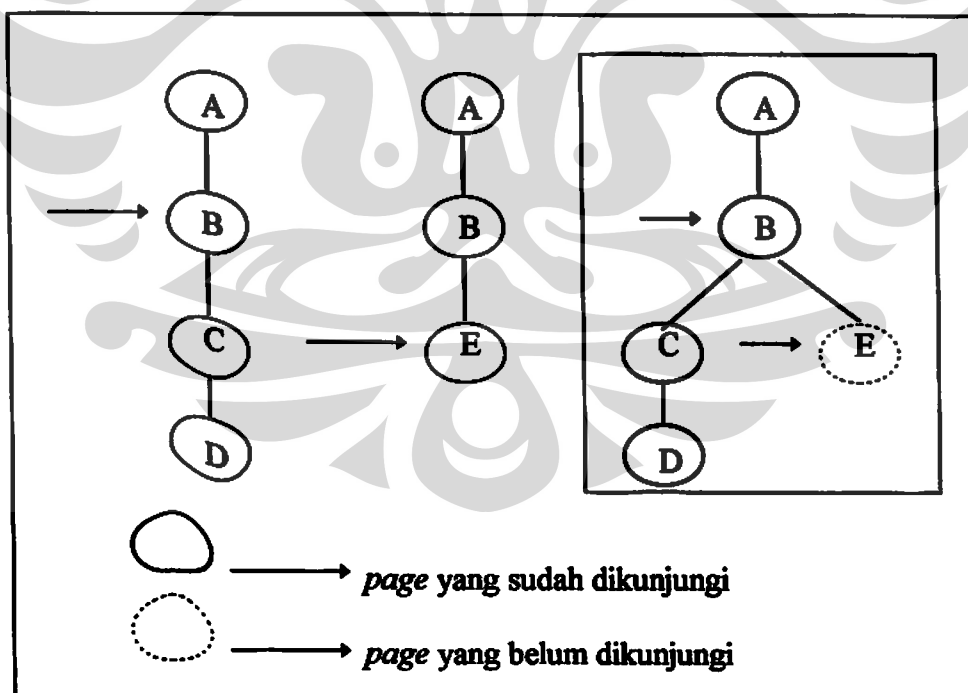
Gambar 3.7a *Page* pindah ke *level* bawah dan *list* bertambah

2. Perpindahan *page* ke *level* di bawahnya dimana *page* tersebut sudah pernah dikunjungi (ada dalam *list*).



Gambar 3.7b *Page* pindah ke *level* bawah dan *list* tetap

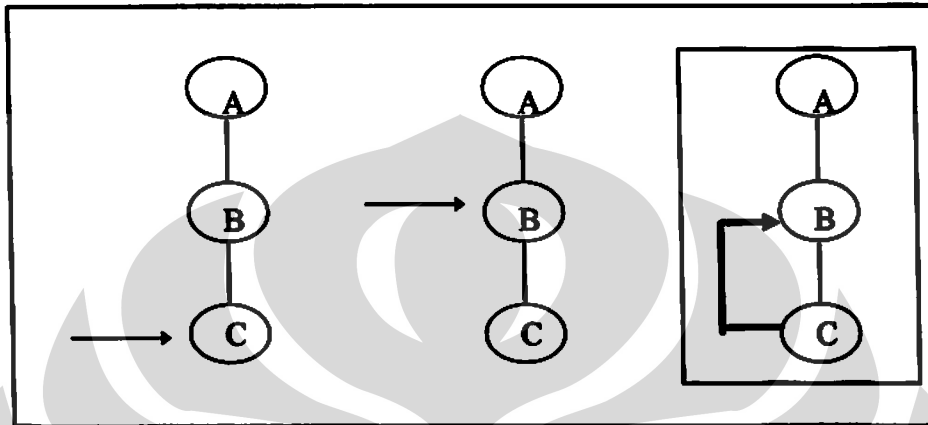
3. Perpindahan *page* ke *level* di bawahnya dimana *page* tersebut belum pernah dikunjungi.



Gambar 3.7c *Page* pindah ke *level* bawah dan *list* berkurang

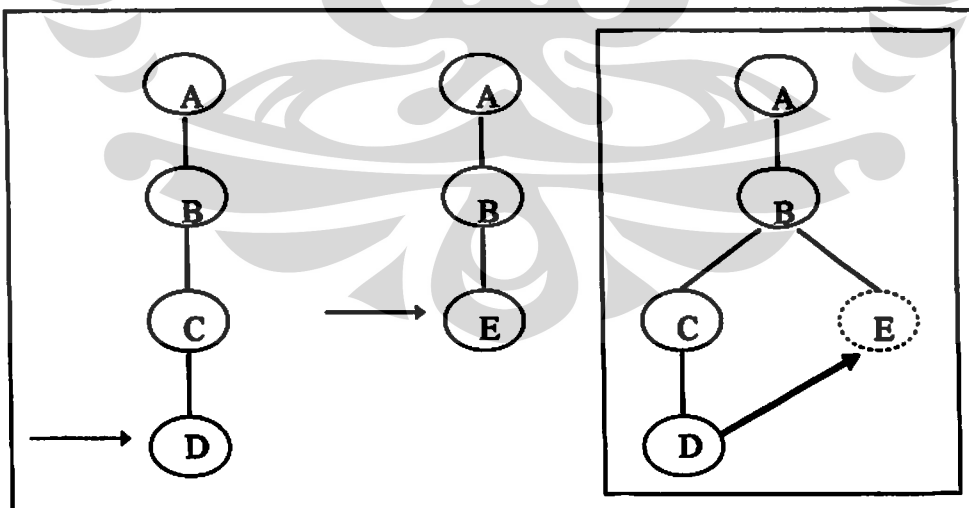
3.3.3.a.ii Perpindahan *page* ke *level* di atasnya

1. Perpindahan *page* ke *level* di atasnya dan *page* tersebut sudah pernah dikunjungi sebelumnya (ada dalam *list*).



Gambar 3.8a Perpindahan *page* ke *level* atas dan *list* tetap

2. Perpindahan *page* ke *level* di atasnya dan *page* tersebut belum pernah dikunjungi sebelumnya.

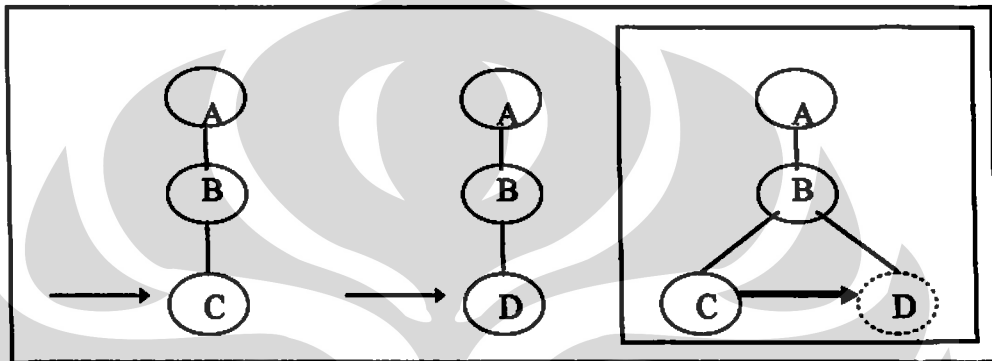


Gambar 3.8b Perpindahan *page* ke *level* atas dan *list* berkurang

3.3.3.a.iii Perpindahan *page* pada *level* yang sejajar

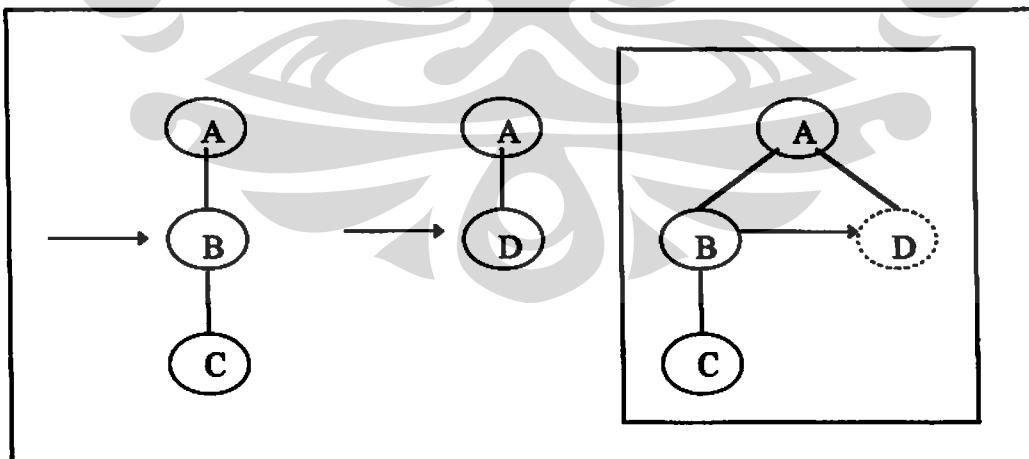
Perpindahan *page* ke *level* yang sejajar akan mengakibatkan perubahan *list page* sebagai berikut :

1. Perpindahan *page* pada *level* yang sejajar dan *list* berubah isinya. *List* tersebut tidak bertambah maupun berkurang.



Gambar 3.9a Perpindahan *page* pada *level* sejajar dan *list* tetap

2. Perpindahan *page* pada *level* yang sejajar dan isi *list* berubah serta menyebabkan *list page* yang lainnya hilang.



Gambar 3.9b Perpindahan *page* pada *level* sejajar dan *list* berkurang

3.3.3.b Home Page

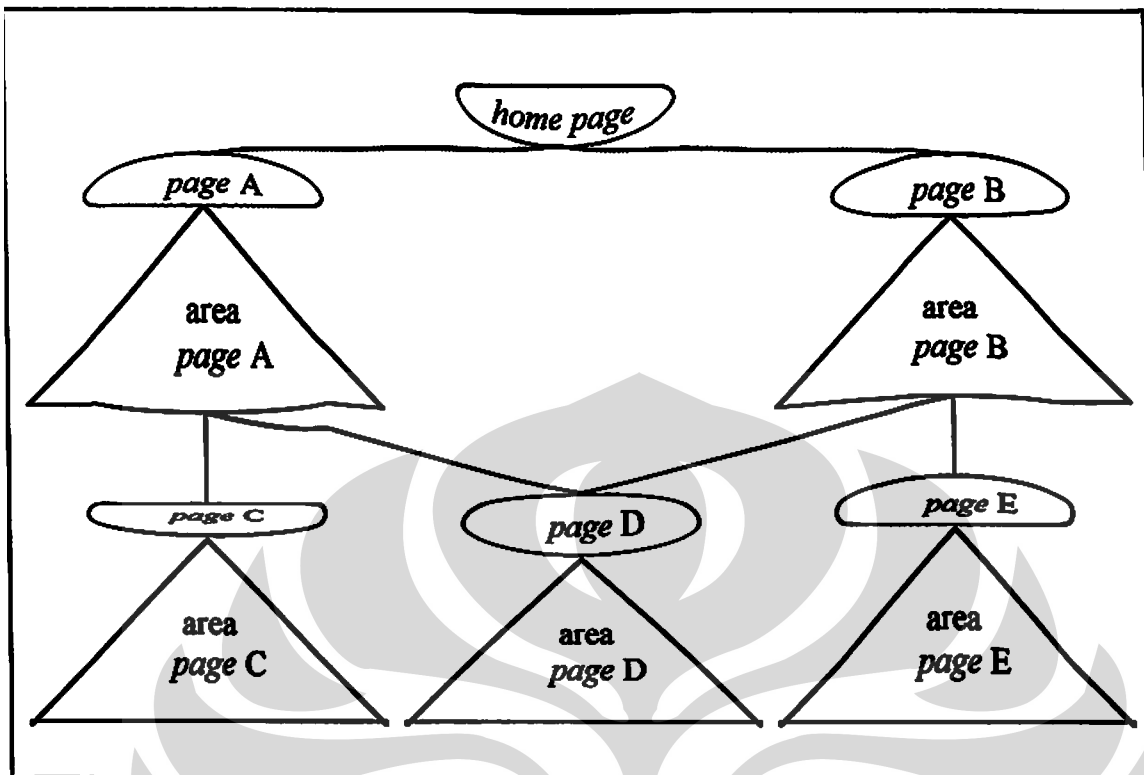
Kendali Home Page dibuat dengan tujuan agar si pemakai dapat kembali ke *page* utama aplikasi yang sedang digunakan. *Page* utama aplikasi adalah *page* yang pertama kali muncul pada saat aplikasi tersebut berjalan.

Applet yang bertugas dalam menjaga *page* utama adalah *Applet Media*. *Applet* yang bertugas mengirim URL *page* utama adalah *Applet Sender* dan *Applet* yang menampilkan kendali Home Page ini adalah *Applet Receiver*. *Applet Media* adalah yang bertugas membaca *file struktur.mttw* di pelayan. Pendefinisian URL *page* utama ini diletakkan pada *file struktur.mttw* pada kata *main*.

3.3.3.c Back To Table

Kendali Back To Table adalah kendali yang mengacu ke beberapa *page* yang berisi informasi yang dianggap penting dan global. *Page* tersebut mempunyai banyak acuan ke *page* lainnya.

Tujuan dibuatnya kendali Back To Table adalah agar pemakai dapat kembali secara dinamis ke *page* acuan utama yang telah didefinisikan. Kendali ini mempermudah pemakai untuk kembali ke *page* lainnya yang diacu oleh *page* acuan utama tersebut. Berikut gambar beberapa *page* acuan utama beserta daerahnya masing-masing :



Gambar 3.10 Hierarki beberapa *page* acuan utama dan daerahnya masing-masing

Page A, B, C, D, E adalah *page* acuan utama yang mempunyai daerahnya masing-masing. Area masing-masing *page* adalah area *page* yang diacu oleh *page* acuan utama tersebut. *Page* acuan utama yang satu dapat diacu oleh salah satu *page* yang berada pada area *page* acuan utama lainnya.

Penggunaan kendali Back To Table pada area *page* acuan utama C akan kembali ke *page* acuan utama C karena *page* acuan utama yang paling dekat adalah *page* acuan utama C bukan *page* acuan utama A. Jadi penggunaan kendali ini bertingkat sesuai dengan area *page* acuan utama mana saat itu berada.

Applet yang bertugas menjaga *list page* acuan utama adalah *Applet Media*. *Applet* ini akan membaca *file struktur.mttw* pada pendefinisian kata *subindeks*. *Applet* yang bertugas mengirim alamat URL *page* acuan utama tersebut ke *Applet Media* adalah *Applet Sender*. *Applet* yang menerima alamat URL *page* acuan utama dan yang mengaktifkan kendali *Back To Table* adalah *Applet Receiver*.

3.3.3.d *Back To Sipintas*

Bila *page* aplikasi SIPINTAS mengacu ke *page* lainnya di luar hierarki *page* aplikasi SIPINTAS sendiri, maka pemakai dapat kembali ke *page* aplikasi SIPINTAS dengan menggunakan kendali *Back To Sipintas*.

Tujuan dibuatnya kendali ini agar pemakai dapat kembali ke *page* yang ada pada hierarki aplikasi SIPINTAS. *Page* yang terakhir dikunjungi oleh pemakai aplikasi yang akan diacu oleh kendali *Back To Sipintas*.

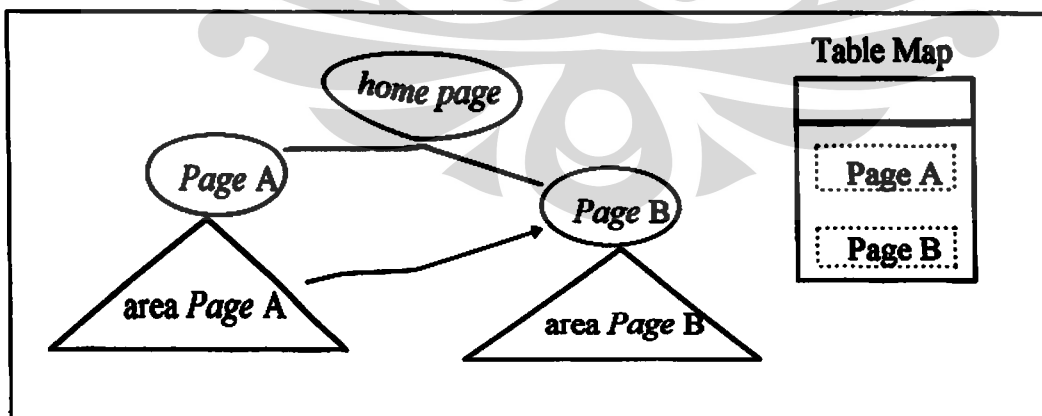
Applet yang menjalankan kendali ini adalah *Applet Sender*, *Applet Receiver* dan *Applet Media*. *Applet Media* inilah yang akan menyimpan URL *page* yang terakhir kali dikunjungi.

3.3.3.e Table Map

Kendali Table Map adalah kendali yang berisi kumpulan *page* acuan utama yang digunakan oleh kendali Back To Table. Jika pemakai menggunakan kendali Back To Table, maka pemakai hanya dapat balik ke *page* acuan utama yang sudah pernah dikunjunginya. Dengan adanya kendali Table Map ini maka pemakai dapat *link* langsung ke *page* acuan utama lainnya yang belum pernah dikunjungi.

Tujuan dibuatnya kendali Table Map agar pemakai dapat *link* ke *page* acuan utama lainnya tanpa harus mengunjunginya terlebih dahulu.

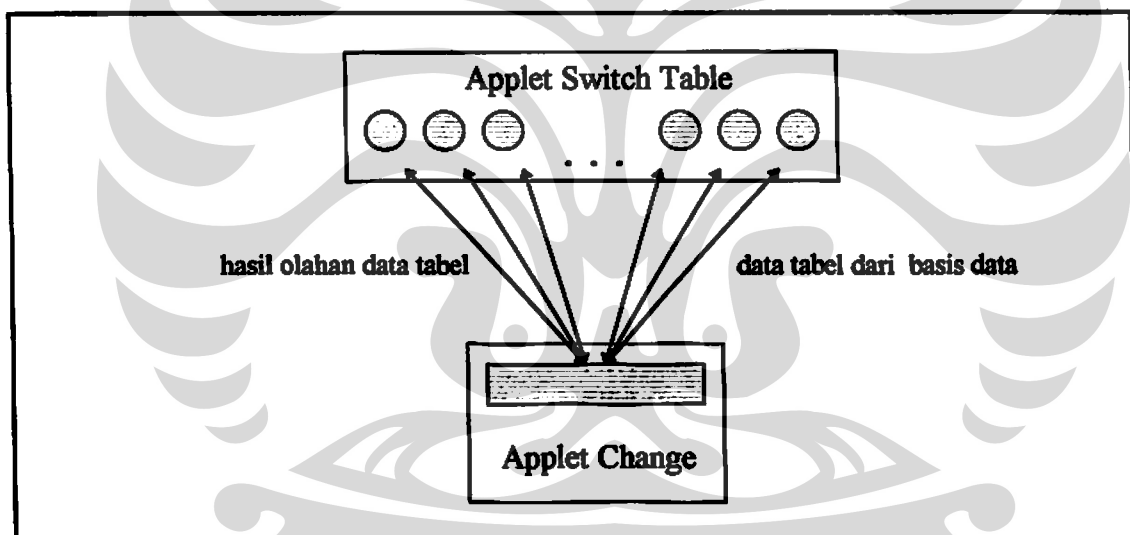
Applet yang menjalankan kendali ini adalah *Applet* Peta. *Applet* Peta akan membaca *file* *struktur.mttw* di pelayan pada pendefinisian kata *subindeks*. Kendali ini akan mengumpulkan *page* acuan utama tersebut dan akan menampilkannya ke pemakai bila kendali tersebut diaktifkan. Berikut adalah gambaran tentang perpindahan *page* acuan utama yang satu ke *page* acuan utama lainnya dalam Table Map :



Gambar 3.11 Perpindahan *page* acuan utama dalam Table Map

3.3.4 Komunikasi antar *applet* aplikasi

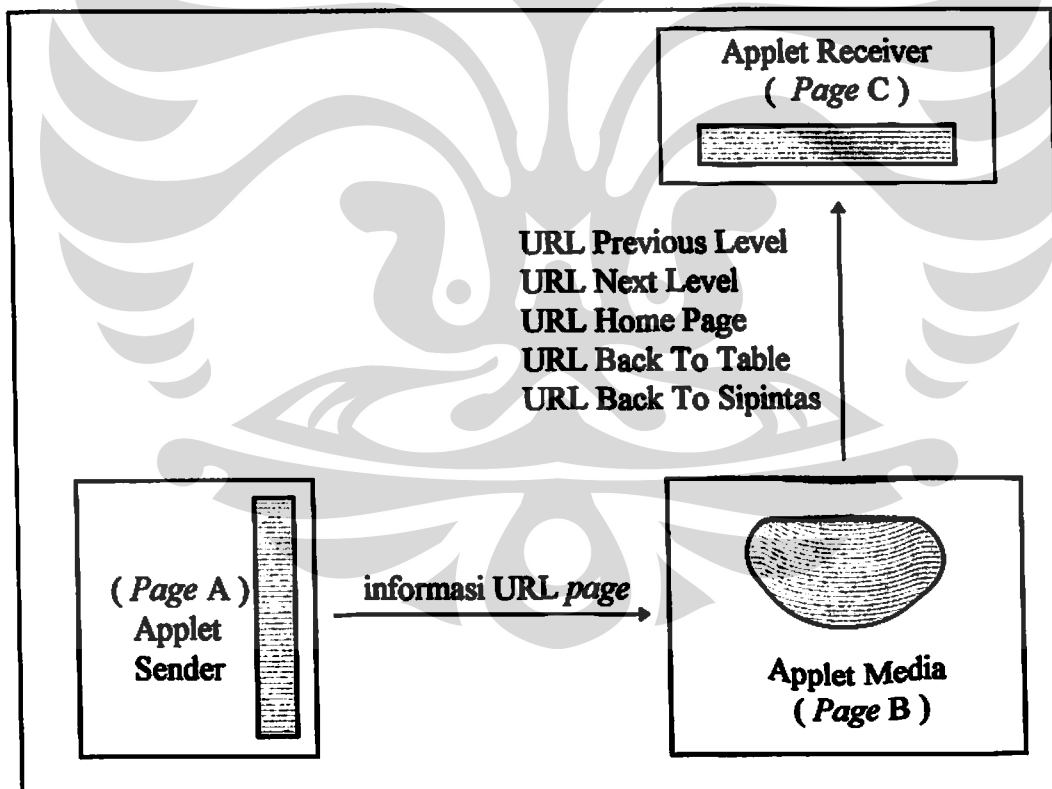
Komunikasi antar *applet* aplikasi menggunakan teknik komunikasi antar *applet* pada *page* yang sama. *Applet* tersebut adalah Applet Switch Table dan Applet Change sebagai *applet* aplikasi. Applet Switch Table akan berada sebanyak data tabel yang diambil dari pelayan basis data. Applet Switch Table yang banyak ini akan berkomunikasi dengan satu Applet Change. Berikut akan digambarkan secara keseluruhan komunikasi antar *applet* yang terjadi pada *page* yang sama beserta informasi yang dipertukarkannya :



Gambar 3.12 Komunikasi antar *applet* aplikasi

3.3.5 Komunikasi antar *applet* kendali

Komunikasi antar *applet* kendali menggunakan teknik komunikasi antar *applet* pada *page* yang berbeda. *Applet* tersebut adalah Applet Sender, Applet Media dan Applet Receiver. Applet Sender mengirim informasi URL *page* yang sedang aktif. Applet Media bertugas sebagai media perantara antara Applet Sender dan Applet Receiver. Applet Receiver yang akan mengambil daftar URL dari Applet Media dan menampilkan menu kendali kepada pemakai. Berikut akan digambarkan secara keseluruhan komunikasi antar *applet* yang terjadi pada *page* yang berbeda beserta informasi yang dipertukarkannya :



Gambar 3.13 Komunikasi antar *applet* kendali

BAB IV

PENERAPAN DAN UJI COBA PROTOTIPE

4.1 PENERAPAN PROTOTIPE APLIKASI MAHASISWA TIDAK TEPAT WAKTU (MTTW) UNTUK KETUA JURUSAN

Penerapan dan uji coba ditujukan kepada pemakai aplikasi SIPINTAS yang digunakan oleh Ketua Jurusan yaitu aplikasi Mahasiswa Tidak Tepat Waktu (MTTW). Tujuan dibuatnya aplikasi MTTW adalah agar Ketua Jurusan dapat mengetahui mahasiswa-mahasiswa yang terlambat atau tidak tepat waktu dalam menyelesaikan masa studinya. Mahasiswa-mahasiswa tersebut dapat dilihat oleh Ketua Jurusan apakah keterlambatan mereka disebabkan oleh mata kuliah yang masih mereka ambil, Kerja Praktek/Proyek Kelompok yang belum terselesaikan atau Tugas Akhir yang terlalu lama.

Teknik yang diterapkan dalam Aplikasi MTTW ini adalah teknik *drill down investigation* (penelusuran lebih lanjut). Melalui teknik ini Ketua Jurusan dapat menelusuri lebih lanjut informasi yang diberikannya secara global ke informasi yang lebih terinci.

Basis data yang digunakan adalah basis data Sybase. Semester yang akan ditampilkan yaitu semester 8 sampai dengan semester 15. Ketua Jurusan dapat melihat mahasiswa-mahasiswa yang tidak tepat waktu antara semester ini. Alasan digunakannya semester 8 untuk semester yang paling awal, karena mahasiswa yang

paling cepat lulus jatuh pada semester ini, dan semester 15 adalah batas akhir semester yang ditetapkan dalam masa perkuliahan.

Data yang dibangun dari basis data adalah data mahasiswa yang belum lulus. Data ini harus diolah ke data mahasiswa yang sudah lulus agar diperoleh bobot masing-masing angkatan. Perhitungan untuk mahasiswa yang belum dan sudah lulus adalah sebagai berikut :

Semester	8	9	10	11	12	13	14	15
mahasiswa yang belum lulus	26	25	21	16	10	6	2	0

semester	8	9	10	11	12	13	14	15
mahasiswa yang sudah lulus	0	1	4	5	6	4	4	2

$26-25$ $25-21$ $21-16$ $16-10$ $10-6$ $6-2$ $2-0$

Gambar 4.1 Perhitungan mahasiswa yang belum dan sudah lulus

Misalkan data angkatan 1986 mempunyai total siswa sebanyak 26 orang. Pada saat semester 8 berlangsung terdapat satu orang mahasiswa yang sudah lulus. Kelulusan tersebut baru terhitung pada saat semester 9 berlangsung. Jadi perhitungan untuk mahasiswa yang sudah lulus pada semester yang bersangkutan adalah mahasiswa yang belum lulus pada semester sebelumnya dikurangi mahasiswa yang belum lulus pada semester yang bersangkutan.

Data mahasiswa yang sudah lulus untuk tiap-tiap angkatan akan dipergunakan untuk perhitungan bobot tiap-tiap angkatan dan bobot untuk tiap-tiap tahun ajaran dan semester tertentu. Berikut akan diterangkan perhitungan tiap-tiap bobot tersebut.

4.1.1 Perhitungan bobot angkatan

Perhitungan bobot masing-masing angkatan menunjukkan rata-rata semester mahasiswa yang lulus pada angkatan tersebut. Perhitungan bobot itu adalah :

$$\frac{\sum (\text{mahasiswa yang lulus pada semester itu } X \text{ semester yang bersangkutan})}{\sum (\text{mahasiswa yang lulus pada angkatan tersebut})}$$

Misalkan angkatan 1986 mempunyai data mahasiswa yang sudah lulus :

semester	8	9	10	11	12	13	14	15
mahasiswa yang sudah lulus	0	1	4	5	6	4	4	2

Tabel 4.1 Data mahasiswa yang sudah lulus

maka perhitungan bobot untuk angkatan 1986 adalah :

$$\frac{(0 \times 8) + (1 \times 9) + (4 \times 10) + (5 \times 11) + (6 \times 12) + (4 \times 13) + (4 \times 14) + (2 \times 15)}{(0 + 1 + 4 + 5 + 6 + 4 + 4 + 2)}$$

$$= 12.1$$

Jadi mahasiswa angkatan 1986 rata-rata lulus pada semester ke 12.

4.1.2 Perhitungan bobot tahun ajaran

Perhitungan bobot tahun ajaran menunjukkan rata-rata semester mahasiswa yang lulus untuk angkatan-angkatan yang ada pada tahun ajaran dan semester yang bersangkutan. Rumus perhitungan bobot tersebut adalah :

$$\frac{\sum (\text{bobot tiap-tiap angkatan})}{\sum \text{angkatan}}$$

Misalkan bobot masing-masing angkatan 1986, 1987, 1988, 1989, dan 1990 untuk tahun ajaran 1993/1994 semester gasal adalah 12.00, 11.90, 11.50, 12.10, dan 10.67. Bobot tersebut diperlihatkan dalam tabel sebagai berikut :

Angkatan	Bobot
1986	12.00
1987	11.90
1988	11.50
1989	12.10
1990	10.67

Tabel 4.2 Bobot angkatan

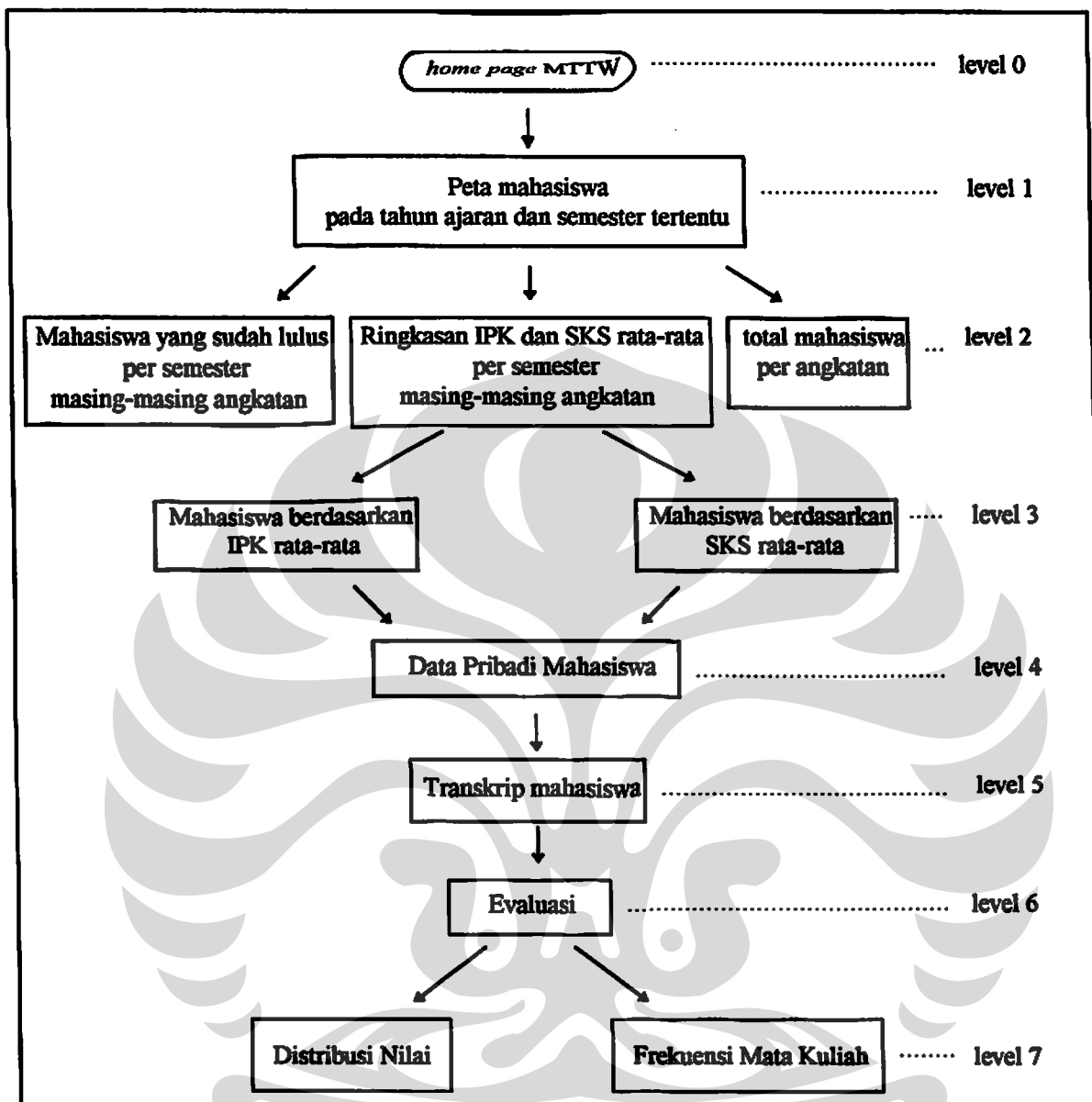
Maka perhitungan untuk bobot tahun ajaran 1993/1994 semester gasal adalah :

$$\frac{12.00 + 11.90 + 11.50 + 12.10 + 10.67}{5} = 11.63$$

Jadi rata-rata mahasiswa-mahasiswa yang lulus untuk angkatan 1986 sampai dengan 1990 tersebut adalah pada semester 12 (mendekati 12) untuk tahun ajaran 1993/1994 semester gasal.

4.1.3 Peta *page* MTTW

Peta *page* MTTW menggambarkan secara keseluruhan *page* yang ada pada aplikasi MTTW. Berikut akan digambarkan peta *page* MTTW beserta *level*-nya masing-masing :



Gambar 4.2 Peta *page* MTTW

Keterangan gambar :

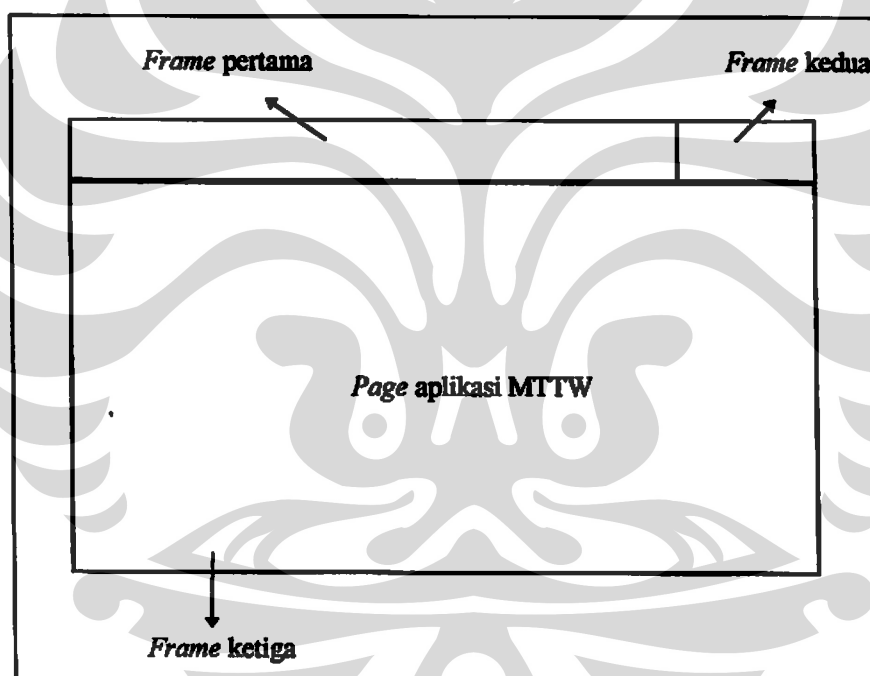
1. *Level 0* : *Page* yang terdapat pada *level* ini adalah *home page* MTTW. Pada *home page* ini terdapat informasi bobot tahun ajaran dan semester tertentu.

2. *Level 1* : *Page* pada *level* ini adalah *page* peta mahasiswa pada tahun ajaran dan semester tertentu. Pada *page* ini terdapat informasi tiap-tiap angkatan seperti total siswa per angkatan, data mahasiswa yang belum dan sudah lulus beserta bobot tiap-tiap angkatan.
3. *Level 2* : *Page* mahasiswa yang sudah lulus dan total siswa tiap angkatan berada pada *level* ini. *Page* ringkasan IPK dan SKS rata-rata mahasiswa yang belum lulus tiap angkatan juga berada pada *level* ini.
4. *Level 3* : *Page* pada *level* ini terbagi atas *page* untuk menampilkan mahasiswa berdasarkan IPK rata-rata dan *page* untuk menampilkan mahasiswa berdasarkan SKS rata-rata.
5. *Level 4* : *Page* pada *level* ini memperlihatkan data pribadi yang dimiliki oleh mahasiswa berupa informasi KP atau TA yang sedang dilakukan beserta informasi jumlah SKS penuh yang diambil oleh mahasiswa.
6. *Level 5* : *Page* pada *level* ini adalah *page* yang berisi transkrip nilai mahasiswa yang dapat dilihat secara keseluruhan.
7. *Level 6* : Pada *level* ini terdapat informasi untuk memilih *page* distribusi nilai dan frekuensi mata kuliah.
8. *Level 7* : Pada *level* ini terdapat *page* distribusi nilai dan frekuensi mata kuliah. *Page* distribusi nilai menyatakan nilai rata-rata yang dimiliki setiap mata kuliah. *Page* Frekuensi mata kuliah menyatakan berapa kali banyaknya mahasiswa yang mengulang untuk setiap mata kuliah.

4.2 IMPLEMENTASI APPLLET JAVA DALAM APLIKASI MTTW

4.2.1 Implementasi Applet Java Kendali

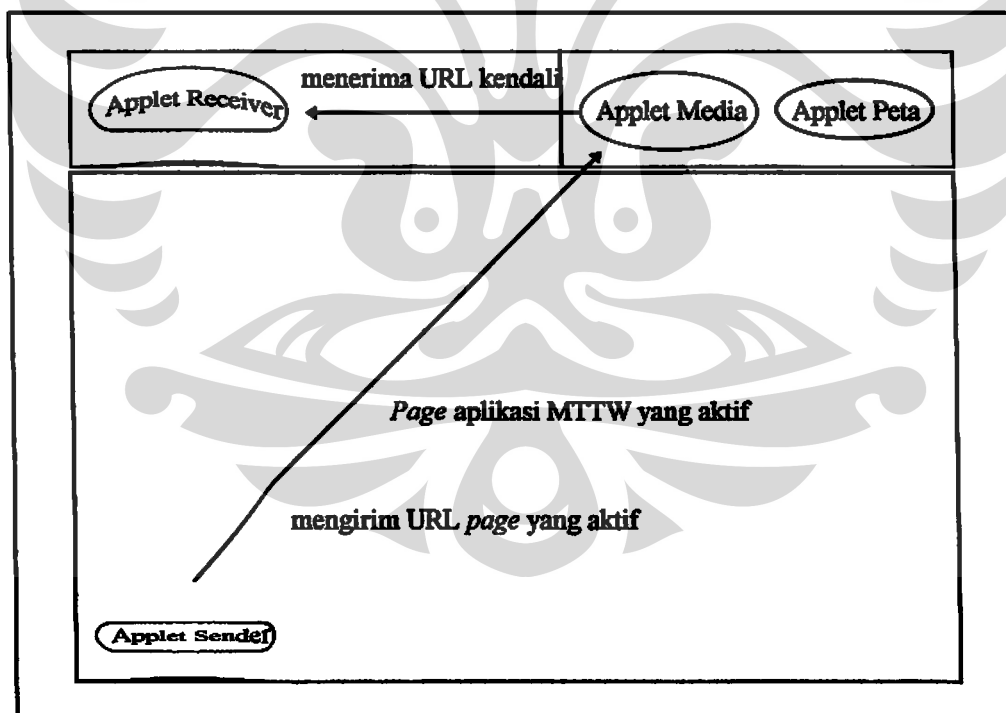
Aplikasi MTTW terbagi atas tiga *frame* yang berbeda. *Frame* pertama adalah *frame* untuk Applet Receiver. *Frame* kedua adalah *frame* untuk Applet Media beserta Applet Peta dan *frame* yang ketiga adalah *frame* untuk menampilkan *page* aplikasi MTTW. Berikut akan digambarkan ketiga *frame* tersebut :



Gambar 4.3 *Frame* dalam aplikasi MTTW

Penggunaan *frame* ini dimaksudkan agar *applet* kendali berada terus pada *frame* yang pertama dan kedua sementara *page* aplikasi MTTW terus berpindah-pindah dari satu *page* ke *page* yang lainnya.

Frame yang pertama terdiri dari Applet Receiver di mana *applet* ini menampilkan menu kendali kepada pemakai. Menu kendali yang ditampilkan adalah kendali Previous dan Next Level, Home Page, Back to Table, dan Back To Sipintas. *Frame* yang kedua terdiri dari Applet Media dan Applet Peta. Applet Peta bertugas untuk menampilkan kendali Table Map. Applet Media bertugas sebagai media perantara antara Applet Sender dan Applet Receiver. Applet Sender berada pada tiap-tiap *page* aplikasi MTTW di mana *applet* ini bertugas untuk mengirimkan URL *page* aplikasi yang sedang aktif saat itu. Berikut akan digambarkan *applet* kendali beserta informasi yang dikirim oleh Applet Sender ke Applet Receiver melalui Applet Media :

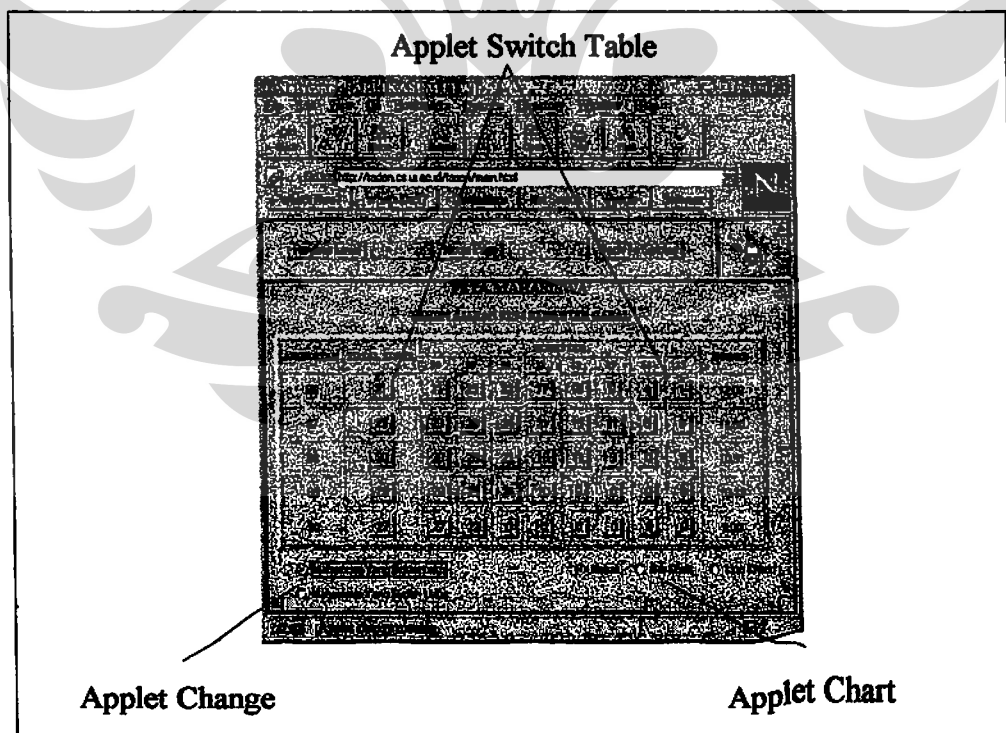


Gambar 4.4 Applet Java Kendali

Jadi Applet Sender mengirim URL ke Applet Receiver melalui *frame* yang berbeda. Applet Media yang bertugas sebagai media perantara dan menjaga URL *list page* yang sudah dikunjungi juga berada pada *frame* yang berbeda. Masing-masing *frame* tersebut adalah *page* HTML yang berbeda. Jadi teknik komunikasi yang digunakan oleh *applet* kendali adalah komunikasi antar *applet* pada *page* yang berbeda.

4.2.2 Implementasi Applet Java Aplikasi dan Presentasi

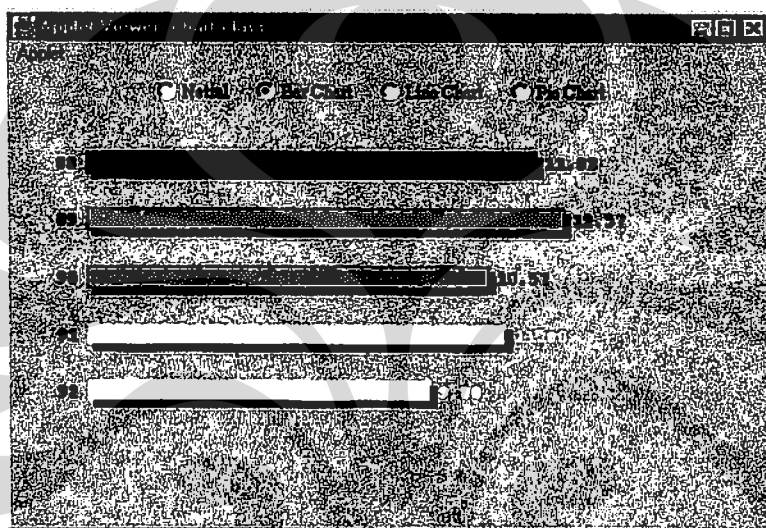
Applet Java aplikasi MTTW dan Applet Java presentasi MCSIR berada pada *page* aplikasi MTTW. *Page* tersebut adalah *page* peta mahasiswa pada tahun ajaran dan semester tertentu. Berikut adalah *page* peta mahasiswa dengan *applet* aplikasi dan presentasi yang berada pada *page* tersebut :



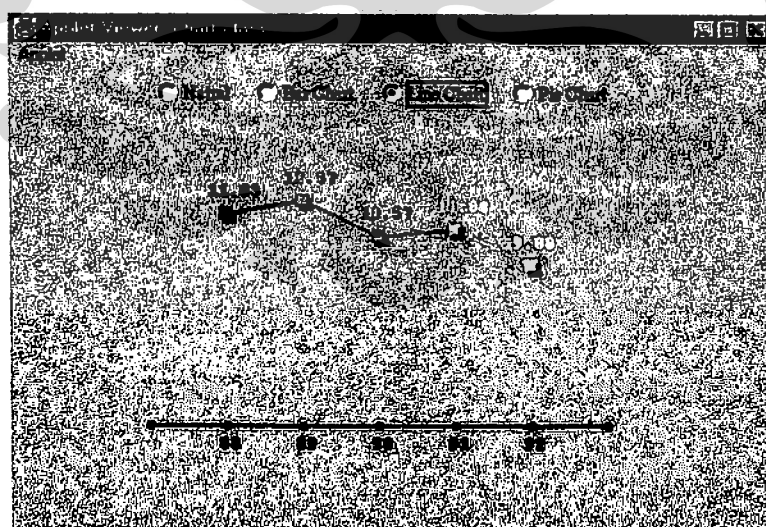
Gambar 4.5 Applet Java Aplikasi dan Presentasi

Applet Switch Table dan Applet Change adalah Applet Java yang digunakan untuk aplikasi. Applet Chart adalah Applet Java yang digunakan untuk presentasi. *Applet* aplikasi saling berkomunikasi dengan menggunakan teknik komunikasi antar *applet* pada *page* yang sama.

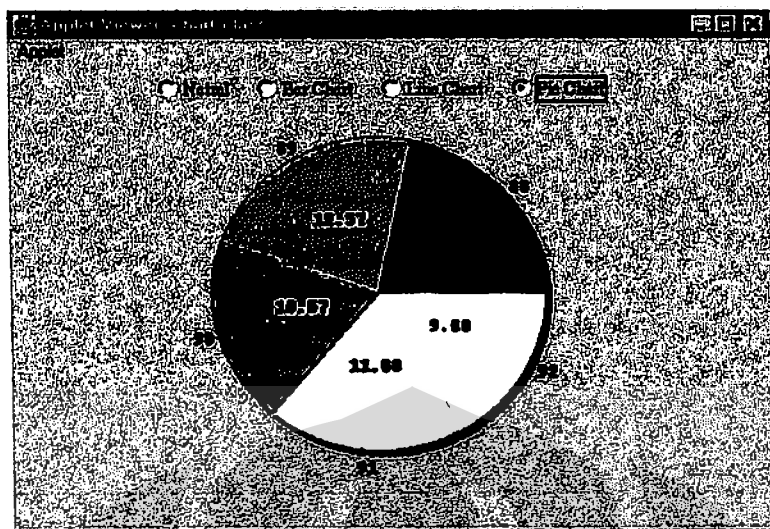
Grafik yang akan ditampilkan oleh *applet* presentasi adalah *bar chart*, *line chart* dan *pie chart*. Berikut adalah rancangan dari masing-masing grafik :



Gambar 4.6a Bar Chart



Gambar 4.6b Line Chart



Gambar 4.6c Pie Chart

Data yang diperoleh dari basis data Sybase didapat dengan menggunakan modul yang berada pada bagian *gateway* yang bekerjasama dengan modul presentasi [SOF96]. Modul *gateway* tersebut bernama InfoAkses dan modul presentasinya bernama Formasi.

Format pemakaian modul InfoAkses dan Formasi adalah sebagai berikut :

```
InfoAkses[Formasi][nama file html][SQL query][login][password]
        [versi sybase][direktori sybase]
```

Jadi data-data yang didapat oleh aplikasi MTTW menggunakan modul InfoAkses. Applet Switch Table sebagai *applet* aplikasi juga menangkap data dari modul InfoAkses melalui modul Formasi yang memberi masukan ke *browser* berupa gabungan dari *file html* dengan *SQL query* yang telah didapat dari basis data Sybase.

4.3 ALGORITMA MODUL-MODUL APPLLET JAVA

4.3.1 Algoritma Modul Applet Java Aplikasi

Modul Applet Java aplikasi terbagi atas dua *applet*. *Applet* pertama yaitu Applet Switch Table yang bertugas menangkap data tabel dari pelayan basis data dan menampilkan kembali data tabel yang telah diolah dari Applet Change. *Applet* kedua yaitu Applet Change yang mengolah data tabel di klien.

Algoritma Applet Java Switch Table :

1. Applet Switch Table menangkap data tabel mahasiswa yang belum lulus dari basis data Sybase melalui argumen Applet Java yang ada.
2. Applet Switch Table juga menerima argumen alamat URL mahasiswa yang belum dan sudah lulus.
3. Applet Switch Table menunggu *event* dari pemakai. Bila pemakai melakukan *event* berupa penekanan tombol Applet Switch Table, maka Applet Switch Table membuka komunikasi dengan Applet Change untuk mengetahui *flag* yang ada pada Applet Change. Jika *flag* = 0 berarti Applet Switch Table harus mengacu ke mahasiswa yang belum lulus. Jika *flag* = 1 berarti harus mengacu ke mahasiswa yang sudah lulus.

Algoritma Applet Java Change :

1. Applet Change pertama kali membuka komunikasi dengan Applet Switch Table. Komunikasi ini dilakukan agar diperoleh data tabel mahasiswa yang belum lulus. Setelah didapat data tabel tersebut, maka dilakukan

perhitungan tabel mahasiswa yang belum lulus ke tabel mahasiswa yang sudah lulus. Tabel ini disimpan oleh Applet Change ke dalam dua *array* yang berbeda.

2. Applet Change menunggu *event* dari pemakai. Jika pemakai menggunakan *event* dengan menekan tombol mahasiswa yang sudah lulus, maka Applet Change membuka komunikasi lagi ke Applet Switch Table dengan mengirimkan data tabel mahasiswa yang sudah lulus dan menset *flag* menjadi 1. Jika *event* dari pemakai adalah dengan menekan tombol mahasiswa yang belum lulus lagi, maka Applet Change membuka komunikasi lagi dengan Applet Switch Table dan mengirimkan data tabel mahasiswa yang belum lulus serta menset *flag* menjadi 0. Demikian seterusnya komunikasi ini berlangsung.

4.3.2 Algoritma Modul Applet Java Presentasi

Modul Applet Java presentasi yaitu modul Applet Chart yang menangkap data X dan Y *axis* dari pelayan basis data dan menampilkan grafik batang, garis dan lingkaran.

Algoritma Applet Java Chart :

1. Applet Chart pertama kali dijalankan dengan menangkap data X dan Y *axis* yang diterimanya dari parameter yang diberikan. Data X dan Y *axis* ini berasal dari pelayan basis data. Applet Chart juga menerima parameter URL

agar grafik yang ditampilkan oleh Applet Chart dapat diacu ke alamat yang diinginkan.

2. Selanjutnya Applet Chart melakukan konversi data *Y axis* dari *string* ke bentuk bilangan real. Data ini dipergunakan untuk perhitungan panjang grafik batang dan garis serta area grafik lingkaran yang akan dibuat.
3. Applet Chart menunggu *event* dari pemakai. Jika pemakai menggunakan *event* untuk menggambar grafik batang, maka Applet Chart akan men-*set array* untuk mendefinisikan daerah grafik batang yang akan digunakan oleh *mouse* dan menggambar grafik batang tersebut melalui metode *paint()* yang ada. Bila pemakai menggunakan *event* untuk grafik garis, maka Applet Chart akan menggambar grafik garis dan men-*set* daerah grafik garis yang akan digunakan oleh *mouse*. Demikian juga bila pemakai menggunakan *event* untuk menampilkan grafik kue, maka Applet Chart akan menggambar grafik kue dan men-*set* daerah grafik kue yang akan digunakan oleh *mouse*. Penggambaran grafik kue ini berbeda dengan penggambaran grafik batang dan garis. Pada grafik kue dipakai rumus juring matematika untuk menggambarkan area lingkaran tertentu beserta mendefinisikan area pergerakan *mouse* yang menggunakan rumus *seper-radian* agar sudut yang dihasilkan didapat secara tepat.
4. Bila pemakai menekan tombol *mouse* pada daerah grafik tertentu, maka Applet Chart akan melakukan *link* ke alamat URL yang telah didefinisikan pada daerah grafik tersebut.

4.3.3 Algoritma Modul Applet Java Kendali

Modul Applet Java kendali terbagi atas tiga. Modul pertama adalah Applet Sender yang bertugas untuk mengirim alamat URL *page* yang aktif. Modul kedua adalah Applet Media yang bertugas sebagai media perantara dan juga bertugas untuk menjaga URL *list page* yang sudah dikunjungi. Modul terakhir adalah Applet Receiver yang bertugas untuk menampilkan kendali Previous dan Next Level, Home Page, Back To Table dan Back To Sipintas. *Applet* lainnya yang bertugas sebagai kendali tetapi tidak termasuk dalam ketiga modul URL di atas adalah Applet Peta.

Algoritma Applet Java Sender :

1. Pertama-tama Applet Sender akan mendapatkan alamat URL *page* yang sedang aktif.
2. Selanjutnya Applet Sender akan mengirim alamat URL tersebut ke Applet Media melalui metode *sendURL(URL)* yang ada pada Applet Media.
3. Applet Sender akan *men-trigger frame* Applet Receiver agar *di-refresh* ulang, supaya Applet Receiver dapat mengambil kembali daftar alamat URL dari Applet Media.

Algoritma Applet Java Media :

1. Applet Media pertama kali membaca *file* struktur.mttw yang ada pada pelayan. Dari *file* ini akan akan terbaca *page* mana saja sebagai *home page*, *page* acuan utama dan *level page* aplikasi.

2. Applet Media menunggu *event* dari Applet Sender. Jika Applet Sender mengirim alamat URLnya ke Applet Media, maka Applet Media akan mengecek alamat URL ini. Jika alamat URL ini ada dalam hierarki *page* aplikasi, maka URL ini akan dimasukkan ke dalam *list page* yang sudah dikunjungi. Selanjutnya Applet Media akan membuat alamat URL yang akan dikirim ke Applet Receiver.
3. URL ini terdapat dalam metode `getURL()` yang akan diakses Applet Receiver.

Algoritma Applet Java Receiver :

1. Applet Receiver akan mengambil alamat URL untuk kendali Previous dan Next Level, Home Page, Back To Table dan Back To Sipintas.
2. Selanjutnya Applet Receiver akan menampilkan kendali ini ke pemakai. Kendali tersebut aktif bila alamat URLnya ada dan pasif jika tidak ada.
3. Applet Receiver menunggu *event* dari pemakai. Jika pemakai menggunakan salah satu kendali yang aktif, maka Applet Receiver akan menampilkan aplikasi MTTW pada *frame* ketiga sesuai dengan alamat URL yang ada.

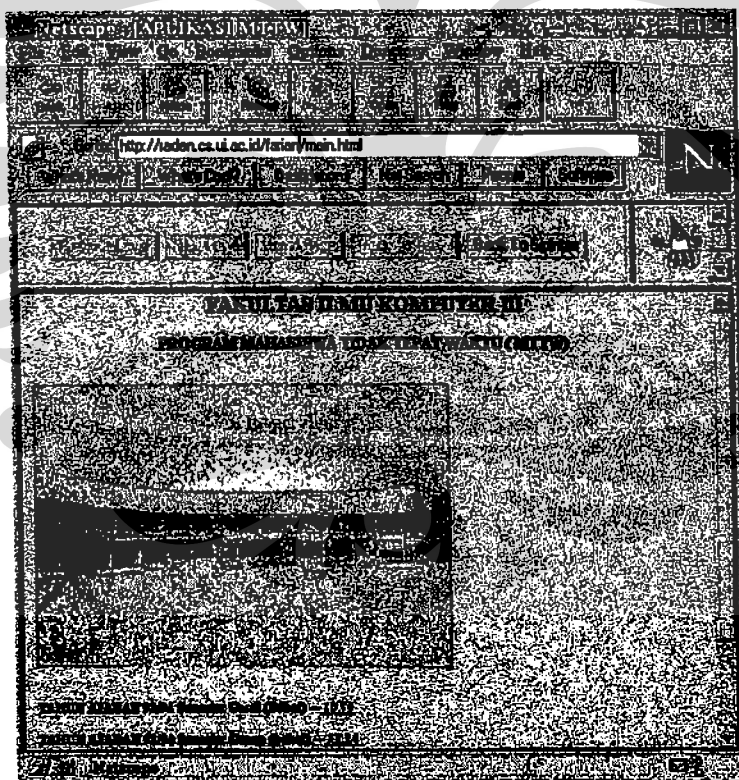
Algoritma Applet Java Peta :

1. Applet Peta pertama kali membaca *file* struktur.mttw. Applet Peta membaca *file* ini pada bagian pendefinisian *page* acuan utama. Beberapa *page* acuan utama yang didefinisikan akan dimasukkan ke dalam *list*.
2. Applet Peta menampilkan kendalinya ke pemakai dalam bentuk *image* yang dapat diklik oleh pemakai.

3. Bila image tersebut diklik oleh pemakai, maka sebuah *window* baru akan muncul. *Window* tersebut akan menampilkan *list page* acuan utama. Jika pemakai melakukan *double click* pada *list page* acuan utama, maka *frame* ketiga yang ada pada *browser* akan di-*refresh* dan akan digantikan oleh *page* acuan utama yang diklik tersebut.

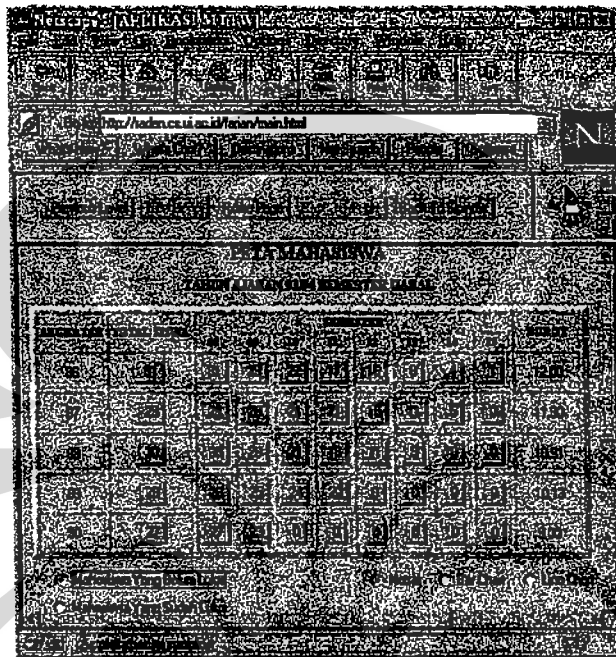
4.4 UJI COBA

Berikut adalah uji coba Applet Java yang diterapkan pada aplikasi MTTW. Saat pertama kali aplikasi MTTW dijalankan, maka *page* yang pertama kali muncul adalah *home page* aplikasi.



Gambar 4.7 Home page MTTW

Bila pemakai melakukan penelusuran lebih lanjut terhadap bobot tahun ajaran, maka *page* yang akan ditampilkan adalah *page* peta mahasiswa. Pada *page* ini terdapat informasi tiap-tiap angkatan. *Default* yang akan ditampilkan pada *page* ini adalah *page* peta mahasiswa yang belum lulus.



Gambar 4.8 *Page* peta mahasiswa

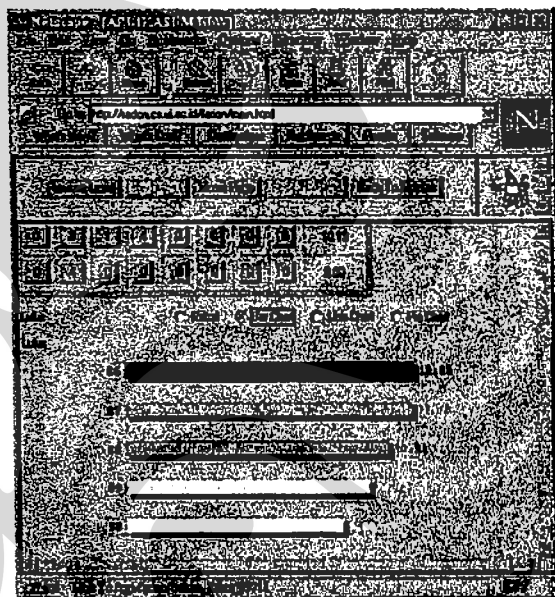
Pada *page* ini terdapat *applet* untuk aplikasi dan presentasi. *Applet* aplikasi yaitu Applet Switch Table dan Applet Change. *Applet* presentasi yaitu Applet Chart. Applet Switch Table akan di-*load* sebanyak data tabel yang akan ditangkap. Akibat banyaknya Applet Switch Table yang akan muncul pada *page* ini, maka *load page* ini cukup lama.

Bila pemakai ingin melihat mahasiswa yang sudah lulus, maka pemakai dapat memilih *Check Box* mahasiswa yang sudah lulus. Perhitungan tabel mahasiswa yang

belum lulus ke tabel mahasiswa yang sudah lulus dilakukan di klien. Pemakai dapat juga memilih *Check Box* untuk menampilkan grafik yang akan ditampilkan oleh Applet Chart. Grafik ini merepresentasikan tiap angkatan beserta bobotnya masing-masing. Berikut adalah gambaran *page* yang akan ditampilkan bila pemakai memilih *Check Box* untuk tabel mahasiswa yang sudah lulus dan grafik *Bar Chart* pada *page* yang sama :

No	Nama	Jenis Kelamin	Mata Kuliah	SKS	Nilai
01					
02					
03					
04					
05					

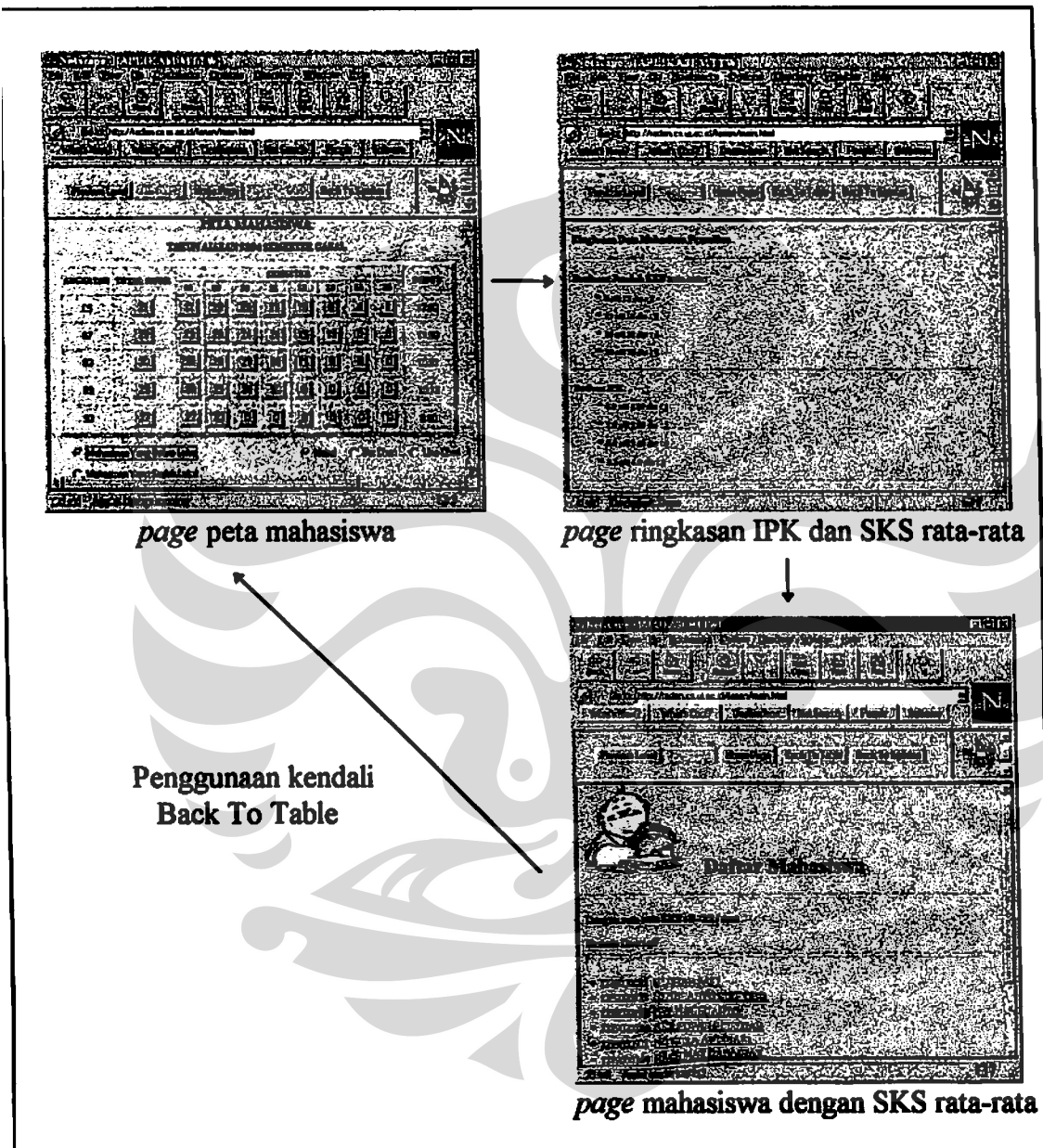
Gambar 4.9a mahasiswa yang sudah lulus



Gambar 4.9b Grafik *Bar Chart*

Penggunaan kendali-kendali yang ada pada *frame* kendali adalah sebagai berikut : misalkan pemakai melakukan penelusuran lebih lanjut terhadap *page* peta mahasiswa ke *page* IPK dan SKS rata-rata, lalu ditelusuri lagi ke *page* nama-nama mahasiswa berdasarkan SKS rata-rata, maka pemakai dapat kembali ke *page* peta mahasiswa dengan menekan tombol kendali *Back To Table*. Berikut adalah gambaran bagaimana pemakai melakukan penelusuran lebih lanjut dari *page* peta mahasiswa

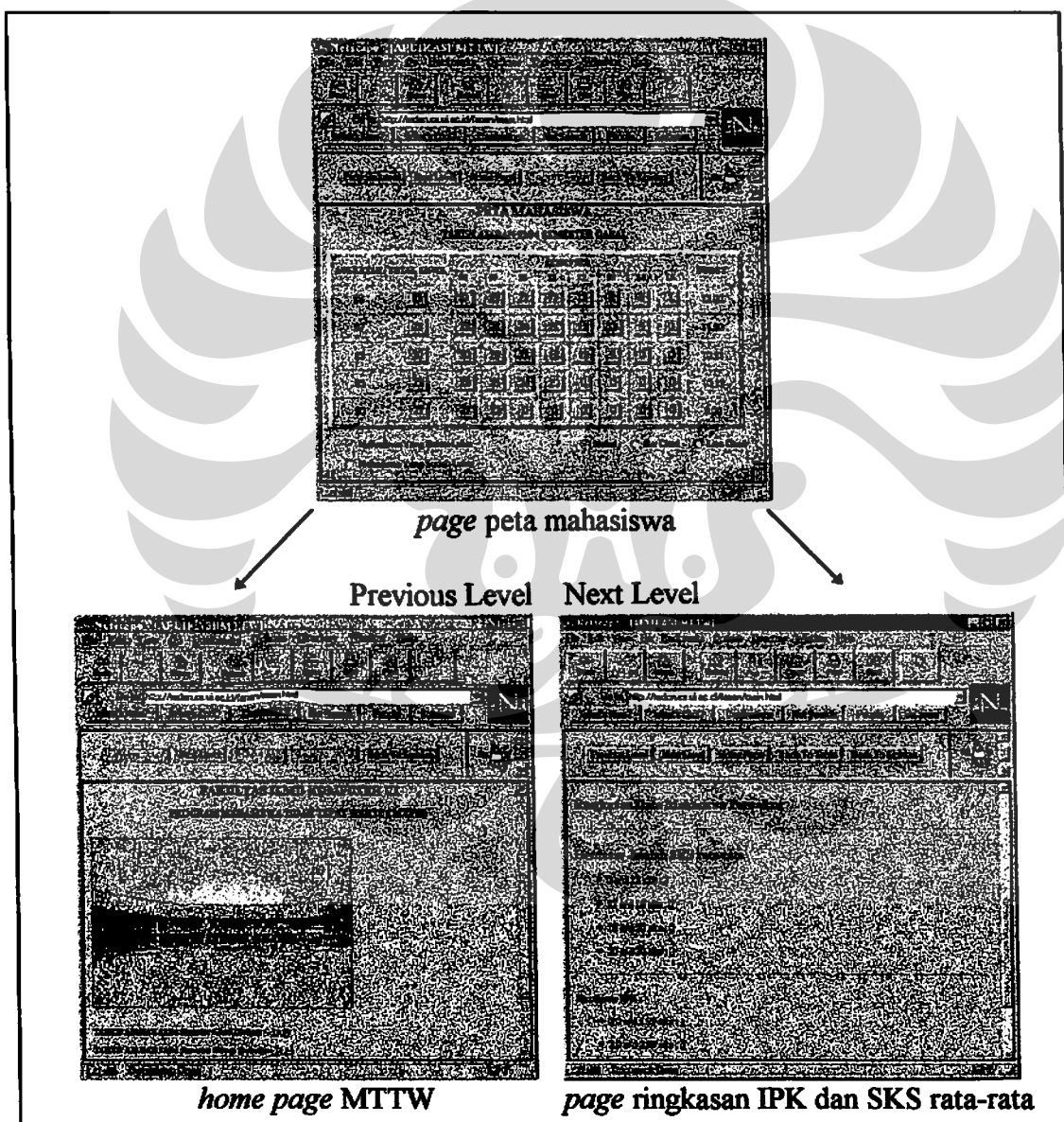
sampai ke *page* nama-nama mahasiswa berdasarkan SKS rata-rata lalu kembali lagi ke *page* peta mahasiswa tersebut dengan menekan tombol kendali Back To Table :



Gambar 4.10 Kendali Back To Table

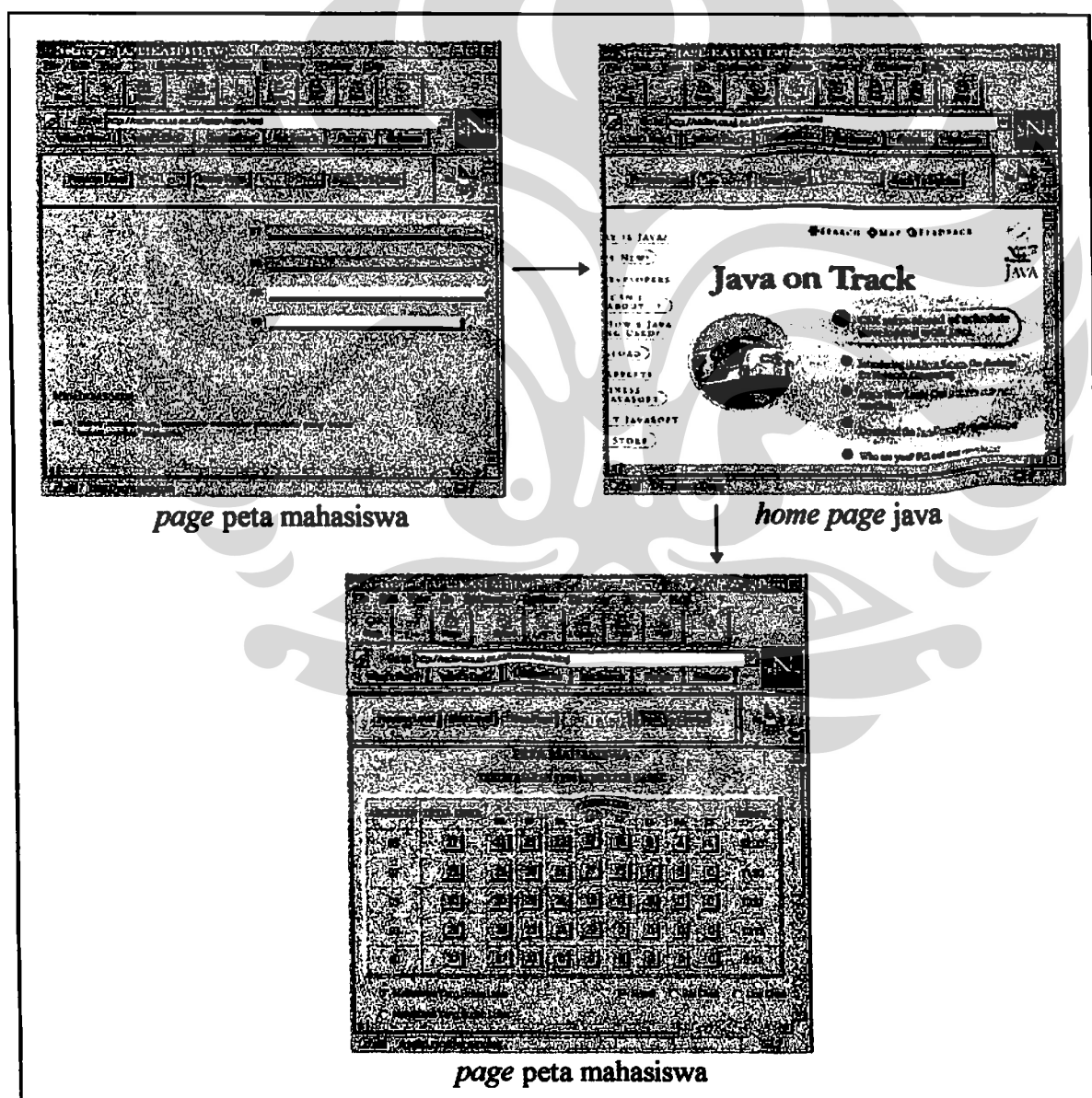
Page peta mahasiswa ini berada dalam pendefinisian kendali Back To Table, sehingga bila kendali ini diaktifkan, maka *page* akan pindah ke *page* peta mahasiswa.

Penggunaan kendali Previous Level dari *page* peta mahasiswa ini akan menyebabkan *page* pindah ke *home page* MTTW dan penggunaan kendali Next Level akan menyebabkan *page* pindah ke *page* ringkasan IPK dan SKS rata-rata. Berikut adalah penggunaan kendali Previous dan Next Level dari *page* peta mahasiswa :



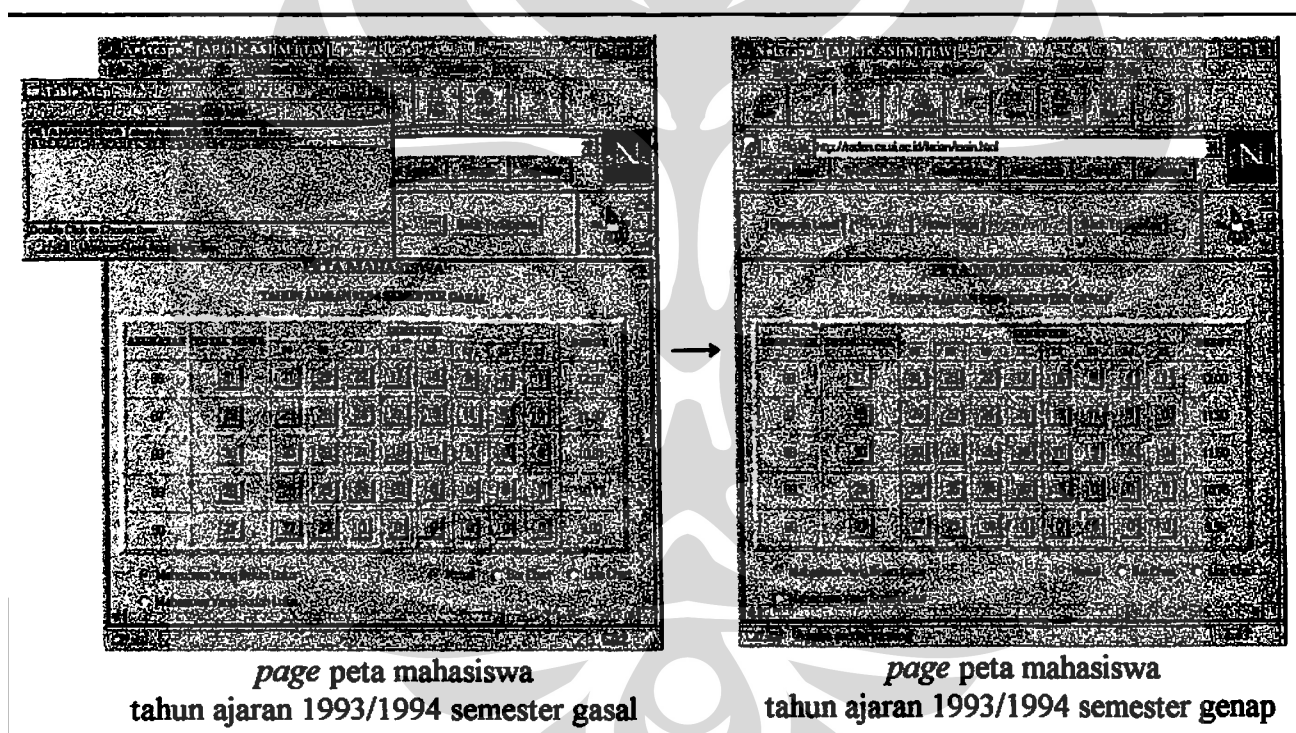
Gambar 4.11 Kendali Previous dan Next Level

Kendali Back To Sipintas akan berguna bila *page* pindah ke *page* yang bukan dalam aplikasi MTTW. Misalkan pemakai pindah ke *page* *java.sun.com* melalui link *http://java.sun.com* yang ada dalam *page* peta mahasiswa. Jika pemakai ingin kembali ke *page* aplikasi MTTW yang terakhir dikunjunginya, maka kendali Back To Sipintas dapat digunakan. Berikut adalah gambaran jika pemakai pindah dari *page* peta mahasiswa ke *page* *java* lalu kembali ke *page* peta mahasiswa melalui kendali Back To Sipintas :



Gambar 4.12 Kendali Back To Sipintas

Kendali Table Map dapat digunakan untuk pindah ke *page* utama lain tanpa melalui kendali-kendali yang ada dalam *frame* kendali. Table Map berada pada *frame* Table Map tersendiri. Misalkan pemakai berada pada *page* peta mahasiswa tahun ajaran 1993/1994 semester gasal dan ingin pindah ke *page* peta mahasiswa tahun ajaran 1993/1994 semester genap, maka kendali Table Map dapat digunakan dengan menekan *image* yang ada dan melakukan *double click* terhadap *page* yang ingin dikunjungi. Berikut adalah gambaran perpindahan *page* peta mahasiswa tahun ajaran 1993/1994 semester gasal ke semester genap :



Gambar 4.13 Kendali Table map

Kendali-kendali yang ada pada *frame* kendali dan *frame* Table Map masih dapat berfungsi selama *frame* tersebut masih tetap ada dalam *browser* walaupun *page* berpindah-pindah pada *frame* aplikasi MTTW.

BAB V

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Pemanfaatan Applet Java yang dirancang dan diimplementasikan dalam tugas akhir yaitu pemanfaatan Applet Java untuk prototipe MCSIR dan pemanfaatan Applet Java untuk prototipe aplikasi MTTW. Applet Java yang dirancang dan diimplementasikan pada prototipe MCSIR (pada bagian presentasi dan kendali) dapat dipakai oleh aplikasi yang memanfaatkan layanan MCSIR, sedangkan Applet Java yang dirancang dan diimplementasikan pada prototipe aplikasi MTTW dibuat khusus untuk aplikasi ini.

Pemanfaatan Applet Java untuk prototipe aplikasi MTTW mempunyai kemampuan dapat mengolah kembali tabel langsung di klien, tanpa perlu mengambil kembali tabel yang diperlukan ke pelayan basis data. Pengolahan yang dilakukan langsung di klien mempunyai waktu eksekusi yang lebih cepat dibandingkan harus mengambil kembali tabel yang diperlukan dari pelayan basis data.

Pemanfaatan Applet Java pada bagian presentasi MCSIR dapat menampilkan data dalam bentuk teks yang berasal dari pelayan basis data ke dalam bentuk grafik *bar*, *line* dan *pie chart*. Grafik yang ditampilkan oleh Applet Java presentasi juga mempunyai kemampuan *link ke page* lainnya.

Pemanfaatan Applet Java pada bagian kendali MCSIR mempunyai kemampuan untuk mengendalikan *page* aplikasi. Applet Java pada bagian kendali ini akan membantu pemakai agar dapat berpindah-pindah *page* ke *page* lain dengan mudah dalam aplikasi SIPINTAS.

Applet Java yang dirancang dan diimplementasikan pada bagian presentasi dan kendali MCSIR dapat bekerja sama dengan bagian gerbang (*gateway*) dan presentasi MCSIR yang lainnya yang diterapkan dengan menggunakan CGI Script.

5.2 SARAN

Saran-saran yang diharapkan penulis dalam pengembangan lebih lanjut sehingga tugas akhir ini dapat disempurnakan adalah :

1. Applet Java yang digunakan untuk aplikasi MTTW dapat dikembangkan dengan menggunakan konsep *multithreading* sehingga eksekusi dapat dilakukan dengan cepat.
2. Pengembangan *applet* kendali yang dapat berkomunikasi dengan *applet* lainnya pada *page* yang berbeda tanpa memakai *applet* media perantara. *Applet* kendali ini tidak akan memerlukan *frame* kendali lagi yang dapat di-*flush* sewaktu-waktu oleh *browser*.
3. Penambahan fasilitas kendali yang lain seperti fasilitas kendali yang dapat memperlihatkan *list page* yang ada sehingga pemakai dapat mengetahui posisinya saat itu.

DAFTAR PUSTAKA

- [ABD95a] Abdat, Sjarif dkk. "*SIPINTAS: an Executive Information System for the University of Indonesia*" in: *Proceeding the Third International Conference on Decision Support System*, Hongkong 1995.
- [ABD95b] Abdat, Sjarif. "Fasilitas Integrasi Informasi Tak Terstruktur dan Terstruktur serta Penerapannya dalam Sistem Informasi Eksekutif", Proposal RUT IV, 1995.
- [COM91] Comer, D.E. *Internetworking With TCP/IP*, 2nd ed., Englewood Cliffs: Prentice Hall Inc., 1991.
- [COR96] Cornell, Gary and Cay S. Horstmann. *Core Java*, Garcia Avenue: The SunSoft Press, 1996.
- [GOS95] Gosling, James and Henry McGilton. *The Java Language Environment*, Garcia Avenue: A Sun Microsystems Inc., 1995.
- [JAV96] JavaSoft. "*JavaSoft Home Page*", <http://java.sun.com/>, 1996.
- [LEM96] Lemay, Laura and Charles L. Perkins. *Teach Yourself Java in 21 Days*, Indianapolis: Sams.net Publishing, 1996.
- [MCL93] Mc. Leod, Raymond. *Management Information System*, 5th ed, New York: Macmillan Publishing Company, 1993.
- [SOF96] Sofianti, Desalina. "Perancangan dan Implementasi Prototipe MCSIR (*Multisources Client-Server Information Retrieval*)", Skripsi Sarjana: Fakultas Ilmu Komputer Indonesia, 1996.
- [STE90] Stevens, W.R. *Unix Network Programming*, Englewood Cliffs: Prentice Hall inc., 1990.
- [TAN89] Tanenbaum, Andrew S. *Computer Networks*, 2nd ed., Englewood Cliffs: Prentice Hall inc., 1989.
- [TUT96] Tutorial Java. "*The Java Tutorial*", <http://raden.cs.ui.ac.id/sun/tutorial/index.html>, 1996.



LAMPIRAN

LAMPIRAN A

PEMAKAIAN PROGRAM APPLET JAVA

Pemakaian program Applet Java terbagi atas pemakaian program Applet Java yang berkomunikasi pada *page* yang sama dan pada *page* yang berbeda, serta pemakaian program Applet Java yang berdiri sendiri.

1. Pemakaian program Applet Java yang berkomunikasi pada *page* yang sama.

Applet Java yang berkomunikasi pada *page* yang sama yaitu Applet Switch Table, Applet Change dan Applet Chart. Berikut akan diterangkan pemakaian program masing-masing *applet* beserta *parameter* yang akan digunakan :

A. Applet Switch Table.

Parameter yang dipakai :

- ◆ `<PARAM NAME=value VALUE="[nilai dari basis data]">`
- ◆ `<PARAM NAME=link_kuliah VALUE="[url mahasiswa belum lulus]">`
- ◆ `<PARAM NAME=link_lulus VALUE="[url mahasiswa sudah lulus]">`
- ◆ `<PARAM NAME=terminate VALUE="ok">`

Keterangan *parameter* :

- **value** digunakan untuk menangkap data tabel yang berasal dari pelayan basis data.
- **link_kuliah** digunakan untuk *link* ke mahasiswa yang belum lulus.

- `link_lulus` digunakan untuk *link* ke mahasiswa yang sudah lulus.
- `terminate` digunakan untuk menghentikan perhitungan Applet Change.

Contoh :

```
<applet codebase=http://raden.cs.ui.ac.id/farian/classes/switch
        code=SwitchTable.class width=30 height=30>
<param name=value value="<!--FIELD, FORMAT=@sem11@">
<param name=link_kuliah value="/cgi-farian/belum_lulus_89?11">
<param name=link_lulus value="/cgi-farian/sudah_lulus_89?11">
</applet>
```

B. Applet Change.

Parameter yang dipakai :

- ◆ `<PARAM NAME=Computes VALUE='[nilai computes]'`

Keterangan parameter :

- `Computes` digunakan untuk menghitung banyaknya semester yang ada.

Contoh :

```
<applet codebase=http://raden.cs.ui.ac.id/farian/classes/switch
        code=change.class width=200 height=300>
<param name=Computes value="8">
</applet>
```

2. Pemakaian program Applet Java yang berkomunikasi pada *page* yang berbeda.

Applet Java yang berkomunikasi pada *page* yang berbeda yaitu Applet Sender, Applet Media, dan Applet Receiver. Berikut akan diterangkan pemakaian program masing-masing *applet* beserta *parameter* yang akan digunakan :

A. Applet Sender.

Parameter yang dipakai :

- ◆ `<PARAM NAME=html VALUE='[url frame Applet Receiver]'`>
- ◆ `<PARAM NAME=target VALUE='[frame Applet Receiver]'`>

Keterangan *parameter* :

- `html` digunakan untuk *merefresh frame* Applet Receiver.
- `target` adalah nama *frame* Applet Receiver.

Contoh :

```
<applet codebase=http://raden.cs.ui.ac.id/farian/classes/menu
      code=sender.class width=5 height=5>
<param name=html value="menu">
<param name=target value="receiver">
</applet>
```

B. Applet Media.

Parameter yang dipakai :

- ◆ `<PARAM NAME=struktur VALUE='[URL file struktur.mttw]'`>

Keterangan *parameter* :

- struktur digunakan untuk mengambil *file* struktur.mttw di pelayan.

Contoh :

```
<applet codebase=http://raden.cs.ui.ac.id/farian/classes/menu
      code=media.class width=10 height=10>
<param name=struktur
value="http://raden.cs.ui.ac.id/farian/classes/menu/struktur.mttw">
</applet>
```

C. Applet Receiver.

Parameter yang dipakai :

- ◆ <PARAM NAME=target VALUE='[*frame* Applet Sender]''>

Keterangan *parameter* :

- target adalah nama *frame* Applet Sender.

Contoh :

```
<applet codebase=http://raden.cs.ui.ac.id/farian/classes/menu
      code=receiver.class width=10 height=10>
<param name=target value="sender">
</applet>
```

2. Pemakaian program Applet Java yang berdiri sendiri.

Applet java yang berdiri sendiri yaitu Applet Chart dan Applet Peta. Berikut akan diterangkan pemakaian program masing-masing *applet* yang berdiri sendiri beserta *parameter* yang akan digunakan :

A. Applet Chart.

Parameter yang dipakai :

- ◆ `<PARAM NAME=x[1-10] VALUE='[nilai X axis]'`>
- ◆ `<PARAM NAME=y[1-10] VALUE='[nilai Y axis]'`>
- ◆ `<PARAM NAME=url[1-10] VALUE='[alamat URL grafik]'`>

Keterangan *parameter* :

- nilai X axis digunakan untuk koordinat X pada grafik.
- nilai Y axis digunakan untuk koordinat Y pada grafik.
- nilai URL digunakan untuk alamat URL agar grafik dapat diacu.

Contoh :

```
<applet codebase=http://raden.cs.ui.ac.id/farian/classes/switch
      code=chart.class width=500 height=300>
<param name=x1 value="<!--FIELD, FORMAT=@angkatan@">
<param name=y1 value="<!--FIELD, FORMAT=@bobot@">
<param name=url1 value="/cgi-farian/belum_lulus_89?10">
<param name=x2 value="<!--FIELD, FORMAT=@angkatan@">
<param name=y2 value="<!--FIELD, FORMAT=@bobot@">
```

```
<param name=url2 value="/cgi-farian/belum_lulus_89?11">
```

```
</applet>
```

B. Applet Peta.

Parameter yang dipakai :

◆ **<PARAM NAME=struktur VALUE='url file struktur.mttw'>**

◆ **<PARAM NAME=image VALUE='url file image'>**

◆ **<PARAM NAME=target VALUE='frame Applet Sender'>**

Keterangan *parameter* :

- struktur digunakan untuk mengambil *file* struktur.mttw di pelayan.
- image digunakan untuk mengambil *file* image di pelayan.
- target adalah nama *frame* Applet Sender.

Contoh :

```
<applet codebase=http://raden.cs.ui.ac.id/farian/classes/menu  
code=peta.class width=50 height=50>
```

```
<param name=struktur  
value="http://raden.cs.ui.ac.id/farian/classes/menu/struktur.mttw">
```

```
<param name=image  
value="http://raden.cs.ui.ac.id/farian/images/rules_icon.gif">
```

```
<param name=target value="sender">
```

```
</applet>
```

LAMPIRAN B

FILE STRUKTUR.MTTW

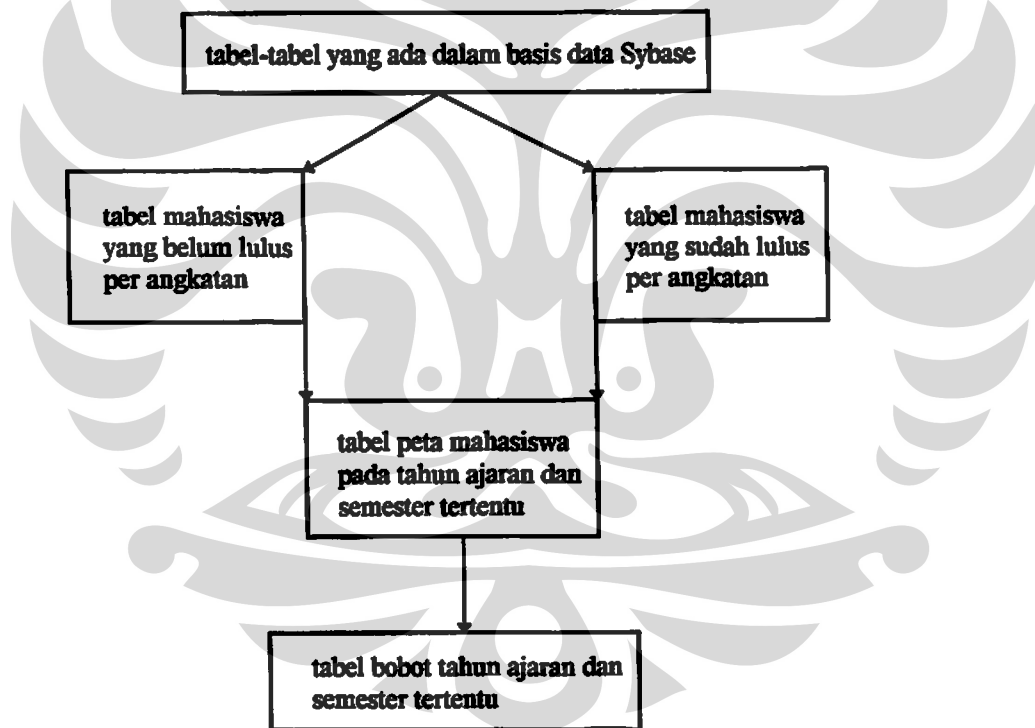
Applet kendali membaca *file* struktur.mttw. Pada *file* ini terdapat pendefinisian kata *main*, *subindeks*, dan *level*. Pendefinisian kata-kata inilah yang nantinya akan digunakan oleh *applet* kendali. Berikut adalah isi dari *file* struktur.mttw :

```
main$
"http://raden.cs.ui.ac.id/cgi-farian/mttw"
subindeks$
"(http://raden.cs.ui.ac.id/cgi-farian/tabel_93_94_1)=(PETA MAHASISWA Tahun Ajaran 93/94 Semester Gasal)"
"(http://raden.cs.ui.ac.id/cgi-farian/tabel_93_94_2)=(PETA MAHASISWA Tahun Ajaran 93/94 Semester Genap)"
level$
0*
"http://raden.cs.ui.ac.id/cgi-farian/mttw"
1*
"http://raden.cs.ui.ac.id/cgi-farian/tabel_93_94_1"
"http://raden.cs.ui.ac.id/cgi-farian/tabel_93_94_2"
2*
"http://raden.cs.ui.ac.id/cgi-farian/satu"
"http://raden.cs.ui.ac.id/cgi-farian/sudah_lulus_89"
...
"http://raden.cs.ui.ac.id/cgi-farian/sudah_lulus_92"
"http://raden.cs.ui.ac.id/cgi-farian/total89"
...
"http://raden.cs.ui.ac.id/cgi-farian/total192"
3*
"http://raden.cs.ui.ac.id/cgi-farian/duatiga"
"http://raden.cs.ui.ac.id/cgi-farian/tiga"
4*
"http://raden.cs.ui.ac.id/cgi-farian/empat"
5*
"http://raden.cs.ui.ac.id/cgi-farian/lima"
6*
"http://raden.cs.ui.ac.id/cgi-farian/evaluasi"
7*
"http://raden.cs.ui.ac.id/cgi-farian/distribusi"
"http://raden.cs.ui.ac.id/cgi-farian/frekuensi"
```


LAMPIRAN C

STORE PROCEDURE APLIKASI MTTW

Store procedure yang ada pada aplikasi MTTW dibuat dengan tujuan membangun tabel mahasiswa yang belum dan sudah lulus, tabel peta mahasiswa, beserta tabel bobot tahun ajaran dan semester tertentu. Tabel-tabel ini dibangun dari tabel-tabel yang sudah ada dalam basis data Sybase. Berikut adalah gambaran bagaimana hierarki tabel-tabel tersebut dibangun :



Tabel-tabel tersebut dibuat dengan *store procedure* yang sudah ada. *Store procedure* yang ada dibagi atas dua yaitu *store procedure* untuk membuat tabel dan *store procedure* untuk mengisi tabel.

Pertama-tama pembuat aplikasi harus membangun semua tabel-tabel yang diinginkan. Kemudian pembuat aplikasi dapat mengisi tabel-tabel tersebut dengan mengetikkan perintah *store procedure* pengisian pada masing-masing tabel.

1. *Store procedure* pembuat tabel

A. Pembuatan tabel mahasiswa yang belum dan sudah lulus.

```
create procedure create_tabel_ [angkatan]
as
create table lulus_ [angkatan]
(semester int, npm char(10), nama varchar(25))
create table kuliah_ [angkatan]
(npm char(10), nama varchar(25))
```

Field tabel mahasiswa yang sudah lulus tiap angkatan

Semester	NPM	Nama
----------	-----	------

Field tabel mahasiswa yang belum lulus tiap angkatan

NPM	Nama
-----	------

B. Pembuatan tabel peta mahasiswa tahun ajaran dan semester tertentu.

```
create proc create_tabel_ [tahun ajaran]
as
create table tabel_ [tahun ajaran dan semester tertentu]
(angkatan int, total_siswa int, sem8 int, sem9 int, sem10 int,
sem11 int, sem12 int, sem13 int, sem14 int, sem15 int,
bobot varchar(5), score float)
create table tabel_ [tahun ajaran dan semester tertentu]
(angkatan int, total_siswa int, sem8 int, sem9 int, sem10 int,
sem11 int, sem12 int, sem13 int, sem14 int, sem15 int,
bobot varchar(5), score float)
```

Field tabel peta mahasiswa

Angkatan	Total Siswa	sem8	sem9	...	sem15	Bobot
----------	-------------	------	------	-----	-------	-------

C. Pembuatan tabel bobot tahun ajaran dan semester tertentu.

```
create proc create_tabel_utama
as
create table tabel_utama (tahun_ajaran varchar(15), bobot varchar(5))
```

Field tabel bobot tahun ajaran dan semester

Tahun Ajaran	Bobot
--------------	-------

2. Store procedure pengisi tabel

A. Pengisian tabel mahasiswa yang belum dan sudah lulus.

```
create procedure proc_tabel_[angkatan]
as
truncate table kuliah_[angkatan]
insert kuliah_[angkatan] select mh003t.kd_mhs,nm_mhs
from mh002t,mh003t where
mh003t.kd_mhs like "12[angkatan]%"
and tahun=(select max(tahun) from mh003t)
and sms=(select max(sms) from mh003t where
tahun=(select max(tahun) from mh003t))
and kd_st_akademis <= 2
and mh003t.kd_mhs=mh002t.kd_mhs

truncate table lulus_[angkatan]
insert lulus_[angkatan] select sms_ke,mh003t.kd_mhs,nm_mhs
from mh002t,mh003t where kd_st_akademis=5
and mh003t.kd_mhs like "128[angkatan]%"
and mh003t.kd_mhs=mh002t.kd_mhs
```

B. Pengisian tabel peta mahasiswa pada tahun ajaran dan semester tertentu.

```
create procedure update_[angkatan pada tahun ajaran dan semester
tertentu]
as
declare @total int, @temp_lulus int, @nilai float, @lulus int
select @lulus=0
select @nilai=0

select @total=count(*) from tabel_[tahun ajaran dan semester tertentu]
where angkatan=[angkatan]
```

```

if (@total=0)
begin
insert into tabel_[angkatan pada tahun ajaran dan semester
tertentu] values
([angkatan],0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

select @total=count(*) from kuliah_[angkatan]
select @total=@total+count(*) from lulus_[angkatan]
update tabel_[angkatan pada tahun ajaran dan semester
tertentu] set total_siswa=@total
where angkatan=[angkatan]

end

select @temp_lulus=count(*) from lulus_[angkatan] where
semester<=7
select @lulus=@lulus+@temp_lulus
select @nilai=@nilai+@temp_lulus*8
select @total=@total-@temp_lulus
update tabel_[tahun ajaran dan semester tertentu]
set sem8=@total where angkatan=[angkatan]

.
.
select @temp_lulus=count(*) from lulus_[angkatan] where
semester=[angkatan itu berada]
select @lulus=@lulus+@temp_lulus
select @nilai=@nilai+@temp_lulus*[semester angkatan itu berada]
select @total=@total-@temp_lulus
update tabel_[tahun ajaran dan semester tertentu]
set sem[angkatan itu berada]=@total where angkatan=[angkatan]

if (@lulus < . 0)
select @nilai=@nilai/@lulus

update tabel_[tahun ajaran dan semester tertentu]
set bobot=str(@nilai,5,2) where angkatan=[angkatan]
update tabel_[tahun ajaran dan semester tertentu]
set score=@nilai where angkatan=[angkatan]

```

C. Pengisian tabel bobot tahun ajaran dan semester tertentu.

```

create proc proc_tabel_utama
as
declare @nilai float

truncate table tabel_utama

repeat
select @nilai=avg(score) from tabel_[tahun ajaran dan semester tertentu]
insert into tabel_utama values ('[tahun ajaran dan semester tertentu] ',
str(@nilai,5,2))
until [tabel peta mahasiswa yang ada]

```