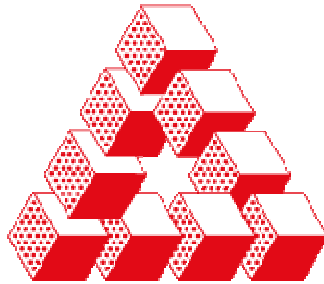
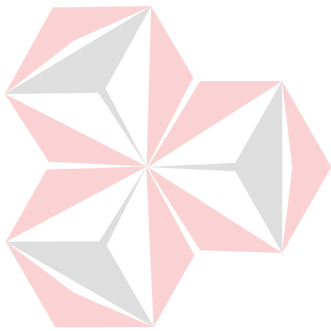


**IMPLEMENTASI SISTEM AUTENTIFIKASI TERINTEGRASI PADA
DOMAIN CONTROLLER DAN APPLICATION SERVER LABKOM
STIKOM SURABAYA**



**STIKOM
SURABAYA**



UNIVERSITAS
Dinamika

Oleh:

Nama : Diki Anggoro Putra

NIM : 07.41010.0195

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

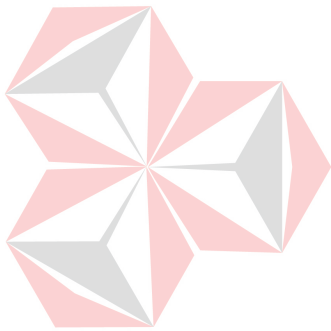
2011

**IMPLEMENTASI SISTEM AUTENTIFIKASI TERINTEGRASI PADA
DOMAIN CONTROLLER DAN APPLICATION SERVER LABKOM
STIKOM SURABAYA**

TUGAS AKHIR

Diajukan sebagai salah satu syarat menyelesaikan

Program Sarjana Komputer



UNIVERSITAS
Dinamika

Oleh:

Nama : Diki Anggoro Putra

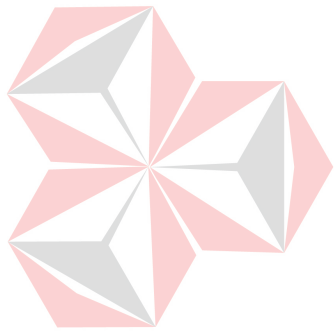
NIM : 07.41010.0195

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

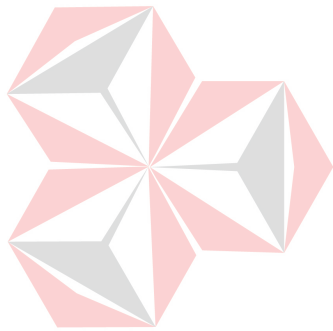
**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

2011



UNIVERSITAS
Dinamika

Dibuat untuk memenuhi salah satu syarat tujuan hidupku.....



UNIVERSITAS
Dinamika

Ditujukan untuk

Semua yang telah mengkritik, menempa dan menjadikanku lebih tegar

Terima kasih setulus-tulusnya...

Tugas Akhir
IMPLEMENTASI SISTEM AUTENTIFIKASI TERINTEGRASI PADA
DOMAIN CONTROLLER DAN APPLICATION SERVER LABKOM
STIKOM SURABAYA

dipersiapkan dan disusun oleh

Diki Anggoro Putra

NIM : 07.41010.0195

Telah diperiksa, diuji, dan disetujui oleh Dewan Penguji
pada: Mei 2011

Susunan Dewan Penguji

Pembimbing

I. Drs. Antok Suprianto, M.MT. _____

II. Kurniawan Jatmika, S.Kom _____

Penguji

I. Anjik Sukmaaji, S.Kom, M.Eng _____

II. Tutut Wuriyanto, M.Kom _____

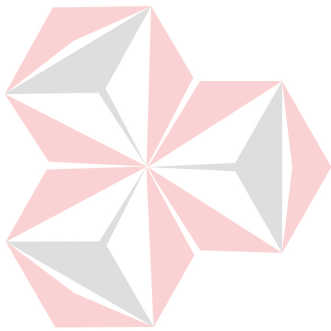
Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Pantjawati Sudarmaningtyas, S.Kom
Pembantu Ketua Bidang Akademik

PERNYATAAN

Dengan ini saya menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain dalam Tugas Akhir ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya. Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya Tugas Akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Surabaya, 31 Mei 2011



UNIVERSITAS
Dinamika
Diki Anggoro Putra
NIM : 07.41010.0195

ABSTRAKSI

Pada saat proses praktikum berlangsung, praktikan *login domain* untuk dapat masuk kedalam sistem praktikum yaitu PDC-Labkom, namun 1 (satu) komputer dapat digunakan oleh banyak praktikum pada sesi yang sama. Terdapat permasalahan yaitu 1 (satu) komputer dapat digunakan untuk *multi account* pada saat mengakses ke *domain controller*, dan tidak adanya pencatatan *management login* yang dapat memantau histori praktikan pada saat praktikum. Sehingga dapat terjadi kecurangan praktikan pada saat *login* PDC-Labkom.

Single sign on (SSO) adalah teknologi yang mengizinkan pengguna jaringan agar dapat mengakses sumber daya dalam jaringan hanya dengan menggunakan satu akun pengguna saja. Seorang pengguna hanya cukup melakukan proses autentifikasi sekali saja untuk mendapat izin akses terhadap semua layanan yang terdapat di dalam jaringan. Untuk layanan aplikasi digunakan *web service*. *Web service* merupakan kumpulan aplikasi logika yang menyediakan data dan *service* bagi aplikasi yang menggunakannya.

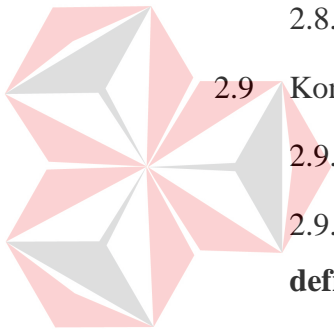
Implementasi *login* PDC-Labkom berbasis *web service* yang menggunakan metode SSO sebagai sarana autentifikasi terintegrasi pada *domain controller* dan *application server* pada sistem praktikum Labkom STIKOM Surabaya. SSO ini dapat menangani pengamanan *multi account* pada proses praktikum. Labkom dapat juga dapat melakukan pencatatan histori praktikan maupun histori kecurangan *multi account* dan pengamanan *management login*

Kata kunci: *Multi Account, Single Sign On, autentifikasi*

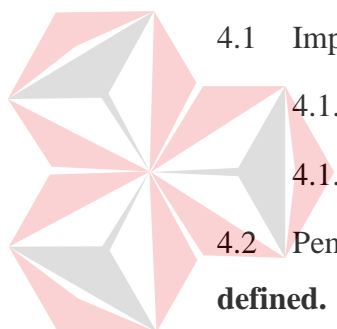
DAFTAR ISI

	Halaman
ABSTRAKSI	Error! Bookmark not defined.
KATA PENGANTAR	Error! Bookmark not defined.
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	17
BAB I PENDAHULUAN.....	Error! Bookmark not defined.
1.1 Latar Belakang Masalah.....	Error! Bookmark not defined.
1.2 Perumusan Masalah.....	Error! Bookmark not defined.
1.3 Batasan Masalah.....	Error! Bookmark not defined.
1.4 Tujuan.....	Error! Bookmark not defined.
1.5 Manfaat Penelitian.....	Error! Bookmark not defined.
1.6 Sistematika Penulisan.....	Error! Bookmark not defined.
BAB II LANDASAN TEORI.....	Error! Bookmark not defined.
2.1 Konsep Dasar Keamanan Informasi.....	Error! Bookmark not defined.
2.1.1 Confidentiality	Error! Bookmark not defined.
2.1.2 Integrity.....	Error! Bookmark not defined.
2.1.3 Availability	Error! Bookmark not defined.
2.2 Kontrol Akses.....	Error! Bookmark not defined.
2.2.1 Identifikasi	Error! Bookmark not defined.
2.2.2 Otentifikasi	Error! Bookmark not defined.
2.2.3 Otorisasi.....	Error! Bookmark not defined.
2.3 Single Sign On (SSO).....	Error! Bookmark not defined.

2.4	Windows Server 2003 Enterprise Edition	Error! Bookmark not defined.
2.4.1	Kelebihan windows server 2003.....	Error! Bookmark not defined.
2.4.2	Konfigurasi domain	Error! Bookmark not defined.
2.5	Application Server.....	Error! Bookmark not defined.
2.6	Password.....	Error! Bookmark not defined.
2.6.1	Otentikasi password.....	Error! Bookmark not defined.
2.7	Konsep Dasar Sistem.....	Error! Bookmark not defined.
2.8	Analisis dan Perancangan Sistem...	Error! Bookmark not defined.
2.8.1	System flow	Error! Bookmark not defined.
2.8.2	Data flow diagram (DFD)...	Error! Bookmark not defined.
2.9	Konsep Dasar Basis Data	Error! Bookmark not defined.
2.9.1	Sistem basis data.....	Error! Bookmark not defined.
2.9.2	Database management system.....	Error! Bookmark not defined.
2.9.3	Bahasa-bahasa yang terdapat dalam DBMS.....	Error! Bookmark not defined.
2.9.4	Fungsi DBMS	Error! Bookmark not defined.
2.10.	Kakas Pemograman	Error! Bookmark not defined.
2.10.1	Definisi .NET.....	Error! Bookmark not defined.
2.10.2	.NET framework	Error! Bookmark not defined.
2.10.3	ASP .NET	Error! Bookmark not defined.
2.10.4	ADO .NET	Error! Bookmark not defined.
2.11	Web Service.....	Error! Bookmark not defined.
BAB III ANALISIS DAN PERANCANGAN SISTEM		Error! Bookmark not defined.



3.1	Analisis Permasalahan.....	Error! Bookmark not defined.
3.2	Model Pengembangan	Error! Bookmark not defined.
3.3	Perancangan Sistem.....	Error! Bookmark not defined.
3.3.1	System flow	Error! Bookmark not defined.
3.3.2	Data flow diagram (DFD)...	Error! Bookmark not defined.
3.3.3	Entity relationship diagram (ERD)...	Error! Bookmark not defined.
3.3.4	Struktur database	Error! Bookmark not defined.
3.3.5	Desain interface	Error! Bookmark not defined.
3.3.6	Desain uji coba	Error! Bookmark not defined.
BAB IV IMPLEMENTASI DAN EVALUASI.....		Error! Bookmark not defined.
4.1	Implementasi Sistem	Error! Bookmark not defined.
4.1.1	Kebutuhan perangkat keras.	Error! Bookmark not defined.
4.1.2	Kebutuhan perangkat lunak	Error! Bookmark not defined.
4.2	Pembuatan dan Implementasi Program	Error! Bookmark not defined.
4.3	Evaluasi Sistem	Error! Bookmark not defined.
4.3.1	Evaluasi hasil uji coba sistem.....	Error! Bookmark not defined.
4.3.2	Analisa hasil uji coba sistem.....	Error! Bookmark not defined.
BAB V PENUTUP.....		Error! Bookmark not defined.
5.1	Kesimpulan.....	Error! Bookmark not defined.
5.2	Saran	Error! Bookmark not defined.
DAFTAR PUSTAKA		Error! Bookmark not defined.
LAMPIRAN.....		Error! Bookmark not defined.



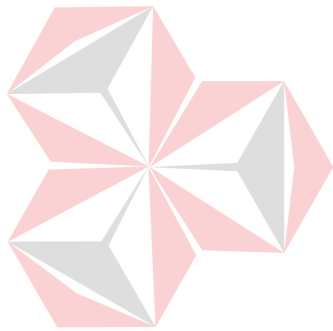
DAFTAR GAMBAR

Halaman

- Gambar 2.1. CIA TRIAD.....**Error! Bookmark not defined.**
- Gambar 2.2. *Arsitektur single sign on***Error! Bookmark not defined.**
- Gambar 2.3. Simbol-Simbol pada *System Flow* ..**Error! Bookmark not defined.**
- Gambar 2.4. Simbol-simbol pada DFD.....**Error! Bookmark not defined.**
- Gambar 3.1. *Block Diagram* Sistem SSO**Error! Bookmark not defined.**
- Gambar 3.2. Struktur proses *Single Sign On*.....**Error! Bookmark not defined.**
- Gambar 3.3. Desain arsitektur jaringan *Single Sign On*..... **Error! Bookmark not defined.**
- Gambar 3.4. Desain arsitektur *background Single Sign On*.....**Error! Bookmark not defined.**
- Gambar 3.5. *System Flow*Memelihara *Single Sign On* **Error! Bookmark not defined.**
- Gambar 3.6. *System Flow Single Sign On***Error! Bookmark not defined.**
- Gambar 3.7. *System Flow* Pelaporan**Error! Bookmark not defined.**
- Gambar 3.8. *Context Diagram* dari DFD.....**Error! Bookmark not defined.**
- Gambar 3.9. DFD *Level 0 Single Sign On***Error! Bookmark not defined.**
- Gambar 3.10. DFD *Level 1 Sub Proses Memelihara Data Master* **Error! Bookmark not defined.**
- Gambar 3.11. DFD *Level 1 Sub Proses Otorisasi Aplikasi* . **Error! Bookmark not defined.**
- Gambar 3.12. *Conceptual Data Model* (CDM) dari ERD ... **Error! Bookmark not defined.**
- Gambar 3.13. *Physical Data Model* (PDM) dari ERD **Error! Bookmark not defined.**
- Gambar 3.14. Desain *Form* Halaman Utama.....**Error! Bookmark not defined.**
- Gambar 3.15. Desain *Form* Buka Aplikasi.....**Error! Bookmark not defined.**
- Gambar 3.16. Desain *Form Master* Mahasiswa.....**Error! Bookmark not defined.**
- Gambar 3.17. Desain *Form Edit* Mahasiswa**Error! Bookmark not defined.**
- Gambar 3.18. Desain *Form Master* Hari**Error! Bookmark not defined.**
- Gambar 3.19. Desain *Form Edit* Hari**Error! Bookmark not defined.**

- Gambar 3.20. Desain *Form Master* Sesi**Error! Bookmark not defined.**
- Gambar 3.21. Desain *Form* Edit Sesi**Error! Bookmark not defined.**
- Gambar 3.22. Contoh Tampilan Rekap Data Histori Praktikum **Error! Bookmark not defined.**
- Gambar 3.23. Contoh Tampilan Rekap Data HistoriPelanggaran **Error! Bookmark not defined.**
- Gambar 4.1. Implementasi Jaringan SSO pada Labkom STIKOM Surabaya
.....**Error! Bookmark not defined.**
- Gambar 4.2. *Home Page (Admin)***Error! Bookmark not defined.**
- Gambar 4.3. Menu Utama Administrator**Error! Bookmark not defined.**
- Gambar 4.4. Menu Buka Aplikasi (*Admin*)**Error! Bookmark not defined.**
- Gambar 4.5. Pesan Data Nim Telah Diberikan Akses (Data Telah Tersimpan)
.....**Error! Bookmark not defined.**
- Gambar 4.6. Pesan Data Nim Tidak Ada Atau Tidak Melakukan Pelanggaran
.....**Error! Bookmark not defined.**
- Gambar 4.7. Menu *master (Admin)*.....**Error! Bookmark not defined.**
- Gambar 4.8. Menu *Master* Mahasiswa (*Admin*) ..**Error! Bookmark not defined.**
- Gambar 4.9. Menu *EditMaster* Mahasiswa (*Admin*) **Error! Bookmark not defined.**
- Gambar 4.10. Pesan Data Praktikan Telah Diberikan Akses (Data Telah Tersimpan).....**Error! Bookmark not defined.**
- Gambar 4.11. Pesan Data Praktikan Sudah Ada**Error! Bookmark not defined.**
- Gambar 4.12. Menu *Master* Hari (*Admin*)**Error! Bookmark not defined.**
- Gambar 4.13. Menu *Edit Master* Hari (*Admin*)**Error! Bookmark not defined.**
- Gambar 4.14. Menu Pesan Data Hari Telah Diberikan Akses (Data Telah Tersimpan).....**Error! Bookmark not defined.**
- Gambar 4.15. Pesan data hari tidak berhasil**Error! Bookmark not defined.**
- Gambar 4.16. Menu *Master* Sesi (*Admin*)**Error! Bookmark not defined.**
- Gambar 4.17. Menu *Edit Master* Sesi (*Admin*)**Error! Bookmark not defined.**
- Gambar 4.18. Pesan Data Sesi Telah Diberikan Akses (Data Telah Tersimpan)
.....**Error! Bookmark not defined.**
- Gambar 4.19. Pesan Data Sesi Tidak Berhasil.....**Error! Bookmark not defined.**

- Gambar 4.20. Menu Laporan (*Admin*)**Error! Bookmark not defined.**
- Gambar 4.21. Menu Histori Mahasiswa (*Admin*) ..**Error! Bookmark not defined.**
- Gambar 4.22. Menu Histori Kecurangan**Error! Bookmark not defined.**
- Gambar 4.23. Menu *About* (*Admin*)**Error! Bookmark not defined.**
- Gambar 4.24. Menu Utama (Praktikum).....**Error! Bookmark not defined.**
- Gambar 4.25. Menu Utama (Praktikan).....**Error! Bookmark not defined.**
- Gambar 4.26. Proses *background login* SSO (Praktikan).... **Error! Bookmark not defined.**
- Gambar 4.27. Halaman *Home* (Praktikan).....**Error! Bookmark not defined.**
- Gambar 4.28. Halaman Akses Praktikan Ditolak ..**Error! Bookmark not defined.**



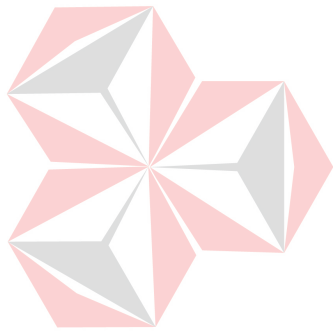
UNIVERSITAS
Dinamika

DAFTAR TABEL

Halaman

Tabel 3.1. Struktur Tabel Praktikan	Error! Bookmark not defined.
Tabel 3.2. Struktur Tabel Hari	Error! Bookmark not defined.
Tabel 3.3. Struktur Tabel Sesi.....	Error! Bookmark not defined.
Tabel 3.4. Struktur Tabel Pelanggaran.....	Error! Bookmark not defined.
Tabel 3.5. Struktur Tabel Logging	Error! Bookmark not defined.
Tabel 3.6. Fungsi-Fungsi Obyek Desain <i>Form</i> Halaman Utama.....	Error! Bookmark not defined.
Tabel 3.7. Fungsi-Fungsi Obyek Desain Buka Aplikasi....	Error! Bookmark not defined.
Tabel 3.8. Fungsi-Fungsi Obyek Desain <i>Master</i> Mahasiswa ...	Error! Bookmark not defined.
Tabel 3.9. Fungsi-Fungsi Obyek Desain <i>Edit</i> Mahasiswa .	Error! Bookmark not defined.
Tabel 3.10. Fungsi-Fungsi Obyek Desain <i>Master</i> Hari	Error! Bookmark not defined.
Tabel 3.11. Fungsi-Fungsi Obyek Desain <i>Edit</i> Mahasiswa .	Error! Bookmark not defined.
Tabel 3.12. Fungsi-Fungsi Obyek Desain <i>Master</i> sesi	Error! Bookmark not defined.
Tabel 3.13. Fungsi-Fungsi Obyek Desain <i>Edit</i> sesi	Error! Bookmark not defined.
Tabel 3.14. Data <i>Logging</i>	Error! Bookmark not defined.
Tabel 3.15. <i>Test Case</i> Data <i>Logging</i>	Error! Bookmark not defined.
Tabel 3.16. Data Praktikan.....	Error! Bookmark not defined.
Tabel 3.17. <i>Test Case</i> Data Praktikan	Error! Bookmark not defined.
Tabel 3.18. Data Hari	Error! Bookmark not defined.
Tabel 3.19. <i>Test Case</i> Data Hari	Error! Bookmark not defined.
Tabel 3.20. Data Sesi	Error! Bookmark not defined.
Tabel 3.21. <i>Test Case</i> Data Sesi.....	Error! Bookmark not defined.
Tabel 4.1. Data Nim	Error! Bookmark not defined.

Tabel 4.2. *Test Case* Data Nim**Error! Bookmark not defined.**
Tabel 4.3. Data Praktikan.....**Error! Bookmark not defined.**
Tabel 4.4. *Test Case* Data Praktikan**Error! Bookmark not defined.**
Tabel 4.5. *Test Case* Data Hari**Error! Bookmark not defined.**
Tabel 4.6. *Test Case* Data Sesi.....**Error! Bookmark not defined.**
Tabel 4.7. Data Praktikan.....**Error! Bookmark not defined.**
Tabel 4.8. *Test Case* Data Praktikan**Error! Bookmark not defined.**
Tabel 4.9. Tabel kuesioner**Error! Bookmark not defined.**
Tabel 4.10. Tabel rangkuman hasil kuesioner**Error! Bookmark not defined.**

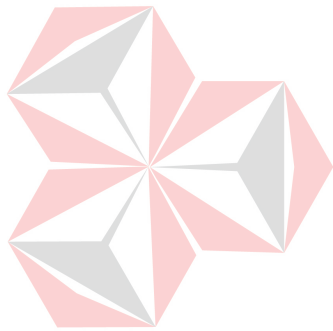


UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Halaman

Lampiran 1 Biodata Penulis	Error! Bookmark not defined.
Lampiran 2 Laporan Histori Praktikan	Error! Bookmark not defined.
Lampiran 3 Laporan Histori Pelanggaran	Error! Bookmark not defined.
Lampiran 4 Kuesioner dan Angket Tugas Akhir ...	Error! Bookmark not defined.
Lampiran 5 Source Code.....	Error! Bookmark not defined.



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Praktikum adalah sebuah pembelajaran kuliah yang dilakukan di laboratorium komputer (Labkom) dan diharapkan dapat menerapkan ilmu yang telah didapat di pembelajaran kuliah di kelas (Mfatihhurrizqi, 2010). Praktikum di STIKOM merupakan mata kuliah (MK) dengan bobot sebesar 1 (satu) sks yang dimaksudkan untuk membekali mahasiswa mengaplikasikan ilmu dan pengetahuan yang diperoleh saat pembelajaran dikelas, melalui kegiatan praktikum ini mahasiswa dapat mengerti bagaimana cara mengaplikasikan konsep yang didapat dikelas dengan mempraktekkan pada saat di Labkom.

Praktikum berlangsung mulai minggu ke 4 (empat) sampai dengan minggu ke 11 (sebelas) perkuliahan, dan pada minggu ke 13 (tiga belas) akan diadakan ujian praktikum. Labkom telah menerapkan konsep *paper less* dalam modul praktikum maupun soal-soal praktikum. Dengan adanya konsep *paper less* ini maka praktikan dapat *download* langsung untuk mendapatkan modul praktikum maupun soal-soal praktikum. Model pembelajaran praktikum ini dibagi menjadi 3 (tiga) bagian yaitu tes awal, tugas praktikum, dan ujian praktikum. Tes awal merupakan 5 (lima) soal pilihan ganda yang berisi tentang materi yang akan digunakan pada saat praktikum tersebut berlangsung, sedangkan model tes awal adalah praktikan STIKOM diberikan 5 (lima) pertanyaan optional secara acak antara praktikan 1 (satu) dengan yang lainnya. Tugas praktikum adalah tugas evaluasi yang diberikan untuk praktikan menjelang akhir praktikum sesuai materi

pada setiap pertemuan mingguan. Setelah selesai mengerjakan tugas praktikum, maka praktikan akan *upload* jawaban ke *server* Labkom. Untuk ujian praktikum praktikan akan diberikan sebuah soal individu yang harus dikerjakan oleh masing-masing praktikan STIKOM. Desain arsitektur pada Labkom adalah *client-server*, dimana *server* memiliki banyak *client* yang terhubung dengan menggunakan *domain* pada *server*. *Client* akan mengakses *application server* yang disebut PDC-Labkom. Materi maupun soal praktikum yang diberikan kepada praktikan berupa *softcopy* yang terdapat pada *server*, sehingga praktikan akan *download* materi maupun soal pada *server* Labkom. Satu praktikan dapat mengikuti beberapa praktikum, namun dengan satu *user* dan *password* yang sama. Untuk memulai praktikum di Labkom praktikan harus memasukkan *user* dan *password* pada awal *login domain*. Lalu praktikan dapat akses *application server* yang disebut PDC-Labkom. Pada PDC-Labkom praktikan dapat melakukan tes awal, dan untuk soal praktikum para praktikan diharuskan *download* dari *server*. Dan yang terakhir para praktikan akan *upload* jawaban ke *server*.

Selama proses praktikum, praktikan akan *log in application server* untuk dapat masuk ke dalam sistem praktikum. Namun praktikan dapat *log in* lebih dari satu *user* pada komputer dan sesi yang sama pada saat melakukan proses praktikum. Permasalahannya adalah 1 (satu) komputer pada sesi yang sama dapat multi *account* praktikan pada saat akses PDC-Labkom. Sehingga praktikan dapat membuka akses *login* PDC-Labkom yang bukan milik dirinya sendiri. Hal ini dapat mengakibatkan praktikan salah satunya dapat *download* jawaban milik temannya yang sudah diupload. Praktikan juga dapat membuka tes awal milik temannya, sehingga praktikan tersebut mengetahui soal tes awal lebih dulu

dengan menggunakan *login* temannya. Saat ini Labkom belum dapat menangani masalah multi *account* tersebut, sehingga praktikan dapat melakukan kecurangan-kecurangan multi *account* pada saat praktikum. Dengan adanya indikasi praktikan dapat melakukan multi *account*, nilai yang didapat oleh praktikan bisa dikatakan ada yang tidak murni dari hasil kerja sendiri, dan dibutuhkan sistem yang dapat menangani masalah multi *account* tersebut.

Salah satu metode yang dapat menangani multi *account* ini adalah *Single sign on*. Menurut Hursti Jani (1997), *Single sign on* (SSO) adalah teknologi yang mengizinkan pengguna jaringan agar dapat mengakses sumber daya dalam jaringan hanya dengan menggunakan satu akun pengguna saja. Dengan menggunakan SSO, seorang pengguna hanya cukup melakukan proses autentikasi sekali saja untuk mendapatkan izin akses terhadap semua layanan yang terdapat di dalam jaringan. Sistem SSO ini dapat diterapkan pula pada konsep *multitasking*, jadi meskipun praktikan membuka beberapa layar *browser* maka SSO ini dapat menanganinya. Keuntungan menerapkan SSO ini dapat meminimalisir *input user* dan *password* yang berulang-ulang dalam kurun waktu tertentu.

Berdasarkan latar belakang di atas, dibuat *web service* teknologi SSO sebagai sarana autentifikasi terintegrasi pada *domain controller* dan *application server* pada sistem praktikum Labkom STIKOM Surabaya. SSO ini merupakan teknologi yang dapat menggunakan *user* dan *password* yang diambil dari *login domain*. Data *user* dan *password* tersebut dapat digunakan untuk mengakses PDC-Labkom sehingga tes awal, tugas praktikum serta ujian praktikum tanpa memasukkan kembali *user* dan *password* kembali. Setiap praktikan hanya dapat *log in application server* di satu komputer untuk sesi tertentu. Dengan cara ini

diharapkan dapat meminimalisasikan adanya kecurangan multi *account* praktikan pada saat melakukan kegiatan praktikum. Dengan adanya metode SSO ini, maka Labkom dapat meminimalisir kemungkinan praktikan akan melakukan kecurangan multi *account* pada saat praktikum. Karena SSO ini dapat berfungsi sebagai autentifikasi atau keamanan multi *account user* dan *password* yang hanya dapat digunakan oleh masing – masing praktikan STIKOM

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan maka dirumuskan permasalahan dalam Tugas Akhir ini, yaitu:

1. Bagaimana merancang dan implementasikan SSO pada aplikasi PDC-Labkom untuk menangani kecurangan multi *account* di Labkom STIKOM Surabaya.
2. Bagaimana merancang dan implementasikan pencatatan histori praktikan dan histori pencatatan pelanggaran Labkom STIKOM Surabaya.
3. Bagaimana mengintegrasikan *web service* pada PDC-Labkom Labkom STIKOM Surabaya.

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, ruang lingkup permasalahan hanya akan dibatasi pada :

1. Sistem SSO dibuat berbasis *web service*.
2. Data *user* diperoleh dari Labkom STIKOM Surabaya periode semester genap tahun akademik 2010.
3. Sistem ini terintegrasi dengan PDC-Labkom.

4. Sistem ini hanya untuk pembatasan akses praktikan Labkom STIKOM Surabaya pada saat praktikum.
5. Role yang digunakan pada sistem SSO Labkom:
 - a. Satu komputer untuk 1 (satu) praktikan pada satu sesi = benar.
 - b. Jika komputer telah digunakan oleh 1 (satu) praktikan pada satu sesi dan praktikan tersebut pindah komputer = salah.
 - c. Praktikan dapat berpindah komputer pada sesi yang sama, jika komputer tersebut bermasalah dan belum digunakan praktikan yang lainnya. = benar.
 - d. Jika 1 (satu) komputer pada sesi yang sama terdapat multi *account* = salah.
6. Bahasa pemrograman yang digunakan adalah Microsoft ASP.Net 2.0 dan databasenya menggunakan SQL Server 2005 Express.

1.4 Tujuan

Dengan mengacu pada perumusan masalah maka tujuan yang hendak dicapai dalam penyusunan Tugas Akhir ini adalah :

1. Mengimplementasikan SSO pada aplikasi PDC-Labkom untuk meminimalisir kecurangan multi *account* di Labkom STIKOM Surabaya.
2. Mengimplementasikan pencatatan histori praktikan dan histori pencatatan pelanggaran Labkom STIKOM Surabaya.
3. Mengintegrasikan *web service* pada *application server* Labkom STIKOM Surabaya.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memiliki beberapa nilai manfaat penulisan, antara lain :

1. Labkom dapat menggunakan sistem *single sign on* yang dapat menjadi salah satu keamanan pada saat praktikum di Labkom.
2. Sebagai salah satu sarana kontrol praktikan Labkom agar dapat melakukan praktikum dari hasil kerja masing-masing.

1.6 Sistematika Penulisan

Sistematika dalam penyusunan Tugas Akhir ini akan dijabarkan dalam setiap bab dengan pembagian sebagai berikut :

BAB I : Pendahuluan

Pada bab ini akan dibahas latar belakang masalah, permasalahan yang ada, batasan masalah serta sistematika penulisan yang berisi penjelasan singkat pada masing-masing bab.

BAB II : Landasan Teori

Pada bab ini dijelaskan landasan teori yang merupakan teori dasar dari teori yang dipakai untuk menyelesaikan permasalahan.

BAB III : Analisis dan Perancangan Sistem

Bab ini membahas tentang analisis dan perancangan sistem, yaitu menganalisis masalah, *System Flow* Terkomputerisasi, *Document Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), struktur tabel, *Interface* dan Desain Uji Coba dan Analisis

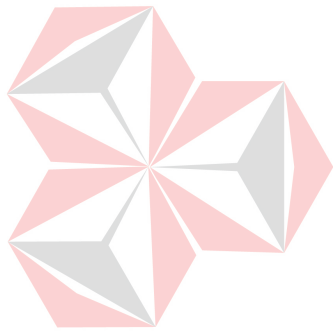
BAB IV : Implementasi dan Evaluasi

Pada bab ini akan dibahas tentang cara penggunaan sistem yaitu merupakan hasil rancangan dengan menggunakan data yang dibutuhkan dan pengujian dari program yang telah dibuat. Pengujian

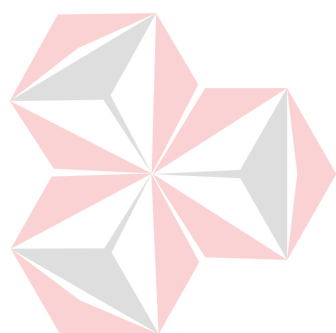
akan dilakukan untuk memastikan apakah program yang dibuat sudah sesuai dengan yang dikehendaki.

BAB V : Penutup

Pada bab ini dibahas tentang kesimpulan dan saran dari penggunaan program aplikasi dan saran pengembangan selanjutnya.



UNIVERSITAS
Dinamika



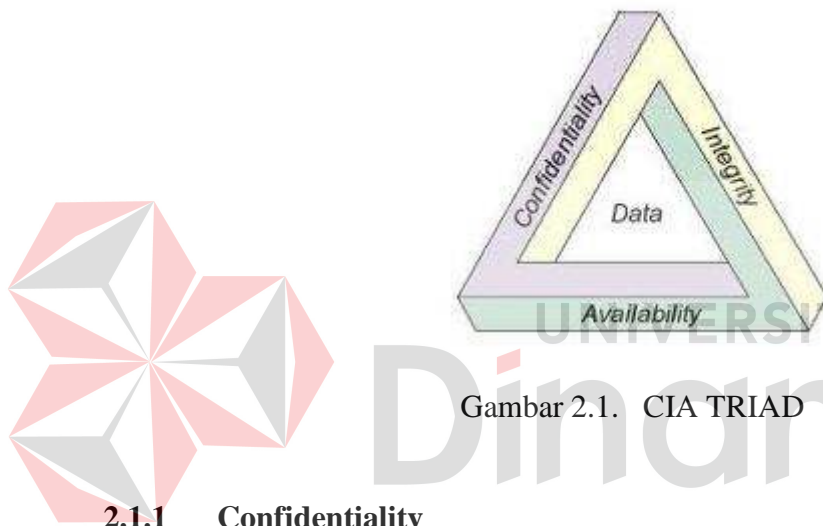
UNIVERSITAS
Dinamika

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Keamanan Informasi

Selama lebih dari 20 tahun, keamanan informasi telah dibangun atas 3 (tiga) kunci dasar dari prinsip kunci keamanan informasi yaitu : *confidentiality* (kerahasiaan), *integrity* (integritas), dan *availability* (ketersediaan) (Dani, 2008).



Gambar 2.1. CIA TRIAD

2.1.1 Confidentiality

Confidentiality (kerahasiaan) berfokus pada upaya untuk menghindari pengungkapan secara tidak sah terhadap informasi yang bersifat rahasia maupun sensitif. Pengungkapan informasi tersebut dapat terjadi secara disengaja, seperti pemecahan sandi untuk membaca informasi, atau dapat terjadi secara tidak disengaja, dikarenakan kecerobohan dari individu dalam menangani informasi.

Sejumlah mekanisme yang sering digunakan untuk mempertahankan konsep *Confidentiality* meliputi (Dani, 2008) :

1. Klasifikasi Data

Merupakan proses pelabelan informasi sehingga masing-masing individu mengetahui siapa yang diizinkan untuk melihatnya dan siapa yang tidak.

2. Enkripsi

Merupakan mekanisme teknis yang digunakan untuk menjaga kerahasiaan (*confidentiality*).

3. Pemusnahan Peralatan (*Equipment Disposal*)

Merupakan segala bentuk usaha / aktifitas yang ditujukan untuk melindungi kerahasiaan suatu informasi ketika tidak lagi dipergunakan dalam media penyimpanan. Beberapa contoh aksi dalam hal ini adalah proses format pada disk sekurang-kurang 7 kali atau lebih, penyobekan kertas (dengan bantuan mesin *shredder*), dan lain sebagainya.

2.1.2 Integrity

Dalam keamanan informasi, *integrity* (integritas atau keutuhan) berarti bahwa data tidak dapat dibuat, diganti, atau dihapus tanpa proses otorisasi. Dengan kata lain, *integrity* merupakan prinsip yang ditujukan untuk menjaga keakuratan suatu informasi. Sebagai contoh, data yang disimpan pada salah satu bagian dari sistem *database* telah melewati persetujuan dengan data terkait yang tersimpan pada bagian lain dari sistem *database*. Adapun tujuan dari *integrity* adalah (Dani, 2008) :

1. Menghindari modifikasi informasi dari *user* atau pengguna yang tidak berhak.
2. Menghindari akses yang tidak sah atau modifikasi informasi yang tidak disengaja dari pengguna yang tidak berhak.
3. Pemeliharaan terhadap konsistensi internal dan eksternal.

- a. Konsistensi internal memastikan bahwa data internal tetap konsisten. Sebagai contoh, pada suatu *database* organisasi, jumlah item yang dimiliki oleh suatu organisasi harus sama dengan jumlah item yang ditampilkan pada *database*.
- b. Konsistensi eksternal menjamin bahwa data yang disimpan pada *database* konsisten dengan dunia nyata. Serupa dengan contoh sebelumnya, jumlah item yang ada secara fisik pada dunia nyata harus sama dengan jumlah item yang terdapat pada *database*.

Beragam usaha yang dapat dilakukan untuk menjaga integritas terhadap suatu data atau informasi meliputi :

1. *Checksums*

Merupakan serangkaian angka yang dihasilkan melalui fungsi matematika untuk memastikan bahwa blok data yang diberikan tidak berubah.

2. Kontrol Akses

Merupakan mekanisme untuk memastikan bahwa individu / pihak tertentu dapat hanya melakukan sejumlah aksi tertentu.

2.1.3 Availability

Availability (ketersediaan) menjamin bahwa pengguna sistem yang berhak memiliki akses tanpa interupsi terhadap sistem dan jaringan. Hal tersebut memastikan bahwa informasi atau sumber daya akan selalu tersedia ketika dibutuhkan.

Bentuk – bentuk usaha yang dapat dilakukan untuk menjaga ketersediaan data meliputi (Dani, 2008) :

1. *Redundant systems* atau implementasi sistem berganda ke dalam suatu infrastruktur (seperti *disk array* atau mesin-mesin yang di-*cluster*).
2. Perangkat lunak anti virus untuk menghentikan *worm* atau program berbahaya lainnya yang mengganggu kondisi jaringan.
3. Penerapan perangkat IPS guna mengantisipasi ancaman serangan tertentu (seperti DDoS) yang dapat mengganggu ketersediaan suatu layanan.

2.2 Kontrol Akses

Akses terhadap informasi yang dilindungi harus dibatasi kepada individu-individu yang berhak mengakses informasi tersebut. Program komputer, dan komputer yang memproses informasi juga harus dilindungi. Hal ini tentunya membutuhkan mekanisme pada tempatnya untuk mengontrol akses terhadap informasi yang dilindungi tersebut. Dalam implementasinya, mekanisme kontrol akses hendaknya seimbang dengan nilai informasi yang dilindungi. Fondasi dasar mekanisme kontrol akses dibangun atas mekanisme Identifikasi dan otentifikasi (Dani, 2008).

2.2.1 Identifikasi

Identifikasi merupakan pernyataan siapakah seseorang tersebut atau apakah sesuatu tersebut. Jika seseorang membuat pernyataan “Hello, my name is John Doe”, maka ia membuat klaim atas jati dirinya. Namun, klaim tersebut bisa berarti benar atau sebaliknya. Sebelum John Doe diberikan akses terhadap informasi yang dilindungi, maka akan menjadi penting untuk dipastikan bahwa seseorang yang mengklaim sebagai John Doe tersebut adalah benar John Doe (Dani, 2008).

2.2.2 Otentifikasi

Otentifikasi tidak lain adalah metode verifikasi atas identitas user, proses-proses, dan peranti-peranti. Verifikasi identitas akan berlaku untuk pengirim maupun penerima informasi (Rafiudin, 2005).

1. *What a Person Knows* (apa yang diketahui user)

Password dan *Personal Identification Number (PIN)* merupakan contoh *what a person knows*. *Password* mungkin digunakan berulang-ulang oleh *user* atau mungkin hanya sekali saja.

2. *What a Person Has* (apa yang dimiliki user)

Token-token (hardware atau software) termasuk kategori *what a person has*.

Smartcard, SecureID, CryptoCard, dan SafeWord adalah beberapa contoh *token*.

3. *What a Person Is* (Siapakah user)

Autentifikasi biometrik termasuk contoh *what a person is* karena identifikasi di dalamnya berdasarkan atribut-atribut fisik seorang *user*. Contoh sistem-sistem *biometrik* di antaranya adalah *plam scan, hand geometry, iris scan, retina pattern, fingerprint, voiceprint, facial recognition*, dan sistem-sistem *signature* dinamis.

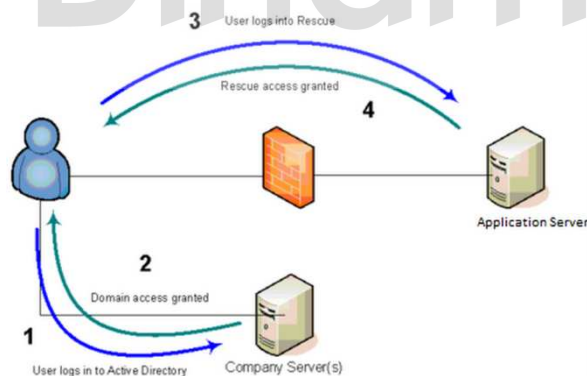
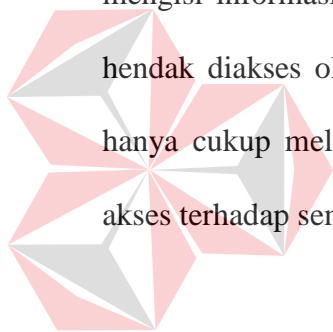
2.2.3 Otorisasi

Otorisasi adalah pemberian hak (*privilege*) melalui perancangan utiliti spesial untuk akses layanan-layanan atau informasi spesifik bagi *user* atau grup *user*. Di lingkungan sistem-sistem yang sifatnya publik, otorisasi terbuka untuk *user guest* atau *anonymous*.

Otorisasi tidaklah lain dari kunci untuk meyakinkan bahwa hanya user yang sah saja yang dapat mengakses informasi (Rafiudin, 2005).

2.3 Single Sign On (SSO)

Teknologi *Single-sign-on* (sering disingkat menjadi SSO) adalah teknologi yang mengizinkan pengguna jaringan agar dapat mengakses sumber daya dalam jaringan hanya dengan menggunakan satu akun pengguna saja. Teknologi ini sangat diminati, khususnya dalam jaringan yang sangat besar dan bersifat heterogen (di saat sistem operasi serta aplikasi yang digunakan oleh komputer adalah berasal dari banyak *vendor*, dan pengguna dimintai untuk mengisi informasi dirinya ke dalam setiap *platform* yang berbeda tersebut yang hendak diakses oleh pengguna). Dengan menggunakan SSO, seorang pengguna hanya cukup melakukan proses autentikasi sekali saja untuk mendapatkan izin akses terhadap semua layanan yang terdapat di dalam jaringan (Hursti, 1997).



Gambar 2.2. Arsitektur *single sign on*

Keuntungan sebuah sistem menggunakan SSO adalah sebagai berikut :

1. Mengurangi tingkat kejenuhan *user* dalam penggunaan *password*.
2. Mengurangi waktu yang digunakan untuk memastikan *password* untuk sebuah identitas yang sama.

3. Dapat mendukung otentifikasi konvensional seperti Windows *credential*.
4. Mengurangi biaya TI seiring dengan berkurangnya *user* yang meminta bantuan mengenai hal otentifikasi yang hal ini adalah permasalahan di *username* dan *password*.
5. Keamanan di semua level akses baik masuk maupun keluar sistem.

2.4 Windows Server 2003 Enterprise Edition

Windows Server jenis ini memang didesain untuk perusahaan berkala menengah ke atas. Windows Server 2003 Enterprise Edition hampir sama dengan Windows Server 2003 Standart Edition. Bedanya hanya terletak pada besarnya *database* yang mampu dikelola (Andi, 2004).

2.4.1 Kelebihan windows server 2003

Dalam Windows Server 2003 ini, semua konfigurasi dikunci. Hanya *administrator* saja yang dapat mengubah konfigurasi. Hal ini berbeda dengan versi – versi sebelumnya. Dengan sistem ini Windows Server 2003 memiliki kinerja, perangkat manajemen, dan sistem keamanan yang baik (Andi, 2004).

2.4.2 Konfigurasi domain

Sistem *domain* atau lebih dikenal dengan istilah *client server* adalah suatu sistem jaringan dimana semua *client* yang ada di jaringan tersebut diatur penuh oleh satu atau lebih *server*. Dalam sistem *domain*, *server* mempunyai hak akses penuh ke komputer lain. Dengan sistem ini semua data dapat dipusatkan ke *server* (Andi, 2004).

2.5 Application Server

Application server adalah *server software* yang memungkinkan *client* untuk mengakses aplikasi yang diinginkan. *Application server* akan mengurus semua operasi aplikasi, termasuk koneksi *database* atau *web service*. Dari sisi pengembangan aplikasi, hal ini jelas memudahkan pengembangan aplikasi. Urusan integrasi antar aplikasi bisa dimungkinkan dengan bantuan *application server*. Urusan keamanan juga telah diurus oleh *application server* tersebut. Masalah koneksi *database* atau sekedar menghadirkan *web service* juga bukan lagi masalah (Noprianto, 2005).

2.6 Password

Password bisa diartikan sebagai suatu bentuk dari data otentifikasi rahasia yang digunakan untuk mengontrol akses ke dalam suatu sumber informasi. *Password* akan dirahasiakan dari mereka yang tidak diizinkan untuk mengakses. Selain itu, bagi mereka yang ingin mengetahui akses tersebut akan diuji, apakah layak atau tidak untuk memperolehnya. Walaupun demikian, *password* bukan berarti suatu bentuk kata-kata. *Password* yang tidak berbentuk kata dan memiliki suatu arti akan lebih sulit untuk ditebak. *Password* kadang-kadang digunakan juga dalam suatu bentuk yang hanya berisi angka (numeric), salah satu contoh adalah *Personal Identification Number (PIN)* (Malik, 2009).

2.6.1 Otentikasi password

Pengesahan atau biasa disebut otentikasi merupakan hal yang sudah pasti di dalam suatu penggunaan *password*. Berikut beberapa penjelasan dari otentikasi *password* :

A. weak authentication

Secara umum, sistem dengan otentikasi yang lemah (*weak authentication*) dicirikan dengan protokol yang memiliki kebocoran *password* langsung di atas jaringan atau membocorkan informasi yang cukup untuk diketahui “penyerang” sehingga *password* dapat di analisis dan ditebak.

B. strong authentication

Walaupun enkripsi yang baik sudah ada sejak beberapa dekade yang lalu, pengembangan dari otentifikasi yang kuat (*strong authentication*) pada protokol langsung baru dimulai tahun 1990 dengan publikasi dari “*EKE family of algorithmn*”

C. inconvenient authentication

Ketidakhadiran otentikasi yang kuat atau adanya gangguan otentikasi (*inconvenient authentication*) membuat para desainer sistem tahun 1980-an mencoba teknik lain untuk menjamin keamanan *password*. Kebanyakan dari sistem yang ada, tidak sepenuhnya *password-based* dan sering membutuhkan sesuatu yang lebih pada bagian pengguna, *administrator*, atau keduanya, untuk mengoperasikan secara halus. Ada tiga metode yang dapat dilakukan, yaitu *one-time password*, *kerberos*, dan SSH.

2.7 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur sistem adalah sebagai berikut (Neuschel, 1976) :

“Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.”

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Prosedur (procedure) didefinisikan oleh Neuschel, 1976 sebagai berikut:

“Prosedur adalah suatu urutan operasi klerikal (tulis-menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.”

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya dalam mendefinisikan sistem (Neuschel,1976) adalah sebagai berikut:

“Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.”

Sedangkan menurut Kendall Sistem informasi didefinisikan sebagai berikut (Kendall & Kendall, 2003) :

“Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.”

2.8 Analisis dan Perancangan Sistem

Penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi

permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Kendall & Kendall, 2003).

Tahap analisis sistem dilakukan setelah tahap perencanaan sistem (*system planning*) dan sebelum tahap desain sistem (*system design*). Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini juga akan menyebabkan kesalahan di tahap selanjutnya.

Dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut:

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

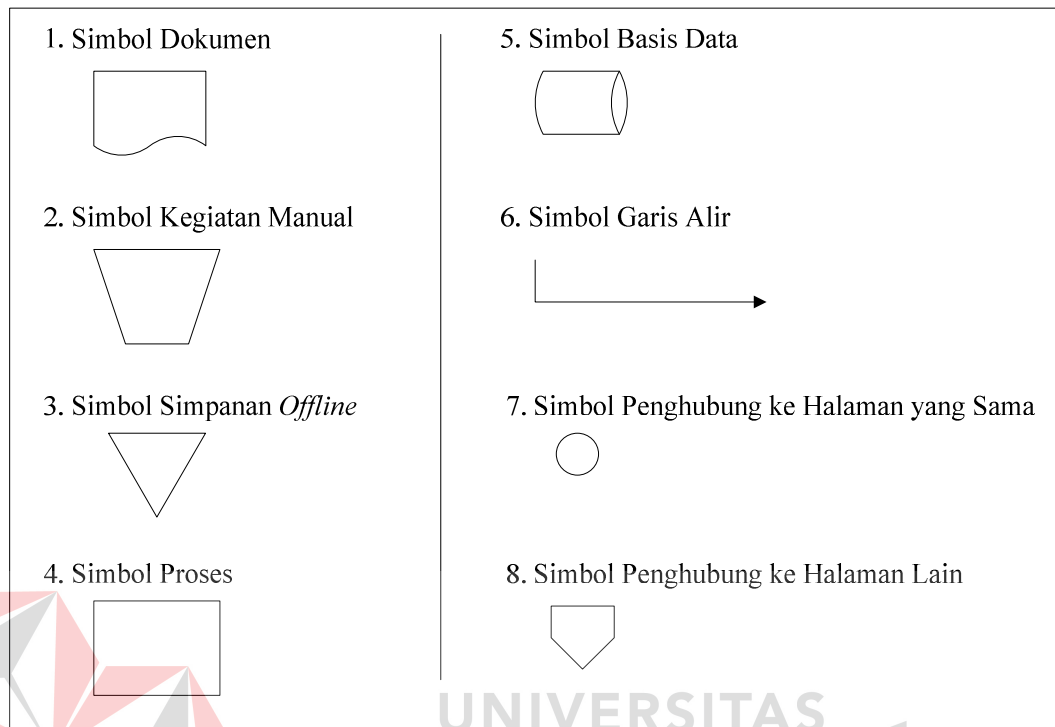
Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan perancangan sistem.

Analisis dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

2.8.1 System flow

System flow atau bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan urutan-urutan dari prosedur yang ada di dalam sistem dan menunjukkan apa yang

dikerjakan sistem. Simbol-simbol yang digunakan dalam *system flow* ditunjukkan pada Gambar 2.3 (Kendall & Kendall, 2003).



Gambar 2.3. Simbol-Simbol pada *System Flow*

1. Simbol dokumen

Menunjukkan dokumen input dan output baik untuk proses manual atau komputer.

2. Simbol kegiatan manual

Menunjukkan pekerjaan manual.

3. Simbol simpanan offline

Menunjukkan file non-komputer yang diarsip.

4. Simbol proses

Menunjukkan kegiatan proses dari operasi program komputer.

5. Simbol basis data

Menunjukkan tempat untuk menyimpan data hasil operasi komputer.

6. Simbol garis alir

Menunjukkan arus dari proses.

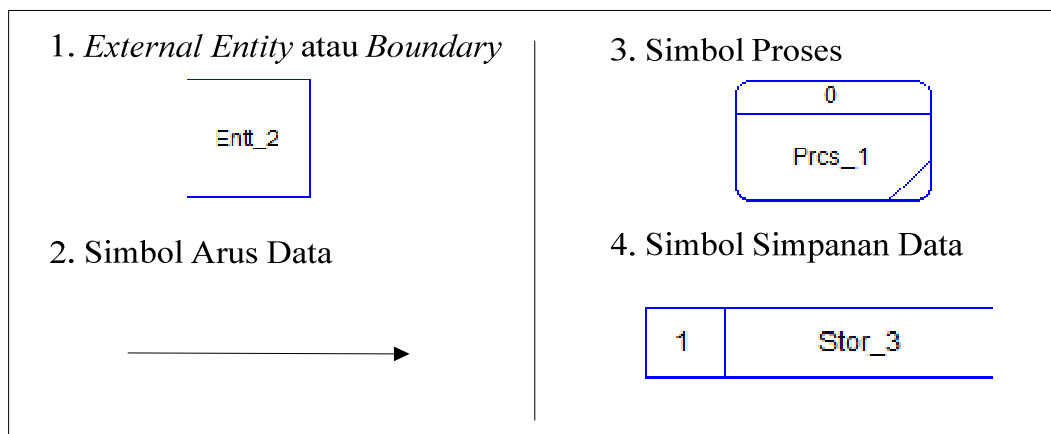
7. Simbol penghubung

Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.

2.8.2 Data flow diagram (DFD)

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas (Kendall & Kendall, 2003).

A. simbol – simbol yang digunakan dalam DFD



Gambar 2.4. Simbol-simbol pada DFD

1 *External Entity* atau *Boundary*

External entity atau kesatuan luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem. *External entity* disimbolkan dengan notasi kotak.

2 Arus Data

Arus Data (*data flow*) di DFD diberi simbol panah. Arus data ini mengalir di antara proses, simpanan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3 Proses

Suatu proses adalah kegiatan yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk menghasilkan arus data yang akan keluar dari proses. Simbol proses berupa lingkaran atau persegi panjang bersudut tumpul.

4 Simpanan Data

Simpanan data merupakan simpanan dari data yang dapat berupa hal-hal sebagai berikut, sebagai gambaran:

1. Suatu file atau *database* di sistem komputer.
2. Suatu arsip atau catatan manual.
3. Suatu kotak tempat data di meja seseorang.
4. Suatu tabel acuan manual.

Simpanan data di DFD disimbolkan dengan sepasang garis horizontal paralel yang tertutup di salah satu ujungnya.

B. context diagram

Context Diagram merupakan langkah pertama dalam pembuatan DFD. Pada context diagram dijelaskan sistem apa yang dibuat dan eksternal entity apa saja yang terlibat. Dalam context diagram harus ada arus data yang masuk dan arus data yang keluar.

C. data flow diagram level 0

DFD level 0 adalah langkah selanjutnya setelah *context diagram*. Pada langkah ini, digambarkan proses-proses yang terjadi dalam sistem informasi.

D. data flow diagram level 1

DFD Level 1 merupakan penjelasan dari DFD level 0. Pada proses ini dijelaskan proses apa saja yang dilakukan pada setiap proses yang terdapat di DFD level 0.

E. entity relational diagram

Entity Relational Diagram (ERD) merupakan penggambaran hubungan antara beberapa entity yang digunakan untuk merancang *database* yang akan diperlukan.

2.9 Konsep Dasar Basis Data

Database merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara *database* Relasional dan Non Relasional. Pada *database* Non Relasional, sebuah *database* hanya merupakan sebuah file (Yuswanto & Subari, 2005).

Database adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya (Marlinda, 2004).

Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, *multiple user* (banyak pemakai), *security* (masalah keamanan), masalah integrasi (kesatuan), dan masalah *data independence* (kebebasan data).

2.9.1 Sistem basis data

Menurut Marlinda (2004), sistem basis data adalah suatu sistem menyusun dan mengelola *record* menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu Perangkat Keras (*Hardware*), Sistem Operasi (*Operating System*), Basis Data (*Database*), Sistem (Aplikasi atau Perangkat Lunak) Pengelola Basis Data (DBMS), Pemakai (*User*), dan Aplikasi (Perangkat Lunak) lain (bersifat opsional).

Kelebihan Sistem Basis Data :

1. Mengurangi kerangkapan data, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga update dilakukan berulang-ulang.

2. Mencegah ketidak konsistenan.
3. Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang.
4. Integritas dapat dipertahankan.
5. Data dapat dipergunakan bersama-sama.
6. Menyediakan *recovery*.
7. Memudahkan penerapan standarisasi.
8. Data bersifat mandiri (*data independence*).
9. Keterpaduan data terjaga, memelihara keterpaduan data berarti data harus akurat. Hal ini sangat erat hubungannya dengan pengontrolan kerangkapan data dan pemeliharaan keselarasan data.

Kekurangan Sistem Basis Data

1. Diperlukan tempat penyimpanan yang besar.
2. Diperlukan tenaga yang terampil dalam mengolah data.
3. Kerusakan sistem basis data dapat mempengaruhi departemen yang terkait.

2.9.2 Database management system

Menurut Marlinda (2004), *Database Management System (DBMS)* merupakan kumpulan *file* yang saling berkaitan dan program untuk pengelolanya. Basis Data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data.

2.9.3 Bahasa-bahasa yang terdapat dalam DBMS

1 *Data Definition Language (DDL)*

Pola skema basis data dispesifikasikan dengan satu set definisi yang diekspresikan dengan satu bahasa khusus yang disebut DDL. Hasil kompilasi perintah DDL adalah satu set tabel yang disimpan di dalam file khusus yang disebut *data dictionary/directory*.

2 *Data Manipulation Language (DML)*

Bahasa yang memperbolehkan pemakai mengakses atau memanipulasi data sebagai yang diorganisasikan sebelumnya model data yang tepat.

3 *Query*

Pernyataan yang diajukan untuk mengambil informasi. Merupakan bagian DML yang digunakan untuk pengambilan informasi.

2.9.4 Fungsi DBMS

1 *Data Definition*

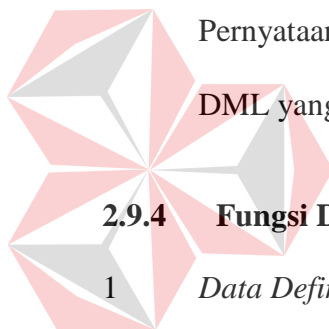
DBMS harus dapat mengolah *data definition* atau pendefinisian data.

2 *Data Manipulation*

DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data.

3 *Data Security dan Integrity*

DBMS dapat memeriksa security dan integrity data yang didefinisikan oleh DBA.



UNIVERSITAS
Dinamika

4 *Data Recovery dan Concurrency*

4.1 DBMS harus dapat menangani kegagalan-kegagalan pengaksesan basis data yang dapat disebabkan oleh kesalahan sistem, kerusakan *disk*, dan sebagainya.

4.2 DBMS harus dapat mengontrol pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat yang bersamaan.

5 *Data Security dan Integrity*

DBMS harus menyediakan data dictionary atau kamus data.

2.10. Kakas Pemrograman

Dalam pengembangan suatu sistem informasi, tentunya membutuhkan suatu kakas atau alat berupa bahasa pemrograman. Salah satu kakas dalam bahasa pemrograman yang sekarang dipakai adalah keluarga Microsoft Visual Studio 2005 yang menggunakan teknologi .NET (Danny & Tommy, 2002).

2.10.1 Definisi .NET

DOT NET (.NET) framework adalah suatu platform baru di dalam pemrograman untuk lingkungan yang terdistribusi luas (internet). Istilah .NET sering diasosiasikan dengan proses yang berjalan pada platform .NET.

Salah satu bentuk keunggulan dari platform ini terrefleksi pada kompilasi sumber kode program, dimana semua sumber kode program akan dikompilasi menjadi *Microsoft Intermediate Language* (MSIL). Selanjutnya MSIL akan dikompilasikan oleh *.NET Compiler* menjadi bahasa mesin pada saat akan digunakan.

.NET merupakan alat untuk mewujudkan visi Microsoft pada jaringan internet dengan membentuk jaringan global yang saling berinteraksi agar dapat memberi pelayanan dan pertukaran data dengan cara yang lebih efisien dan terjamin dari segi keamanan (Danny & Tommy, 2002).

2.10.2 .NET framework

Microsoft .NET Framework adalah produk software yang merupakan inti dari .NET teknologi. Produk ini bekerja secara terintegrasi dengan produk Microsoft lainnya, misalnya *Internet Information Service (IIS)*. Ia terdiri dari beberapa modul seperti salah satu contohnya adalah ASP .NET. ASP .NET inilah yang digunakan untuk mengembangkan sistem informasi dalam bahasan kali ini (Danny & Tommy, 2002).

2.10.3 ASP .NET

ASP .NET merupakan hasil pengembangan lebih lanjut dari ASP (Active Server Page), tetapi ia berbeda dari ASP, karena ASP .NET dibuat dengan dasar pemikiran yang berbeda sehingga program ASP tidak dapat dijalankan sebagai program ASP .NET . VBScript tidak lagi digunakan pada ASP .NET, sebagai gantinya anda dapat menggunakan VB .NET, C#, atau bahasa pemrograman lainnya. Penggunaan bahasa yang berbeda ini dimungkinkan karena ASP .NET mengadopsi konsep multi-language dalam pengembangan aplikasi program.

Pada dasarnya ASP .NET dapat direpresentasikan sebagai suatu tingkatan (*hierarchy*) *classes* atau kelas pemrograman yang menyediakan layanan dasar. Program ASP .NET mengandalkan penggunaan *Namespace* sebagai *Application*

Program Interface (API). *NameSpace* adalah skema penamaan untuk mengelompokkan tipe yang saling berhubungan (Danny & Tommy, 2002).

2.10.4 ADO .NET

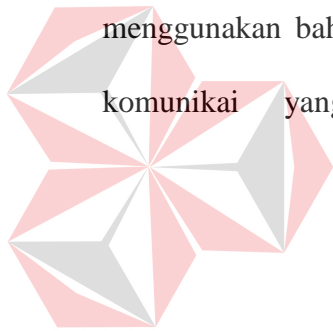
ADO .NET sebagai modul yang bekerja pada disconnected fashion (lingkungan terputus), seperti pada layanan web adalah komponen kunci untuk mengakses sumber data (*database*) untuk memperoleh baris data atau memanipulasi *database*. Ia merupakan pengembangan lebih lanjut dari ADO (ActiveX Data Objects). Bersama dengan ASP .NET, ia memungkinkan terbentuknya halaman web yang dinamis (halaman web yang berubah-ubah tergantung pada inputan pengguna) (Danny & Tommy, 2002).

2.11 Web Service

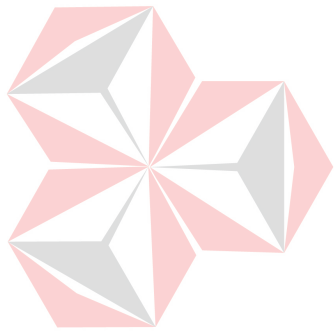
Web service merupakan kumpulan aplikasi logika yang menyediakan data dan *service* bagi aplikasi-aplikasi yang lain (Danny Ryan dan Tommy Ryan, 2002). Adapun aplikasi terdistribusi tersebut dapat diakses oleh aplikasi-aplikasi *client* tanpa memperhatikan sistem operasi maupun bahasa pemrograman. Sebelum adanya *web service* terdapat teknologi CORBA dan OMG yang menggunakan bahasa Java dan DCOM dari *Microsoft*. Kekurangan yang dimiliki oleh kedua teknologi ini adalah program yang akan dipakai untuk mengakses komponen tersebut harus dibuat dengan bahasa yang sama dengan bahasa yang dipakai untuk membuat komponen tersebut untuk CORBA dan untuk DCOM cuma bisa dipakai di *platform* *Microsoft*.

Layanan yang disediakan oleh komponen *web service* umumnya berupa operasi-operasi logika maupun operasi *query* yang dimanfaatkan oleh banyak

client. *Service* tersebut dapat dimanfaatkan secara langsung dan juga dapat dimanfaatkan oleh *web service* lain. Sebagai contoh *web service* yang menangani operasi perkalaian dapat dimanfaatkan secara langsung yaitu program *client* dapat langsung memanggil *web service* tersebut, begitu juga dengan *web service* pembagian/penjumlahan/pengurangan dan *web service* - *web service* tersebut dapat pula dimanfaatkan oleh *web service* yang lain misalnya *web service* kalkulator dimana program *client* memberikan inputan kepada *web service* kalkulator dan *web service* ini akan memanggil *web service* yang menangani operasi-operasi matematika yang sesuai dengan inputan yang diberikan oleh program *client*. Program *client* yang memanfaatkan layanan tersebut dapat dibuat menggunakan bahasa pemrograman yang berbeda selama mempunyai standard komunikasi yang sama dengan komponen *web service* tersebut.



UNIVERSITAS
Dinamika



UNIVERSITAS
Dinamika

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Untuk menyelesaikan permasalahan autentifikasi terintegrasi pada *domain controller* dan *application server* Labkom STIKOM dilakukan tahapan – tahapan pengembangan sistem yang meliputi analisis permasalahan, perancangan diagram alir yang menunjukkan alur jalan dari sistem, desain arsitektur yang menunjukkan hubungan antar elemen. Perancangan sistem autentifikasi terintegrasi pada *domain controller* dan *application server* Labkom STIKOM Surabaya terdiri dari perancangan *data flow diagram*, *entity relationship diagram* yang terdiri dari *conceptual data model*, dan *physical data model*. Dalam bab ini juga dilengkapi dengan struktur tabel dan *interface* pada autentifikasi terintegrasi pada *domain controller* dan *application server* Labkom STIKOM Surabaya.

3.1 Analisis Permasalahan

Laboratorium komputer yang berada pada lantai 6 STIKOM Surabaya digunakan mahasiswa untuk proses mengajar akademik terutama adalah praktikum. Praktikum di STIKOM merupakan mata kuliah (MK) dengan bobot sebesar 1 (satu) sks yang dimaksudkan untuk membekali mahasiswa mengaplikasikan ilmu dan pengetahuan yang diperoleh saat pembelajaran dikelas, melalui kegiatan praktikum ini mahasiswa dapat mengerti bagaimana cara mengaplikasikan konsep yang didapat dikelas dengan mempraktekkan pada saat di Labkom.

Praktikum berlangsung mulai minggu ke 4 (empat) sampai dengan minggu ke 11 (sebelas) perkuliahan, dan pada waktu minggu ke 13 (tiga belas)

akan diadakan ujian praktikum. Labkom telah menerapkan konsep *paper less* dalam modul praktikum maupun soal-soal praktikum. Dengan adanya konsep *paper less* ini maka praktikan dapat *download* langsung untuk mendapatkan modul praktikum maupun soal-soal praktikum. Model pembelajaran praktikum ini dibagi menjadi 3 (tiga) bagian yaitu tes awal, tugas praktikum, dan ujian praktikum. Tes awal merupakan 5 (lima) soal pilihan ganda yang berisi tentang materi yang akan digunakan pada saat praktikum tersebut berlangsung, sedangkan model tes awal adalah praktikan STIKOM diberikan 5 (lima) pertanyaan optional secara acak antara praktikan 1 (satu) dengan yang lainnya. Tugas praktikum adalah tugas evaluasi yang diberikan untuk praktikan menjelang akhir praktikum sesuai materi pada setiap pertemuan mingguan. Setelah selesai mengerjakan tugas praktikum, maka praktikan akan *upload* jawaban ke *server* Labkom. Untuk ujian praktikum praktikan akan diberikan sebuah soal individu yang harus dikerjakan oleh masing-masing praktikan STIKOM. Desain arsitektur pada Labkom adalah *client-server*, dimana *server* memiliki banyak *client* yang terhubung dengan menggunakan *domain* pada *server*. *Client* akan mengakses *application server* yang disebut PDC-Labkom. Materi maupun soal praktikum yang diberikan kepada praktikan berupa *softcopy* yang terdapat pada *server*, sehingga praktikan akan *download* materi maupun soal pada *server* Labkom. Satu praktikan dapat mengikuti beberapa praktikum, namun dengan satu *user* dan *password* yang sama. Untuk memulai praktikum di Labkom praktikan harus memasukkan *user* dan *password* pada awal *login domain*. Lalu praktikan dapat akses *application server* yang disebut PDC-Labkom. Pada PDC-Labkom praktikan dapat melakukan tes

awal, dan untuk soal praktikum para praktikan diharuskan *download* dari *server*. Dan yang terakhir para praktikan akan *upload* jawaban ke *server*.

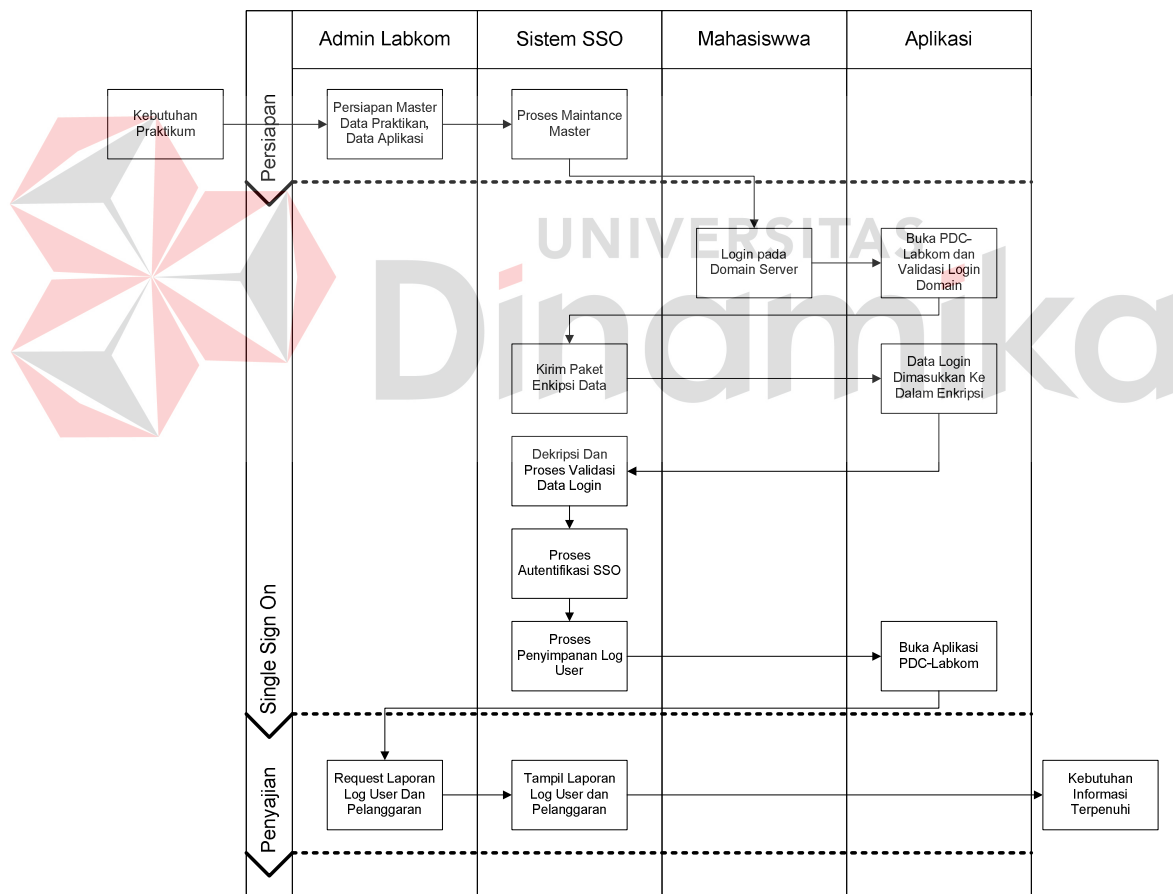
Selama proses praktikum, praktikan akan *log in application server* untuk dapat masuk ke dalam sistem praktikum. Namun praktikan dapat *log in* lebih dari satu *user* pada komputer dan sesi yang sama pada saat melakukan proses praktikum. Permasalahannya adalah 1 (satu) komputer pada sesi yang sama dapat multi *account* praktikan pada saat akses PDC-Labkom. Sehingga praktikan dapat membuka akses *login* PDC-Labkom yang bukan milik dirinya sendiri. Hal ini dapat mengakibatkan praktikan salah satunya dapat *download* jawaban milik temannya yang sudah di*upload*. Praktikan juga dapat membuka tes awal milik temannya, sehingga praktikan tersebut mengetahui soal tes awal lebih dulu dengan menggunakan *login* temannya. Saat ini Labkom belum dapat menangani masalah multi *account* tersebut, sehingga praktikan dapat melakukan kecurangan-kecurangan multi *account* pada saat praktikum. Dengan adanya indikasi praktikan dapat melakukan multi *account*, nilai yang didapat oleh praktikan bisa dikatakan ada yang tidak murni dari hasil kerja sendiri, dan dibutuhkan sistem yang dapat menangani masalah multi *account* tersebut.

Berdasarkan latar belakang di atas, dibuat *web service* teknologi SSO sebagai sarana autentifikasi terintegrasi pada *domain controller* dan *application server* pada sistem praktikum Labkom STIKOM Surabaya. SSO ini merupakan teknologi yang dapat menggunakan *user* dan *password* diambil dari *login domain*. Data *user* dan *password* tersebut dapat digunakan untuk mengakses PDC-Labkom sehingga tes awal, tugas praktikum serta ujian praktikum tanpa memasukkan kembali *user* dan *password*. Setiap praktikan hanya dapat *log in application*

server di satu komputer untuk sesi tertentu. Hal ini dapat meminimalisasikan adanya kecurangan multi *account* praktikan pada saat melakukan kegiatan praktikum. Sehingga metode SSO yang dapat menjadi sarana autentifikasi untuk meminimalisir adanya kecurangan multi *account* pada saat praktikum pada sesi tertentu.

3.2 Model Pengembangan

Berdasarkan analisis di bagian 3.1, berikut disajikan *Block Diagram* untuk menjelaskan alur proses yang terjadi dalam sistem secara umum.

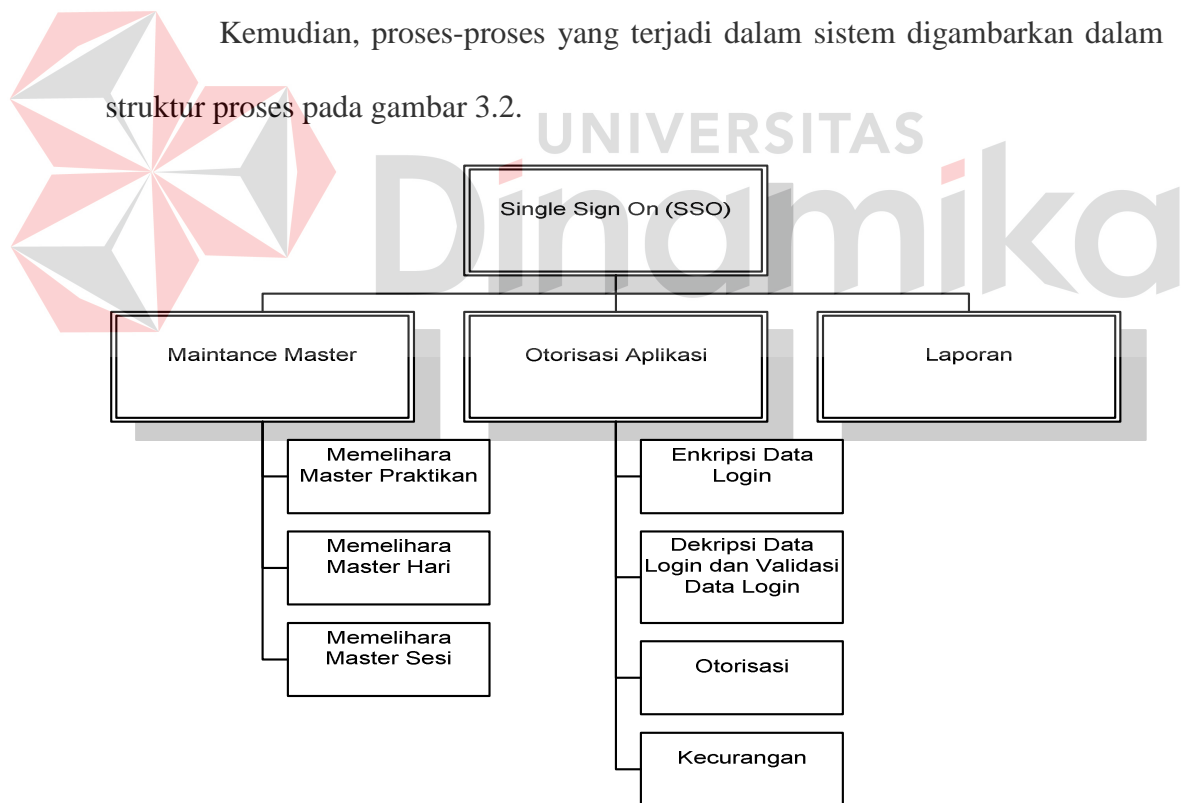


Gambar 3.1. *Block Diagram* Sistem SSO

Pada block diagram ini terbagi menjadi 3 proses utama yaitu tahap persiapan, *single sign on*, dan penyajian. Tahap persiapan adalah pemeliharaan

data *master* yang meliputi data *master* praktikan, hari dan sesi. Tahap *single sign on* adalah autentifikasi *request login*. Proses dimulai dari aplikasi *request login* pada sistem SSO. Sistem SSO mengirimkan paket enkripsi yang digunakan aplikasi untuk membungkus data *login* dan dikirimkan kembali ke sistem SSO. Kemudian paket tersebut dideskripsi untuk membuka paket enkripsi. Data *login* di validasi untuk mengetahui apakah praktikan dapat melakukan praktikum atau tidak. Setelah melalui proses validasi maka praktikan dapat melakukan proses praktikum selanjutnya. Tahap yang terakhir adalah penyajian, pada tahap ini *administrator* dapat melihat laporan histori praktikan maupun histori kecurangan praktikan.

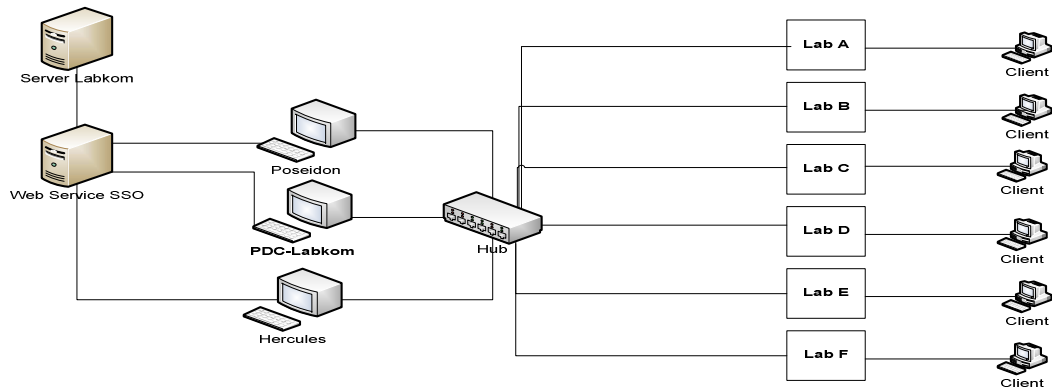
Kemudian, proses-proses yang terjadi dalam sistem digambarkan dalam struktur proses pada gambar 3.2.



Gambar 3.2. Struktur proses *Single Sign On*

Desain arsitektur jaringan SSO yang dirancang pada Labkom STIKOM

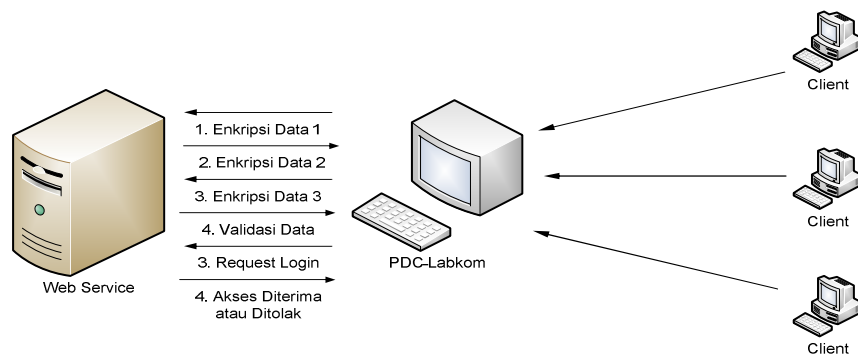
Surabaya dapat dilihat pada gambar 3.3.



Gambar 3.3. Desain arsitektur jaringan *Single Sign On*

Desain arsitektur jaringan terbagi menjadi 3 bagian utama yaitu *server*, *application server*, dan *client*. *Web service SSO* akan ditauh pada server sehingga dapat diakses oleh *client*. *Application server* terdiri dari PDC-Labkom, Poseidon, dan Hercules yang akan terhubung pada hub. *Client* terdiri dari beberapa lab yang terhungung pada hub, dan masing – masing lab tersebut mempunyai banyak *client* yang akan digunakan oleh praktikan

Desain arsitektur *background SSO* yang terjadi pada saat mengakses *application server* server dapat dilihat pada gambar 3.4.



Gambar 3.4. Desain arsitektur *background Single Sign On*

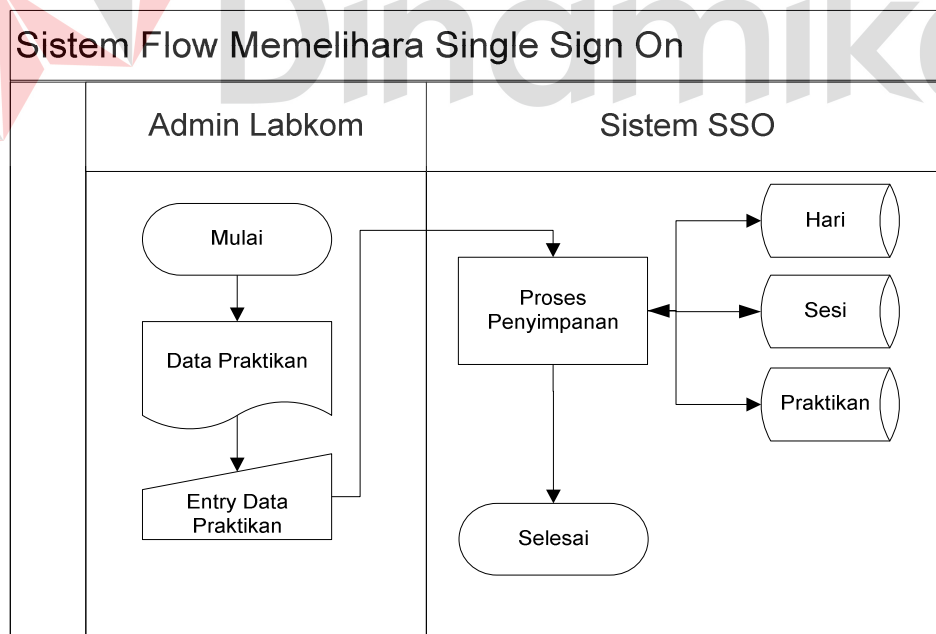
3.3 Perancangan Sistem

Dari analisis permasalahan yang dijelaskan pada 3.1, perancangan sistem yang dijelaskan secara berurutan sebagai berikut :

1. *System Flow*
2. *Data Flow Diagram (DFD)*
3. *Entity Relationship Diagram (ERD)*
4. *Struktur Database*
5. *Desain Interface*

3.3.1 System flow

System flow diawali oleh bagian admin Labkom melakukan *maintance* data *login* praktikan, hari dan sesi. Data *login* praktikan tersebut akan dibagikan pada setiap praktikan. Dapat dilihat pada Gambar 3.5.

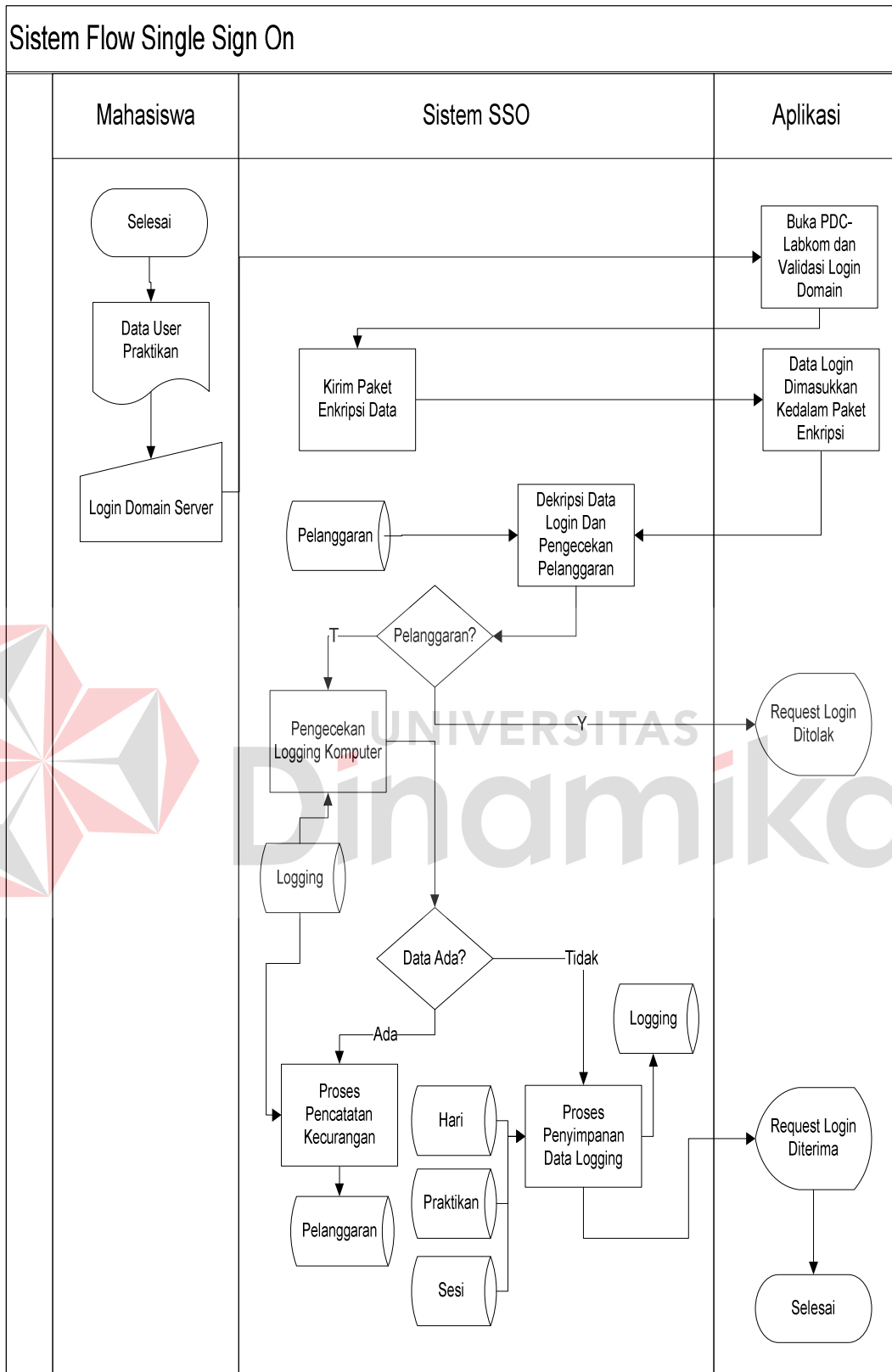


Gambar 3.5. *System Flow* Memelihara *Single Sign On*

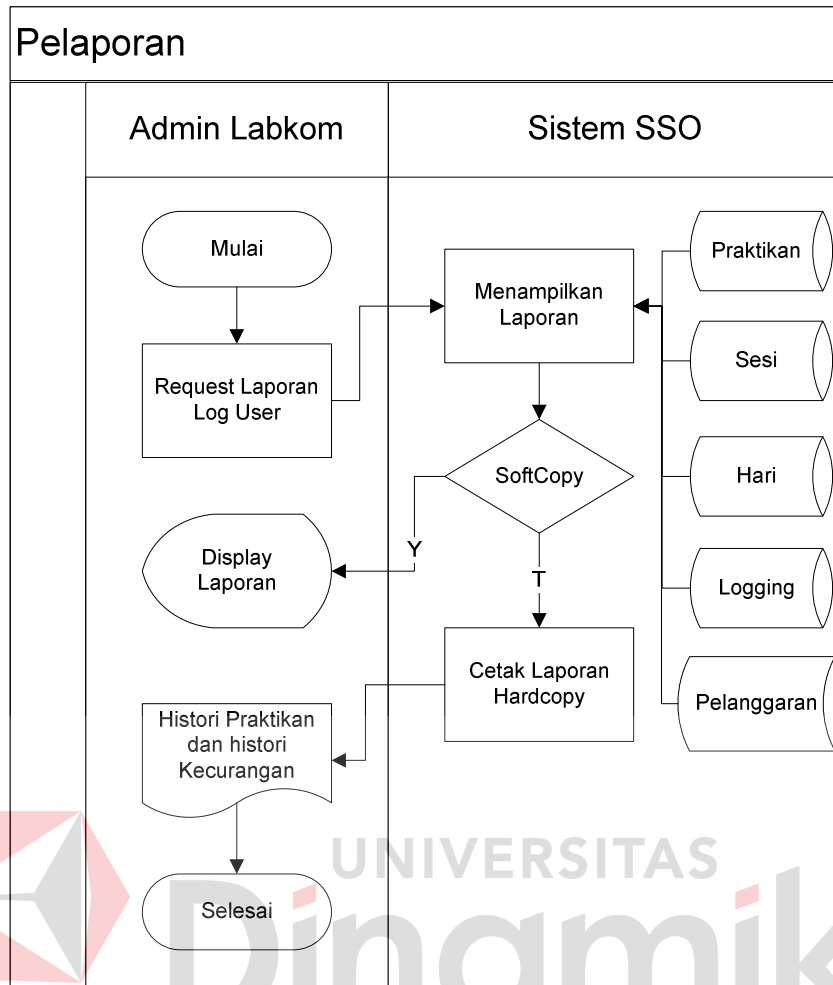
Pada saat proses praktikum berlangsung, praktikan memasukkan data *login* berupa *user* dan *password* pada saat *login domain server*. Kemudian

praktikan membuka sistem praktikum yang di sebut PDC-LABKOM. Pada saat PDC-LABKOM diakses, maka PDC-LABKOM akan meminta *request login* otomatis ke dalam *web-servis* dimana diambil dari *login domain server*. *Web-servis* akan menerima *request* tersebut dan akan mengirimkan enkripsi data *login* ke PDC-LABKOM. PDC-LABKOM akan memasukkan data *login domain server* praktikan ke dalam paket enkripsi data tersebut yang kemudian akan dikirim kembali ke *web-servis*. *Web-servis* akan membuka paket enkripsi tersebut dengan dekripsi untuk dapat membuka paket enkripsi tersebut. Setelah data *login* didapatkan, maka *web-servis* akan memeriksa data pelanggaran, apakah praktikan tersebut sebelumnya sudah melakukan pelanggaran. Jika benar maka *request login* tersebut akan ditolak, jika tidak maka *web-service* akan memeriksa data *logging* praktikan ada atau tidak di dalam *database* praktikan. Jika data ada maka praktikan akan menerima akses dapat masuk kedalam sistem PDC-LABKOM, yang kemudian data praktikan tersebut dimasukkan ke dalam *database logging*. Bentuk desain umum implementasi teknologi *Single Sign On* dapat dilihat pada gambar 3.6.

System flow pelaporan diawali oleh bagian admin Labkom *request* laporan yang diinginkan. Laporan terdiri dari 2 jenis : yaitu laporan histori peraktikan dan laporan histori kecurangan Sistem akan mengambil data ke dalam *database* sesuai dengan data yang dibutuhkan admin Labkom. Output laporan dapat berupa *softcopy* laporan yang akan ditampilkan dilayar ataupun *hardcopy* laporan yang berupa dokumen yang dapat dicetak. Bentuk desain umum implementasi teknologi *single sign on* dapat dilihat pada gambar 3.7.



Gambar 3.6. *System Flow Single Sign On*



Gambar 3.7. System Flow Pelaporan

3.3.2 Data flow diagram (DFD)

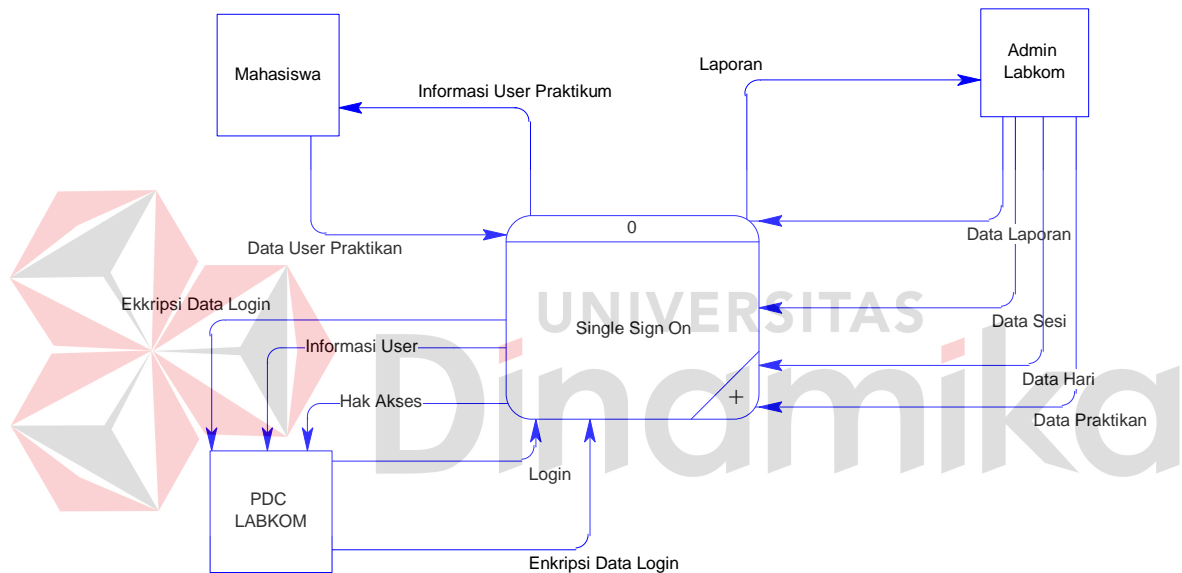
Data Flow Diagram adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

Data Flow Diagram merupakan suatu metode pengembangan sistem yang terstruktur (*structure analysis and design*). Penggunaan notasi dalam *data flow diagram* ini sangat membantu sekali untuk memahami suatu sistem pada

semua tingkat kompleksitas. Pada tahap analisis penggunaan notasi ini dapat membantu dalam berkomunikasi dengan pemakai sistem untuk memahami sistem secara logika.

A. context diagram

Diagram ini menggambarkan rancangan global/ keseluruhan dari proses yang ada pada DFD. Gambar 3.8 berikut ini merupakan tampilan dari context diagram sistem yang dirancang.



Gambar 3.8. *Context Diagram* dari DFD

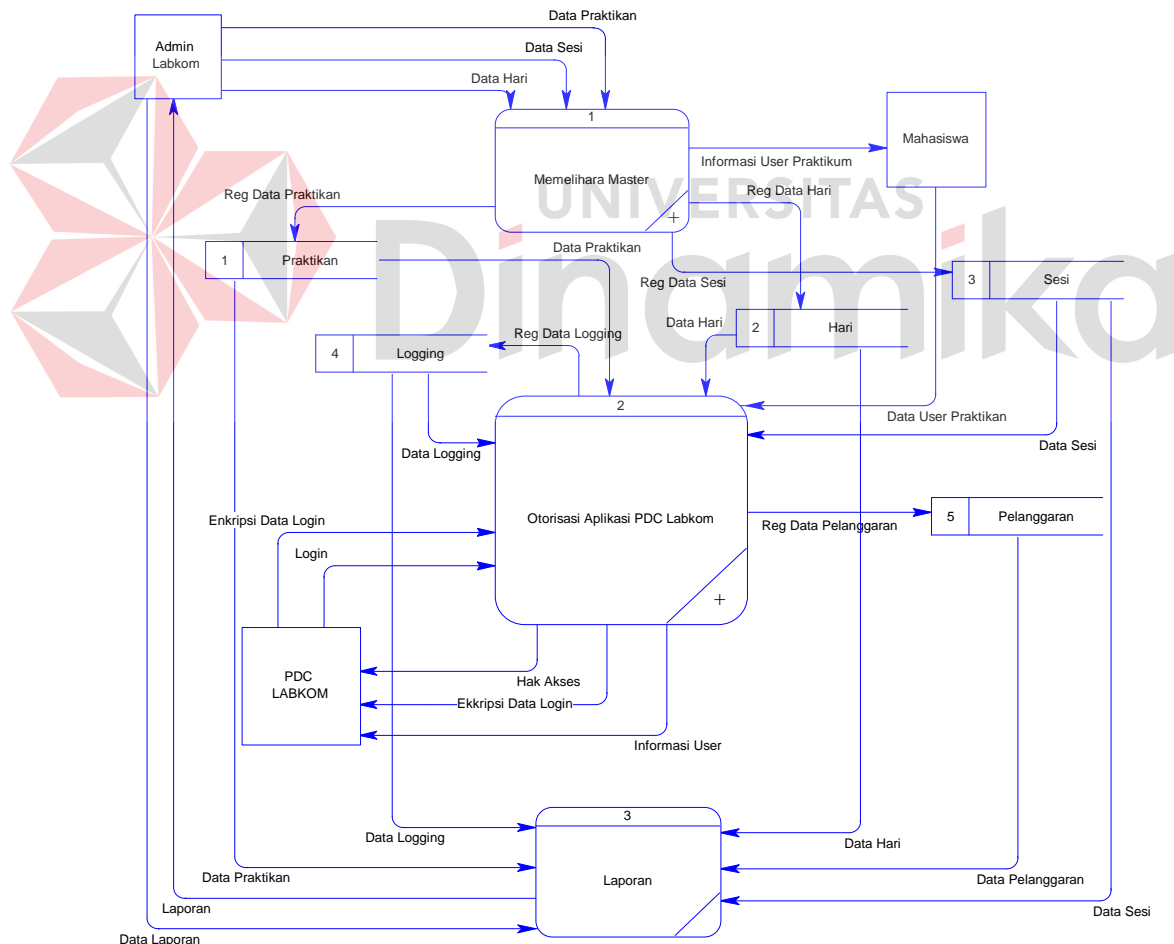
B. DFD level 0

Dari *context diagram single sign on*, sistem yang terjadi dapat dipecah lagi menjadi beberapa proses dan di dekomposisikan menjadi DFD level 0 yang terdiri dari 3 (empat) subproses, yaitu :

1. Memelihara *Master* praktikan, hari, dan sesi. Proses ini digunakan untuk memelihara data master, seperti memasukkan data master baru, mengedit data master, dan menghapus data master.

2. Otorisasi Aplikasi. Proses ini digunakan untuk proses *single sign on*, mulai dari pencatatan data *logging*, pencatatan data pelanggaran, dan penentuan apakah praktikan tersebut dapat hak akses untuk dapat melakukan proses praktikum
3. Laporan. Proses ini digunakan untuk menampilkan laporan ke admin Labkom. Laporan yang di *request* oleh admin Labkom akan di tampilkan sesuai kebutuhan yang diinginkan.

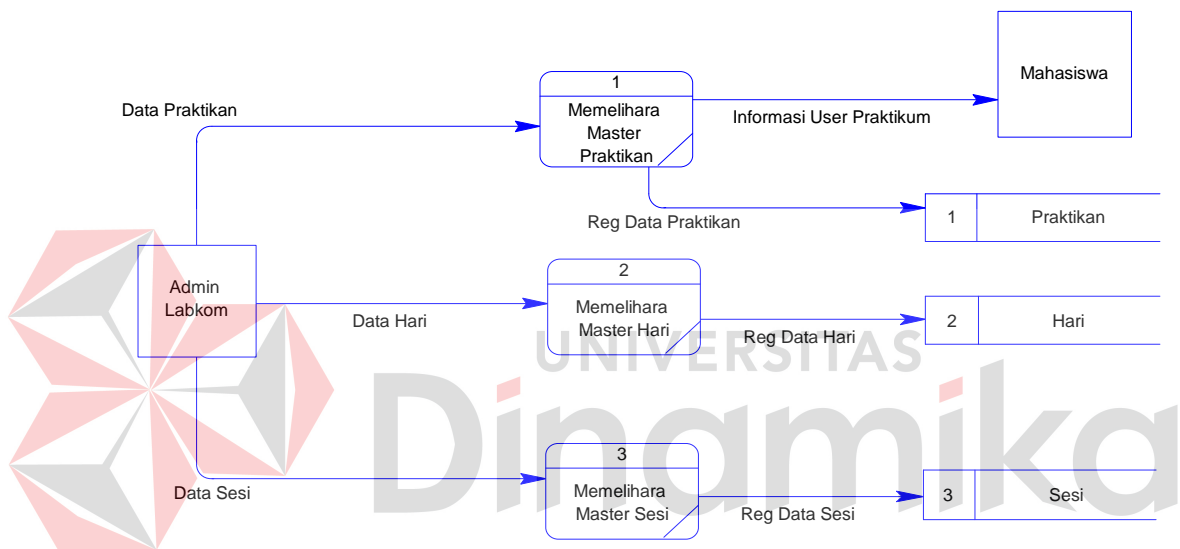
Penjelasan lebih lengkap mengenai DFD Level 0 *single sign on* dapat dilihat pada Gambar 3.9.



Gambar 3.9. DFD Level 0 Single Sign On

C. DFD level 1 memelihara master

Dari DFD Level 1 Memelihara *Master*, proses yang terjadi dapat dipecah lagi menjadi beberapa subproses, yaitu subproses Memelihara *Master* Praktikan yang digunakan untuk memelihara data praktikan, subproses Memelihara *Master* Hari yang digunakan untuk memelihara data hari, dan subproses Memelihara *Master* Sesi yang digunakan untuk memelihara sesi. Penjelasan lebih lengkap mengenai DFD Level 1 Memelihara *Master* dapat dilihat pada Gambar 3.10.

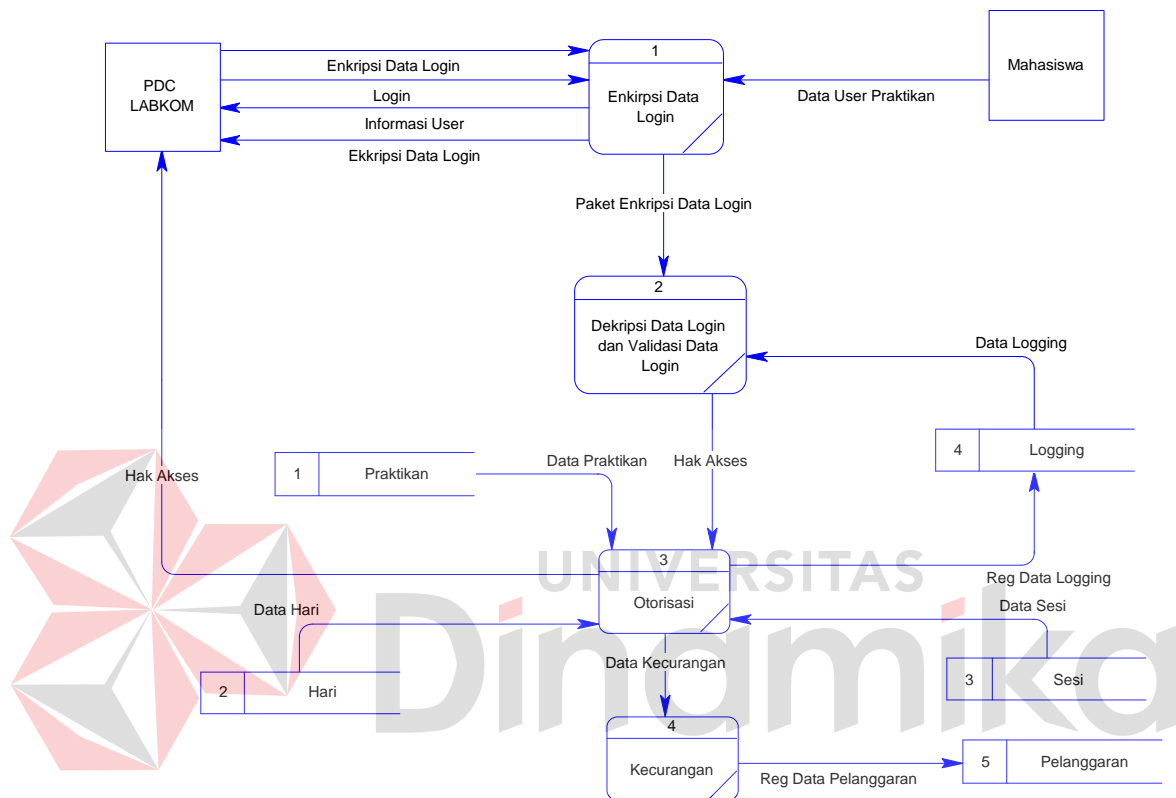


Gambar 3.10. DFD Level 1 Sub Proses Memelihara Data *Master*

D. DFD level 1 otorisasi aplikasi

Dari DFD Level 1 otorisasi aplikasi, proses yang terjadi dapat dipecah lagi menjadi beberapa subproses, yaitu subproses enkripsi data *login* yang digunakan untuk mengirim paket enkripsi data *login* jika ada *request* dari aplikasi untuk mendapatkan hak akses ke sistem, dekripsi data *login* dan validasi data *login* digunakan untuk membuka paket enkripsi data *login* dan memeriksa data *login* kedalam *database* praktikan, Otorisasi digunakan untuk proses pemberian hak akses masuk ke dalam sistem jika data *login* praktikan benar dan kemudian

dilakukan proses pencatatan data *logging* kedalam *database logging*, Kecurangan adalah pencatatan data kecurangan jika praktikan tersebut terdeteksi melakukan hal yang tidak dibenarkan seperti *dual login*. Penjelasan lebih lengkap mengenai DFD Level 1 Otorisasi Aplikasi dapat dilihat pada Gambar 3.11.



Gambar 3.11. DFD Level 1 Sub Proses Otorisasi Aplikasi

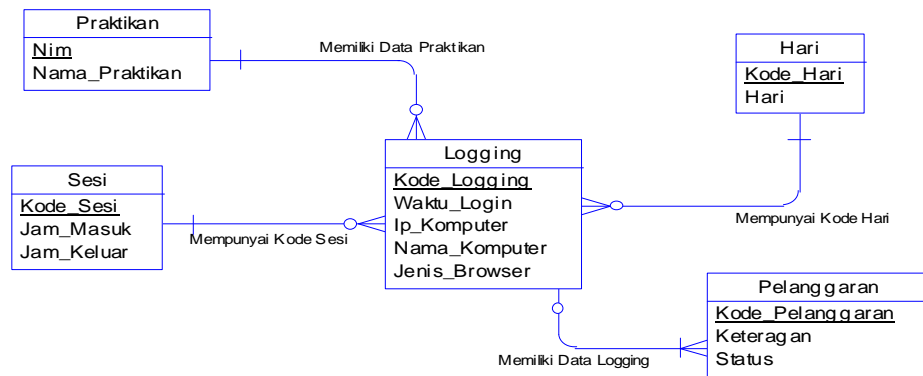
3.3.3 Entity relationship diagram (ERD)

Entity Relationship Diagram digunakan untuk menggambarkan, menentukan, dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan database. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari pemakai. Dalam ERD, data tersebut digambarkan dengan menggunakan simbol entitas.

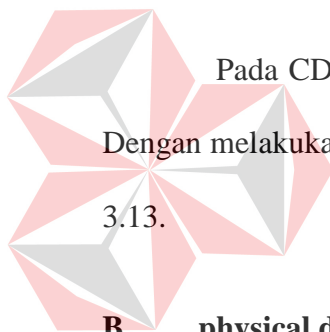
Dalam implementasi sistem *single sign on*, ada entitas yang saling terkait untuk menyediakan data yang dibutuhkan oleh sistem yang disajikan dalam

bentuk *conceptual data model* (CDM) dan *physical data model* (PDM). ERD dalam bentuk CDM dapat dilihat pada Gambar 3.12.

A. conceptual data model (CDM)

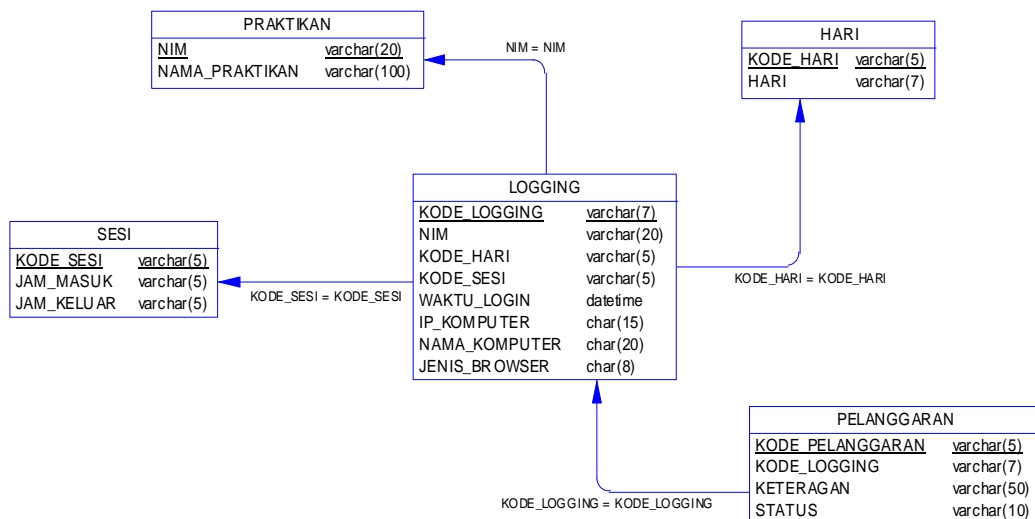


Gambar 3.12. *Conceptual Data Model* (CDM) dari ERD



Pada CDM implementasi sistem *single sign on*, memiliki 5 (lima) tabel. Dengan melakukan generate CDM, maka akan didapat PDM seperti pada Gambar 3.13.

B. physical data Model (PDM)



Gambar 3.13. *Physical Data Model* (PDM) dari ERD

PDM ini merupakan gambaran dari struktur database dari implementasi sistem *single sign on*.

3.3.4 Struktur database

Struktur *database* menggambarkan data-data yang ada dalam *database* beserta tipe dan kegunaannya.

1. NamaTabel : Praktikan
 - Primary Key : Nim
 - Foreign Key : -
 - Fungsi : Menyimpan data Master Praktikan

Tabel 3.1. Struktur Tabel Praktikan

Field	Tipe	Ukuran	Keterangan
Nim	Varchar	20	Kode praktikan sebagai identitas pratikan
Nama_Praktikan	Nvarchar	100	Nama praktikan

2. NamaTabel : Hari
 - Primary Key : Kode_Hari
 - Foreign Key : -
 - Fungsi : Menyimpan data Master Hari

Tabel 3.2. Struktur Tabel Hari

Field	Tipe	Ukuran	Keterangan
Kode_Hari	Varchar	5	Kode hari sebagai identitas hari
Hari	Nvarchar	7	Nama hari

3. NamaTabel : Sesi
 - Primary Key : Kode_Sesi

Foreign Key : -

Fungsi : Menyimpan data Master Sesi

Tabel 3.3. Struktur Tabel Sesi

Field	Tipe	Ukuran	Keterangan
Kode_Sesi	Varchar	5	Kode sesi sebagai identitas sesi
Jam_Masuk	Nvarchar	5	Keterangan jam mulai masuk praktikum
Jam_Keluar	Nvarchar	5	Keterangan jam selesai praktikum

4. NamaTabel : Pelanggaran

Primary Key : Kode_Pelanggaran

Foreign Key : Kode_Logging

Fungsi : Menyimpan data Pelanggaran

Tabel 3.4. Struktur Tabel Pelanggaran

Field	Tipe	Ukuran	Keterangan
Kode_Pelanggaran	Varchar	5	Kode pelanggaran sebagai identitas pelanggaran
Kode_Logging	Varchar	7	Kode <i>logging</i> sebagai identitas <i>logging</i>
Keterangan	Nvarchar	50	Digunakan untuk memberikan keterangan pelanggaran
Status	Nvarchar	10	Status digunakan untuk menandai praktikan yang sudah konfirmasi pelanggaran

5. NamaTabel : Logging

Primary Key : Kode_Logging

Foreign Key : Nim, Kode_Hari, Kode_Sesi, Kode_Aplikasi

Fungsi : Menyimpan data *Logging*

Tabel 3.5. Struktur Tabel Logging

Field	Tipe	Ukuran	Keterangan
Kode_Logging	Varchar	7	Kode <i>logging</i> sebagai identitas <i>logging</i>
Nim	Varchar	20	Kode praktikan sebagai identitas pratikan
Kode_Hari	Varchar	5	Kode hari sebagai identitas hari
Kode_Sesi	Varchar	5	Kode sesi sebagai identitas sesi
Waktu_Login	DateTime		Waktu <i>login</i> menggunakan aplikasi
Ip_Komputer	Nchar	15	Pencatatan ip komputer
Nama_Komputer	Nchar	20	Pencatatan nama komputer
Jenis_Browser	Nchar	8	Pencatatan <i>browser</i> yang dipakai

3.3.5 Desain interface

Desain *interface* dibuat sebelum membuat *interface* yang sesungguhnya.

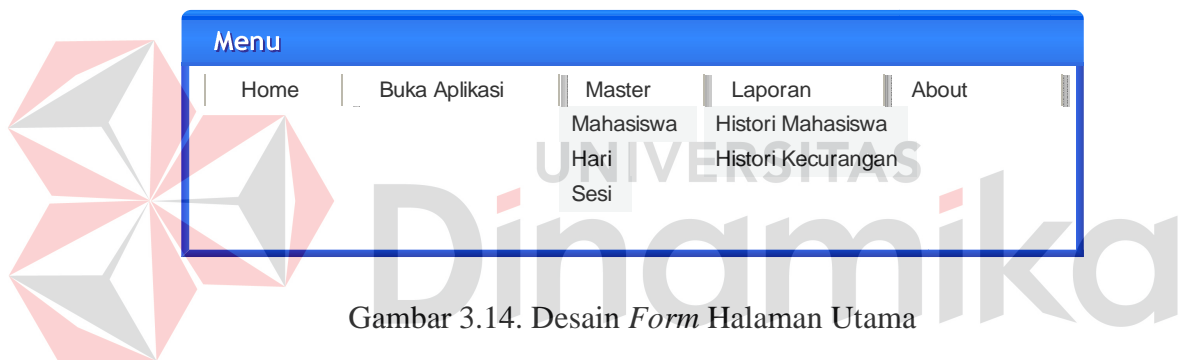
Desain ini dapat digunakan sebagai pembuatan *interface* program yang sesuai dengan kebutuhan *user*. Apabila desain ini sudah cukup *user friendly* dengan *user* maka selanjutnya dapat dibuat programnya sehingga apabila program digunakan oleh *user*, *user* akan menemukan kemudahan dalam menggunakan program ini. Namun apabila desain yang dibuat kurang diminati oleh *user* maka desain dapat diubah sebelum bertindak pada pembuatan program. Dalam aplikasi ini terdapat beberapa desain *interface*:

A. desain interface

Desain *interface* merupakan perancangan tampilan monitor masukan dari pengguna kepada sistem yang kemudian akan disimpan dalam *database*.

A.1 form halaman utama

Form yang akan muncul setelah *user* berhasil *login* adalah halaman utama. *Form* ini terdiri dari menu Home, Buka Aplikasi, Master Mahasiswa, Laporan, dan About yang digunakan untuk pengolahan data-data lebih lanjut untuk menghasilkan informasi. Gambar form utama dapat dilihat pada Gambar 3.14.



Gambar 3.14. Desain *Form* Halaman Utama

Fungsi-fungsi obyek dalam desain form utama adalah sebagai berikut :

Tabel 3.6. Fungsi-Fungsi Obyek Desain *Form* Halaman Utama

Nama Obyek	Tipe Obyek	Fungsi
Menu	<i>Link</i>	Digunakan memilih menu yang sesuai.

A.2 form buka aplikasi

Form akan muncul setelah *user* berhasil *login* dan *login* sebagai admin. *Form* ini berguna untuk membuka kunci, jika praktikan tersebut tercatat melakukan pelanggaran atau kecurangan pada saat praktikum. Gambar *form* buka aplikasi dapat dilihat pada Gambar 3.15.

Gambar 3.15. Desain *Form* Buka Aplikasi

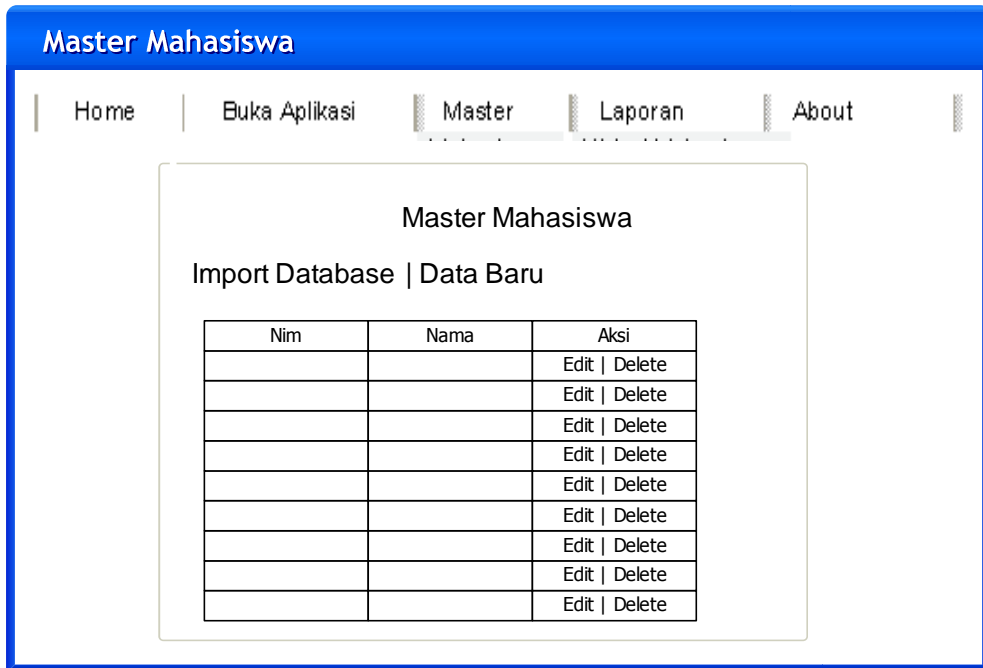
Fungsi-fungsi obyek dalam desain *form* buka aplikasi adalah sebagai berikut :

Tabel 3.7. Fungsi-Fungsi Obyek Desain Buka Aplikasi

Nama Obyek	Tipe Obyek	Fungsi
<i>Field</i>	<i>Textbox</i>	Digunakan menginputkannim.
Ok	<i>Button</i>	Digunakan untuk membuka akses
<i>Cancel</i>	<i>Button</i>	Digunakan untuk mengosongkan <i>field</i>

A.4 form master mahasiswa

Form akan muncul setelah *user* berhasil *login* dan *login* sebagai admin. Pada *form* ini berguna untuk memelihara data *master* mahasiswa. Gambar *form master* mahasiswa dapat dilihat pada Gambar 3.16. Jika ingin menambahkan data mahasiswa baru maka dapat dilihat pada Gambar 3.17. Jika ingin merubah data mahasiswa yang telah dipilih maka dapat dilihat pada Gambar 3.17.

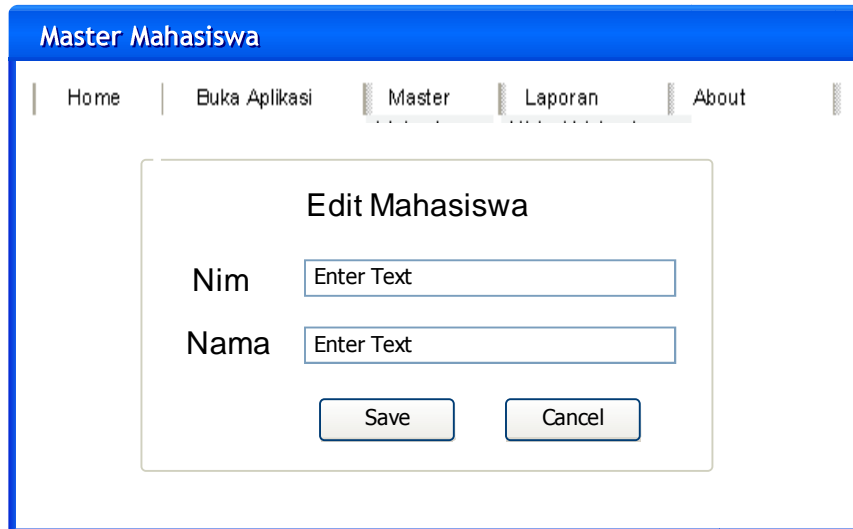


Gambar 3.16. Desain *Form Master* Mahasiswa

Fungsi-fungsi obyek dalam desain *form master* mahasiswa adalah sebagai berikut :

Tabel 3.8. Fungsi-Fungsi Obyek Desain *Master* Mahasiswa

Nama Obyek	Tipe Obyek	Fungsi
<i>Import Database</i>	<i>Label</i>	Digunakan untuk memasukkan data praktikan dari database luar.
Data Baru	<i>Link</i>	Membuka menu data baru praktikan dengan menampilkan menu data baru
<i>Edit</i>	<i>Link</i>	Membuka menu edit praktikan dengan menampilkan menu edit
<i>Delete</i>	<i>Link</i>	Digunakan untuk menghapus data yang akan dihapus



Gambar 3.17. Desain *Form Edit Mahasiswa*

Fungsi-fungsi obyek dalam desain *form* edit mahasiswa adalah sebagai

berikut :

Tabel 3.9. Fungsi-Fungsi Obyek Desain *Edit Mahasiswa*

Nama Obyek	Tipe Obyek	Fungsi
<i>Field</i>	<i>Textbox</i>	Digunakan menginputkan data praktikan.
<i>Save</i>	<i>Button</i>	Digunakan untuk menyimpan data
<i>Cancel</i>	<i>Button</i>	Digunakan untuk mengosongkan field

A.5 form master hari

Form akan muncul setelah *user* berhasil *login* dan *login* sebagai admin.

Pada *form* ini berguna untuk memelihara data *master* hari. Gambar *form master*

hari dapat dilihat pada Gambar 3.18. Jika ingin merubah data hari yang telah

dipilih maka dapat dilihat pada Gambar 3.19.



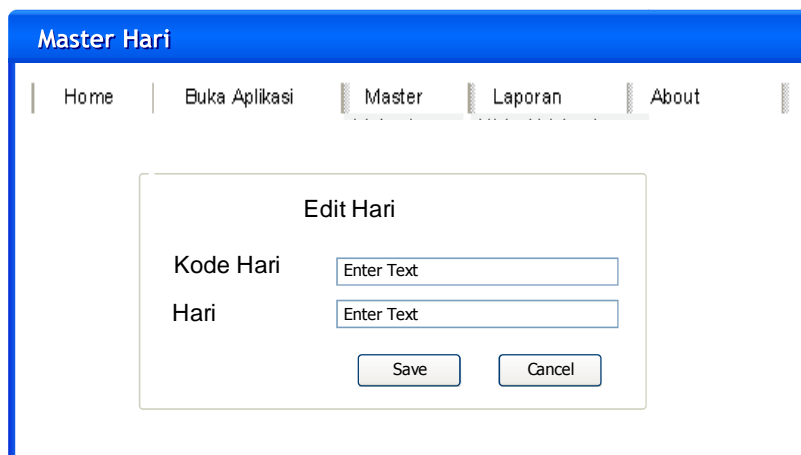
Gambar 3.18. Desain *Form Master Hari*

Fungsi-fungsi obyek dalam desain *form master* hari adalah sebagai

berikut:

Tabel 3.10. Fungsi-Fungsi Obyek Desain *Master Hari*

Nama Obyek	Tipe Obyek	Fungsi
<i>Edit</i>	<i>Link</i>	Membuka menu edit hari dengan menampilkan menu edit
<i>Delete</i>	<i>Link</i>	Digunakan untuk menghapus data yang akan dihapus



Gambar 3.19. Desain *Form Edit Hari*

Fungsi-fungsi obyek dalam desain *form* edit hari terdapat pada tabel 3.12.

Tabel 3.11. Fungsi-Fungsi Obyek Desain *Edit* Mahasiswa

Nama Obyek	Tipe Obyek	Fungsi
<i>Field</i>	<i>Textbox</i>	Digunakan menginputkan data hari.
<i>Save</i>	<i>Button</i>	Digunakan untuk menyimpan data
<i>Cancel</i>	<i>Button</i>	Digunakan untuk mengosongkan <i>field</i>

A.6 form master sesi

Form akan muncul setelah *user* berhasil *login* dan *login* sebagai admin.

Pada *form* ini berguna untuk memelihara data *master* sesi. Gambar *form master* sesi dapat dilihat pada Gambar 3.20. Jika ingin merubah data sesi yang telah dipilih maka dapat dilihat pada Gambar 3.21.



Gambar 3.20. Desain *Form Master Sesi*

Fungsi-fungsi obyek dalam desain *form master* Sesi dapat dilihat pada

Tabel 3.13:

Tabel 3.12. Fungsi-Fungsi Obyek Desain *Master* sesi

Nama Obyek	Tipe Obyek	Fungsi
<i>Edit</i>	<i>Link</i>	Membuka menu edit sesi dengan menampilkan menu edit
<i>Delete</i>	<i>Link</i>	Digunakan untuk menghapus data

Gambar 3.21. Desain *Form* Edit Sesi

Fungsi-fungsi obyek dalam desain *form* edit Sesi adalah sebagai berikut :

Tabel 3.13. Fungsi-Fungsi Obyek Desain *Edit* sesi

Nama Obyek	Tipe Obyek	Fungsi
<i>Field</i>	<i>Textbox</i>	Digunakan menginputkandata sesi.
<i>Save</i>	<i>Button</i>	Digunakanuntukmenyimpan data
<i>Cancel</i>	<i>Button</i>	Digunakanuntukmengosongkan <i>field</i>

A.7 rekap data histori praktikum

Data ditampilkan berdasarkan data *logging* yang masuk ke dalam sistem. Kemudian berdasarkan fasilitas *filtering*, data dapat ditampilkan sesuai kebutuhan. Jika ingin menampilkan rekap data histori praktikum, maka dipilih tanggal mulai sampai akhir yang ingin ditampilkan. Lalu tombol *pross* digunakan untuk menampilkan data yang sudah dipilih. Laporan histori praktikum ini berupa

softcopy , dan jika diinginkan cetak *hardcopy* maka tombol cetak digunakan untuk mencetak laporan histori praktikan.

REKAP HISTORI PRAKTIKUM									
Tanggal Mulai			<input type="text"/>	▼	Tanggal Akhir			<input type="text"/>	▼
<u>Proses</u>					<u>Cetak</u>				
KodeLogging	NIM	Hari	Sesi	Aplikasi	Waktu Login	Ip Komputer	Nama Komputer	Jenis Browser	

Gambar 3.22. Contoh Tampilan Rekap Data Histori Praktikum

A.8 rekap data histori pelanggaran

Data berdasarkan data pelanggaran yang sudah masuk ke dalam sistem. Jika ingin menampilkan rekap data histori pelanggaran, maka dipilih tanggal mulai sampai akhir yang ingin ditampilkan. Lalu tombol proses digunakan untuk menampilkan data yang sudah dipilih. Laporan histori pelanggaran ini berupa *softcopy* , dan jika diinginkan cetak *hardcopy* maka tombol cetak digunakan untuk mencetak laporan histori pelanggaran. Tombol cetak digunakan untuk mencetak.

REKAP HISTORI PELANGGARAN									
Tanggal Mulai			<input type="text"/>	▼	Tanggal Akhir			<input type="text"/>	▼
<u>Proses</u>					<u>Cetak</u>				
Kode Pelanggaran	Kode Logging	Nim	Nama	Hari	Sesi	Aplikasi	Nama Komputer	Keterangan	

Gambar 3.23. Contoh Tampilan Rekap Data Histori Pelanggaran

3.3.6 Desain uji coba

Desain uji coba bertujuan untuk memastikan bahwa aplikasi telah dibuat sesuai dengan kebutuhan atau tujuan yang diharapkan. Kekurangan atau kelemahan aplikasi pada tahap ini akan dievaluasi sebelum di implementasikan secara nyata. Desain uji coba dasar sistem ini dilakukan dengan menggunakan *Black Box Testing* dimana aplikasi akan diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang telah dibuat telah sesuai dengan tujuan.

A. desain uji coba fitur buka aplikasi

Form buka aplikasi digunakan untuk membuka hak akses bagi praktikan yang tercatat telah melakukan pelanggaran *dual login*. Contoh data praktikan yang telah melakukan kecurangan yang digunakan terlihat pada Tabel 3.14. Sedangkan penjelasan desain *test case* buka aplikasi dapat terlihat pada Tabel 3.15.

Tabel 3.14. Data *Logging*

Nama Field	Data 1	Data 2
Nim	07410100195	070195

Tabel 3.15. *Test Case Data Logging*

Test Case ID	Tujuan	Input	Output Diharapkan
1	Deskripsi nim yang valid	Memasukkan data 1 (satu) seperti pada tabel 3.14.	Pesan "Nim telah di berikan hak akses dan dapat melakukan praktikum"
2	Deskripsi nim yang tidak valid	Memasukkan data 2 (dua) seperti pada tabel 3.14.	pesan "Data praktikan tidak melakukan pelanggaran"

B. desain uji coba fitur edit praktikan

Form edit praktikan digunakan untuk memelihara data praktikan. Contoh data praktikan terlihat pada Tabel 3.16. Sedangkan penjelasan desain *test case* Edit Praktikan dapat terlihat pada Tabel 3.7.

Tabel 3.16. Data Praktikan

Nama Field	Data 1	Data 2
Nim	07410100195	070195
Nama	Diki Anggoro Putra	Diki Anggoro Putra

Tabel 3.17. *Test Case* Data Praktikan

Test Case ID	Tujuan	Input	Output Diharapkan
1	Deskripsi data valid	Memasukkan data 1 (satu) seperti pada tabel 3.16.	Muncul pesan "Proses Simpan Data Praktikan berhasil"
2	Deskripsi data tidak valid	Memasukkan data 2 (dua) seperti pada tabel 3.16.	Muncul pesan "Proses Penyimpanan Data tidak berhasil"

C. desain uji coba fitur edit hari

Form edit praktikan digunakan untuk memelihara data hari. Contoh data hari terlihat pada Tabel 3.18. Sedangkan penjelasan desain *test case* Edit hari dapat terlihat pada Tabel 3.19.

Tabel 3.18. Data Hari

Nama Field	Data 1	Data 2
Kode Hari	hari01	ha01
Hari	Senin	Senin

Tabel 3.19. *Test Case* Data Hari

Test Case ID	Tujuan	Input	Output Diharapkan
1	Deskripsi data valid	Memasukkan data 1 (satu) seperti pada tabel 3.18.	Muncul pesan "Proses Penyimpanan Data Hari berhasil"
2	Deskripsi data tidak valid	Memasukkan data 2 (dua) seperti pada tabel 3.18.	Muncul pesan "Proses Penyimpanan Data Hari tidak berhasil"

D. desain uji coba fitur edit sesi

Form edit praktikan digunakan untuk memelihara data sesi. Contoh data sesi terlihat pada Tabel 3.20. Sedangkan penjelasan desain *test case* Edit sesi dapat terlihat pada Tabel 3.21.

Tabel 3.20. Data Sesi

Nama Field	Data 1	Data 2
Kode Sesi	sesi01	se01
Jam Masuk	07.30	09.10
Jam Keluar	07.30	09.10

Tabel 3.21. *Test Case* Data Sesi

Test Case ID	Tujuan	Input	Output Diharapkan
1	Deskripsi data valid	Memasukkan data 1 (satu) seperti pada tabel 3.23.	Muncul pesan "Proses Penyimpanan Data Sesi berhasil"
2	Deskripsi data tidak valid	Memasukkan data 2 (dua) seperti pada tabel 3.23.	Muncul pesan "Proses Penyimpanan Data Sesi tidak berhasil"

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Implementasi Sistem

Tahap ini merupakan pembuatan perangkat lunak yang disesuaikan dengan rancangan atau desain sistem yang telah dibuat. Aplikasi yang dibuat akan diterapkan berdasarkan kebutuhan. Selain itu aplikasi ini akan dibuat sedemikian rupa sehingga dapat memudahkan pengguna untuk menggunakan aplikasi implementasi sistem autentifikasi terintegrasi pada *domain controller* dan *application server* labkom STIKOM.

Sebelum menjalankan aplikasi ini, ada hal yang harus diperhatikan yaitu kebutuhan sistem. Sesuai dengan kebutuhan untuk merancang implementasi teknologi *single sign on* diperlukan perangkat keras dan perangkat lunak.

implementasi sistem autentifikasi terintegrasi pada *domain controller* dan *application server* labkom STIKOM, diperlukan dukungan *software* dan *hardware* sebagai berikut :

4.1.1 Kebutuhan perangkat keras

Perangkat keras adalah komponen fisik peralatan yang membentuk sistem komputer, serta peralatan lain yang mendukung komputer dalam menjalankan tugasnya.

A. kebutuhan minimum client

Untuk menjalankan aplikasi ini sebagai *client* membutuhkan komputer dengan spesifikasi minimum sebagai berikut:

1. *Processor*1Ghz
2. *Memory* dengan RAM 512 MB
3. VGA on Board
4. Monitor Super VGA (800x600) dengan minimum 256 warna
5. Keyboard + mouse

B. kebutuhan minimum server

Untuk menjalankan aplikasi ini sebagai *server* membutuhkan komputer dengan spesifikasi minimum sebagai berikut:

1. *Processor*1Ghz
2. *Memory* dengan RAM 1 GB
3. VGA on Board
4. Monitor Super VGA (800x600) dengan minimum 256 warna
5. Keyboard + mouse

4.1.2 Kebutuhan perangkat lunak

Perangkat lunak adalah komponen non fisik yang digunakan untuk membuat sistem komputer dapat berjalan dan melakukan tugasnya.

A. kebutuhan minimum client

Adapun perangkat lunak yang dibutuhkan dan telah diujicobakan pada komputer *client* yaitu:

1. *Operating System* : WindowsXP Service Pack 2
2. *Browser* : Internet Explorer versi 6.0

B. kebutuhan minimum server

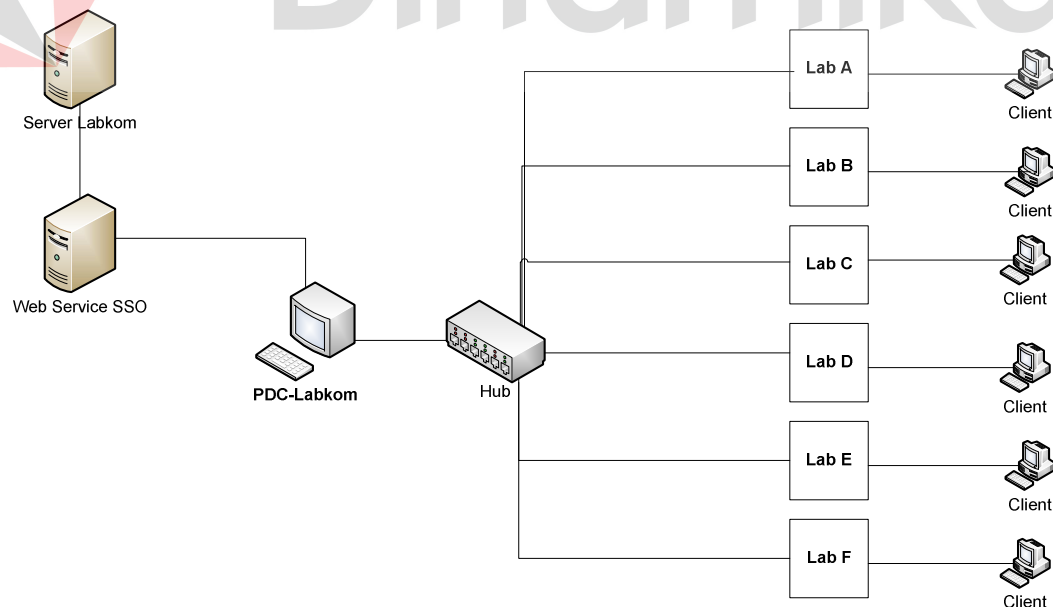
Adapun perangkat lunak yang dibutuhkan dan telah diuji cobakan pada komputer *server* yaitu:

1. *Operating System* : Windows XP Service Pack 2
2. *Web Server*: IIS 5.0
3. *Programming Language* : Visual Studio 2005
4. *Database*: SQL Server 2005 Express

4.2 Pembuatan dan Implementasi Program

Aplikasi ini dibuat menggunakan Microsoft ASP.Net 2.0 dengan *database engine* Microsoft SQL Server 2005 Express. *Source code* atau listing program dari aplikasi yang dibuat terdapat pada lampiran 5.

Tahap akhir implementasi program adalah integrasi *server* Laboratorium STIKOM Surabaya, terdapat pada Gambar 4.1



Gambar 4.1. Implementasi Jaringan SSO pada Labkom STIKOM Surabaya

Desain arsitektur *web servis* SSO akan terintegrasi dengan *server* yang berada di Labkom. Kemudian *web servis* SSO di akses oleh *application server* seperti PDC-Labkom yang digunakan untuk praktikum. Namun pada tugas akhir kali ini batasan masalahnya hanya pada *application server* PDC-Labkom. Selurur *application server* akan terhubung pada hub dimana hub akan dihubungkan pada masing – masing laboratorium komputer. Laboratorium komputer akan memiliki banyak *client* yang dapat mengakses *application server* tersebut. Sehingga jika *client* mengakses PDC-Labkom maka *web servis* akan melakukan validasi *login* aplikasi yang diambil dari *login domain*.

4.3 Evaluasi Sistem

Tahapan evaluasi implementasi teknologi *single sign on* sebagai sarana autentifikasi pada sistem praktikum terbagi menjadi dua yaitu Evaluasi hasil uji coba sistem dan Analisa hasil uji coba sistem. Evaluasi hasil uji coba dilakukan untuk menguji kembali semua tahapan yang sudah dilakukan selama pengujian berlangsung dan analisa hasil uji coba sistem bertujuan untuk menarik kesimpulan terhadap hasil-hasil uji coba yang dilakukan terhadap sistem. Uji coba dilakukan dalam tahapan beberapa *test case* yang telah disiapkan sebelumnya.

4.3.1 Evaluasi hasil uji coba sistem

Untuk memastikan bahwa sistem telah dibuat sesuai dengan kebutuhan atau tujuan yang diharapkan maka dilakukan beberapa uji coba. Uji coba meliputi pengujian terhadap fitur dasar aplikasi, uji coba perhitungan dan uji coba validasi pengguna terhadap aplikasi dengan menggunakan *black box testing*.

A. home page

Inilah halaman yang pertama kali akan ditampilkan ketika admin membuka situs *single sign on*. Status pertama kali akan otomatis muncul sesuai dengan *user* waktu *login* Windows.



Gambar 4.2. Home Page (Admin)

B. menu utama

Ada 2 tingkatan hak akses *user* dalam program ini, yaitu sebagai :

1. Administrator
2. Praktikan

Menu utama yang akan muncul dan halaman yang bisa diakses oleh *user* tergantung kepada tingkatan hak akses *user* tersebut.

a. administrator

1. Menu Utama

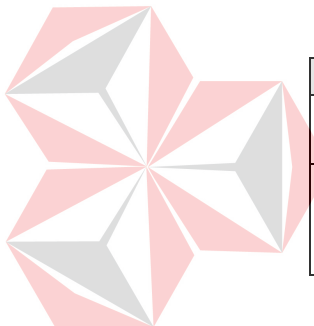
Dalam menu utama Admin terdapat menu Home, Master, Laporan, dan About seperti terlihat di Gambar 4.3.

Gambar 4.3. Menu Utama Administrator

2. Menu Buka Aplikasi

Dalam menu buka aplikasi ini berfungsi sebagai pembuka kunci jika ada mahasiswa yang melakukan kecurangan pada saat praktikum. Setelah dibuka kunci, selanjutnya praktikan dapat mengakses kembali aplikasi Labkom. Namun dengan catatan nim praktikan tersebut sudah tercatat pada tabel kecurangan. Data yang digunakan terlihat pada Tabel 4.1 dan *test case* data buka aplikasi dapat dilihat pada Tabel 4.2.

Tabel 4.1. Data Nim



Nama Field	Data-1	Data-2
Nim	07410100190	07410100195
Status	Terdaftar curang	Tidak terdaftar curang

Tabel 4.2. Test Case Data Nim

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
1	Deskripsi nim yang valid	Memasukkan data 1 (satu) seperti pada tabel 4.1	Akan muncul pesan pembukaan nim berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Muncul pesan 3. Data dapat tersimpan
2	Deskripsi nim yang tidak valid	Memasukkan data 2 (satu) seperti pada tabel 4.1	Akan muncul pesan pembukaan nim tidak berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Muncul pesan

Menu buka aplikasi dapat dilihat pada Gambar 4.4. Berdasarkan uji coba No. 1 (satu) pada Tabel 4.2 ditunjukkan pada Gambar 4.5. Gambar 4.6 menjelaskan hasil uji coba No. 2 (dua) pada Tabel 4.2.



Gambar 4.4. Menu Buka Aplikasi (*Admin*)



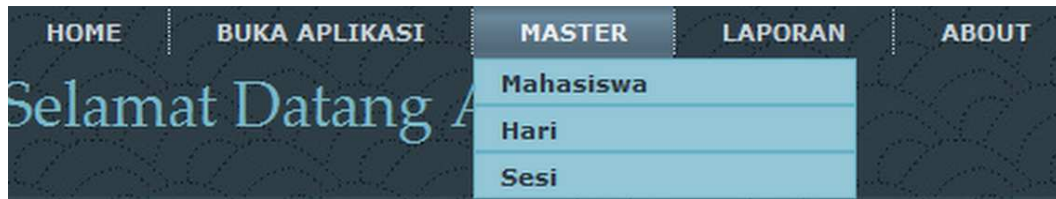
Gambar 4.5. Pesan Data Nim Telah Diberikan Akses (Data Telah Tersimpan)



Gambar 4.6. Pesan Data Nim Tidak Ada Atau Tidak Melakukan Pelanggaran

3. Menu Master

Dalam Menu *Master* terdapat Submahasiswa, hari dan sesi seperti terlihat pada Gamabr 4.7.



Gambar 4.7. Menu *master* (Admin)

a. Master Mahasiswa

Dalam *master* mahasiswa ini berfungsi sebagai mengelolah data praktikan. Data praktikan yang tercatat dalam *database* ini akan dapat mengakses aplikasi praktikum. Dapat dilihat pada Gambar 4.8. Klik import *database* berguna untuk memindahkan data praktikan dari *database* luar kedalam *database* SSO.

Klik data baru dan edit akan menampilkan halaman edit data praktikan, seperti pada Gambar 4.9. Klik delete berguna untuk menghapus data praktikan yang dapat mengakses aplikasi Labkom. Data yang digunakan terlihat pada Tabel 4.3 dan *test case* data praktikan dapat dilihat pada Tabel 4.4.

Tabel 4.3. Data Praktikan

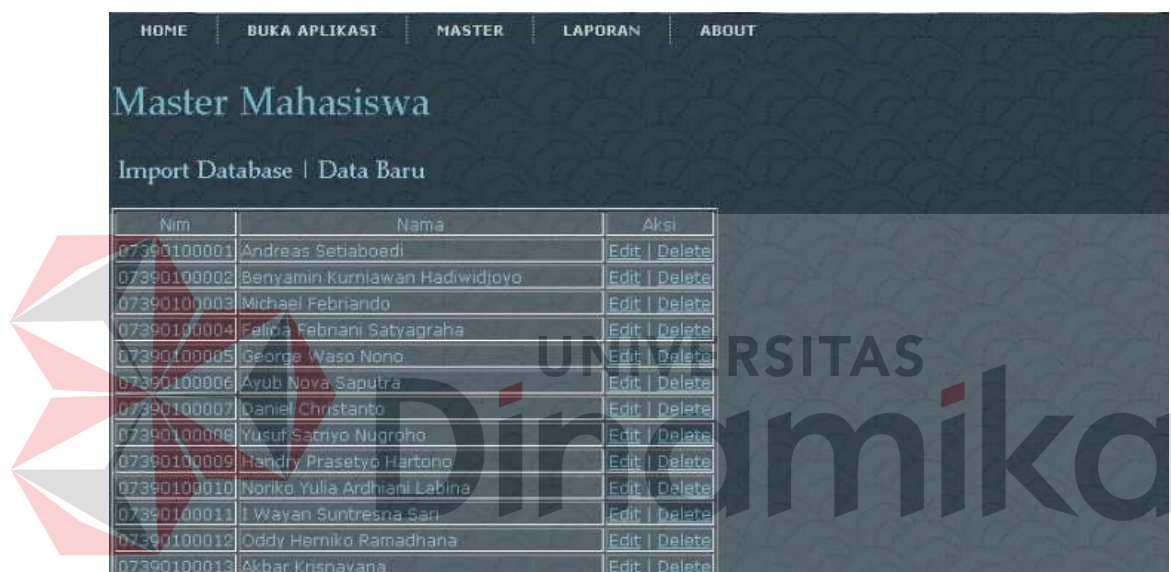
Nama Field	Data-1	Data-2
Nim	07410100190	07410100195
Status	Praktikan Baru	Praktikan Ada

Tabel 4.4. *Test Case* Data Praktikan

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
1	Deskripsi praktikan yang valid	Memasukkan data 1 (satu) seperti pada tabel4.3	Akan muncul pesan penambahan praktikan berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Muncul pesan 3. Data dapat tersimpan

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
2	Deskripsi praktikan yang tidak valid	Memasukkan data 2 (dua) seperti pada tabel 4.3	Akan muncul pesan penambahan praktikan tidak berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Muncul pesan

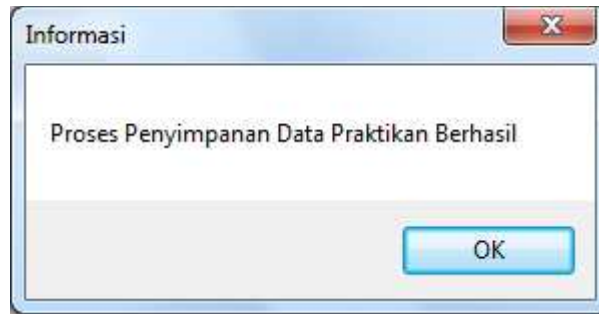
Berdasarkan uji coba No. 1 (satu) pada Tabel 4.4 ditunjukkan pada Gambar 4.10. Gambar 4.11 menjelaskan hasil uji coba No. 2 (dua) pada Tabel 4.4.



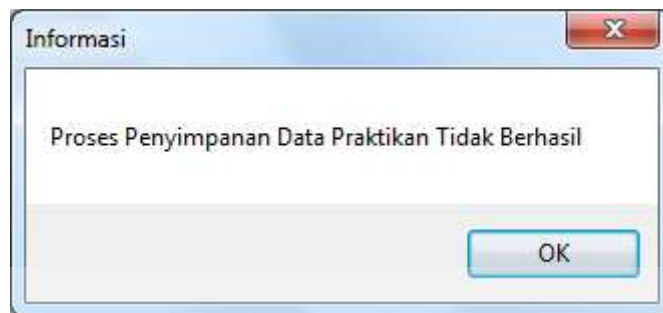
Gambar 4.8. Menu *Master Mahasiswa (Admin)*



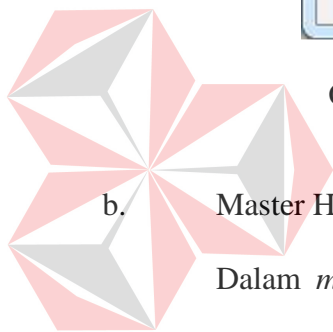
Gambar 4.9. Menu *EditMaster Mahasiswa (Admin)*



Gambar 4.10. Pesan Data Praktikan Telah Diberikan Akses (Data Telah Tersimpan)



Gambar 4.11. Pesan Data Praktikan Sudah Ada



b. Master Hari

Dalam *master* hari ini berfungsi sebagai mengelolah data hari. Dapat dilihat pada Gambar 4.12. Klik edit akan menampilkan halaman edit data hari, seperti pada Gambar 4.13. *Test case* data hari dapat dilihat pada Tabel 4.5.

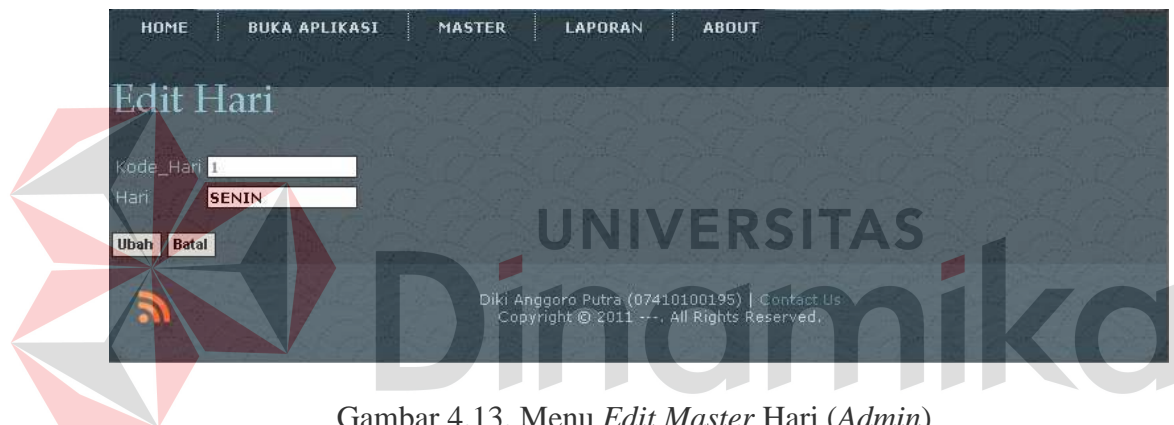
Tabel 4.5. *Test Case* Data Hari

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
1	Deskripsi praktikan yang valid	Memasukan data yang dibutuhkan lengkap.	Akan muncul pesan perubahan hari berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Muncul pesan 3. Data dapat tersimpan
2	Deskripsi praktikan yang tidak valid	Kosong	Akan muncul pesan perubahan hari tidak berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Muncul pesan

Berdasarkan uji coba No. 1 (satu) pada Tabel 4.5 ditunjukkan pada Gambar 4.14. Gambar 4.15 menjelaskan hasil uji coba No. 2 (dua) pada Tabel 4.5.



Gambar 4.12. Menu *Master Hari (Admin)*



Gambar 4.13. Menu *Edit Master Hari (Admin)*



Gambar 4.14. Menu Pesan Data Hari Telah Diberikan Akses (Data Telah Tersimpan)



Gambar 4.15. Pesan data hari tidak berhasil

c. Master Sesi

Dalam *master* sesi ini berfungsi sebagai mengelolah data sesi. Dapat dilihat pada Gambar 4.16. Klik edit akan menampilkan halaman edit data hari, seperti pada Gambar 4.17. *Test case* data hari dapat dilihat pada Tabel 4.6.

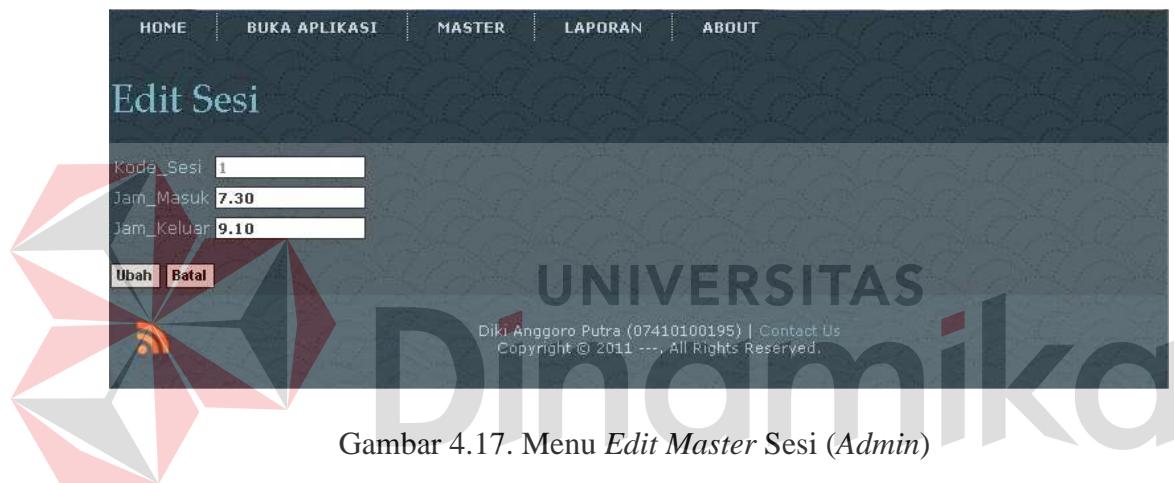
Tabel 4.6. *Test Case* Data Sesi

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
1	Deskripsi praktikan yang valid	Memasukan data yang dibutuhkan lengkap.	Akan muncul pesan perubahan sesi berhasil	1. Sukses 2. Muncul pesan 3. Data dapat tersimpan
2	Deskripsi praktikan yang tidak valid	Kosong	Akan muncul pesan perubahan sesi tidak berhasil	1. Sukses 2. Muncul pesan

Berdasarkan uji coba No. 1 (satu) pada Tabel 4.6 ditunjukkan pada Gambar 4.18. Gambar 4.19 menjelaskan hasil uji coba No. 2 (dua) pada Tabel 4.6.



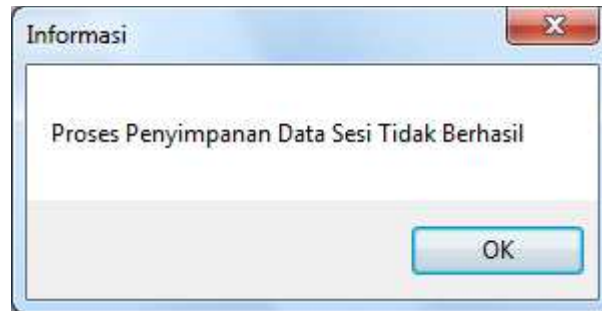
Gambar 4.16. Menu *Master Sesi* (Admin)



Gambar 4.17. Menu *Edit Master Sesi* (Admin)



Gambar 4.18. Pesan Data Sesi Telah Diberikan Akses (Data Telah Tersimpan)



Gambar 4.19. Pesan Data Sesi Tidak Berhasil

4. Menu Laporan

Dalam Menu Laporan terdapat Sub Histori Mahasiswa, dan Histori Kecurangan seperti terlihat pada Gambar 4.20.



Gambar 4.20. Menu Laporan (Admin)

a. Histori Mahasiswa

Histori mahasiswa ini berfungsi sebagai laporan histori praktikan mengakses. Tanggal mulai dan tanggal akhir berfungsi sebagai *filter* laporan tanggal yang akan dipilih. Tombol proses digunakan untuk menampilkan sesuai *filter* yang sudah dipilih. Tombol cetak digunakan untuk mencetak laporan histori mahasiswa. Seperti terlihat pada Gambar 4.21. Contoh laporan yang sudah di cetak dapat dilihat pada lampiran 3.

Kode Logging	Nim	Hari	Jam Masuk	Waktu Login	Ip Komputer	Nama Komputer	Jenis Browser
LOG0013	07410100195	JUMAT	13.30	5/20/2011 12:00:00 AM	172.25.87.160	LAB	IE
LOG0011	07410100195	KAMIS	9.30	5/19/2011 12:00:00 AM	172.25.87.160	LAB	IE
LOG0012	07410100356	KAMIS	9.30	5/19/2011 12:00:00 AM	172.25.87.160	LAB	IE
LOG0009	07410100212	SABTU	13.30	5/17/2011 12:00:00 AM	127.0.0.1	LAB	IE
LOG0010	07410100219	SABTU	15.30	5/17/2011 12:00:00 AM	127.0.0.1	LAB	IE
LOG0008	07410100215	JUMAT	18.30	5/16/2011 12:00:00 AM	127.0.0.1	LAB	IE
LOG0007	07410100303	KAMIS	15.30	5/15/2011 12:00:00 AM	127.0.0.1	LAB	IE
LOG0006	07410100295	RABU	13.30	5/14/2011 12:00:00 AM	127.0.0.1	LAB	IE
LOG0005	07410100160	SELASA	11.30	5/13/2011 12:00:00 AM	127.0.0.1	LAB	IE

Gambar 4.21. Menu Histori Mahasiswa (*Admin*)

b. Histori Kecurangan

Histori kecurangan ini berfungsi sebagai laporan histori kecurangan paraktikan. Tanggal mulai dan tanggal akhir berfungsi sebagai *filter* laporan tanggal yang akan dipilih. Pilih aplikasi berfungsi sebagai *filter* laporan aplikasi yang akan ditampilkan. Tombol proses digunakan untuk menampilkan sesuai *filter* yang sudah dipilih. Tombol cetak digunakan untuk mencetak laporan histori mahasiswa. Seperti terlihat pada Gambar 4.22. Contoh laporan yang sudah di cetak dapat dilihat pada lampiran 4.

Kode Pelanggaran	Kode Logging	Nim	Nama Praktikan	Hari	Jam Masuk	Nama Komputer	Keterangan	Status
P0006	LOG0011	07410100195	Diki Anggoro Putra	KAMIS	9.30	LAB	Melakukan Kecurangan	Buka
P0001	LOG0001	07410100196	Sofyan	SENIN	15.30	LAB	Melakukan Kecurangan	Buka
P0003	LOG0004	07410100199	Ali Zainal Abidin	SELASA	9.30	LAB	Melakukan Kecurangan	Buka
P0005	LOG0010	07410100219	Johan Agus Susanto	SABTU	15.30	LAB	Melakukan Kecurangan	Tutup
P0004	LOG0006	07410100295	Dwi Wibowo	RABU	13.30	LAB	Melakukan Kecurangan	Buka

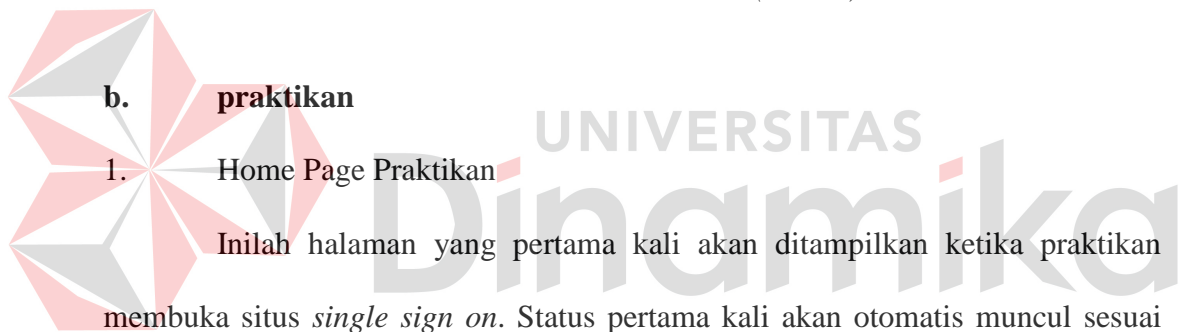
Gambar 4.22. Menu Histori Kecurangan

5. About

Dalam Menu About ini sebagai ucapan terima kasih kepada orang-orang yang telah membantu menyelesaikan TA kali ini. Seperti yang terlihat pada Gambar 4.23.



Gambar 4.23. Menu *About* (Admin)



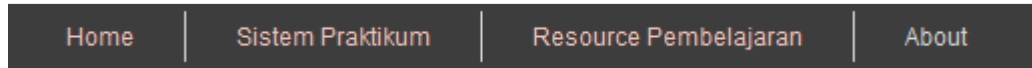
dengan *user* waktu *login* Windows.



Gambar 4.24. Menu Utama (Praktikum)

2. Menu Utama Praktikan

Dalam menu utama praktikan terdapat Home dan About. Seperti terlihat pada gambar Gambar 4.25.



Gambar 4.25. Menu Utama (Praktikan)

Data praktikan yang digunakan terlihat pada Tabel 4.7 dan test case data praktikan dapat dilihat pada Tabel 4.8.

Tabel 4.7. Data Praktikan

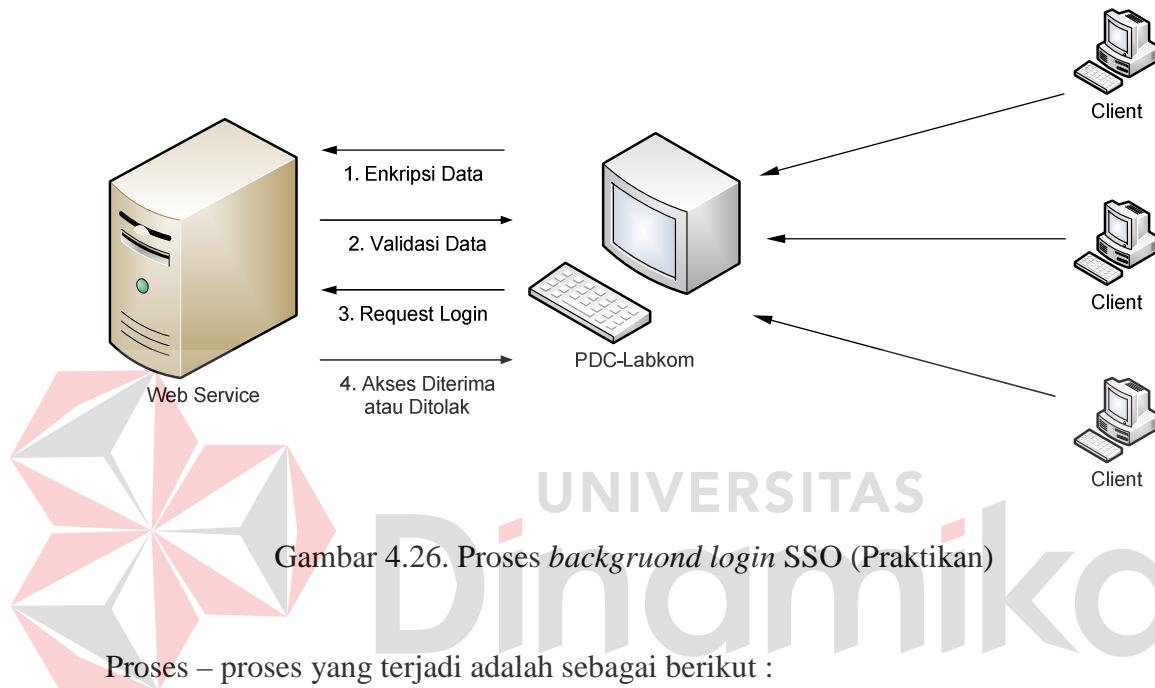
Nama Field	Data-1	Data-2
Nim	07410100195	07410100190
Status	Buka	Tutup

Tabel 4.8. Test Case Data Praktikan

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
1	Deskripsi praktikan yang valid	Memasukkan data 1 (satu) seperti pada tabel 4.7	Dalam <i>group boxlogin</i> , praktikan akan diberikan akses masuk	<ol style="list-style-type: none">1. Sukses2. Label pada <i>group box</i> berubah sesuai keterangan3. Praktikan dapat akses
2	Deskripsi praktikan yang tidak valid	Memasukkan data 2 (dua) seperti pada tabel 4.7	Dalam <i>group boxlogin</i> , praktikan tidak diberikan akses masuk	<ol style="list-style-type: none">1. Sukses2. Label pada <i>group box</i> berubah sesuai keterangan3. Praktikan tidak dapat akses

3. Menu Home

Pada saat praktikan mengakses PDC-Labkom proses SSO akan terjadi pada *background login*, sehingga praktikan tidak mengetahui apa yang terjadi sebenarnya di *background login* SSO tersebut. Proses pada *background login* dapat dilihat pada Gambar 4.26.



Gambar 4.26. Proses *background login* SSO (Praktikan)

Proses – proses yang terjadi adalah sebagai berikut :

1. Enkripsi Data

Waktu praktikum dimulai, praktikan akan langsung mengakses PDC-Labkom. Pada saat *client* mengakses PDC-Labkom maka PDC-Labkom akan memanggil *web service*. Lalu PDC-Labkom akan mengirim nim praktikan yang diambil pada saat *login domain*, kemudian akan dibungkus kedalam paket enkripsi data berdasarkan hari praktikum, lalu dikirimkan ke *web service*.

2. Validasi Data

Web service akan menerima paket enkripsi data yang dikirim oleh PDC-Labkom dan akan di deskripsi menggunakan hari praktikum, sehingga

mendapatkan nim dari praktikan tersebut. Nim akan di cocokkan kedalam *database* mahasiswa, apakah nim tersebut ada didalam *database web service SSO*. Jika ada maka proses dapat dilanjutkan untuk validasi nim tersebut akan diberi akses atau tidak.

3. Request Login

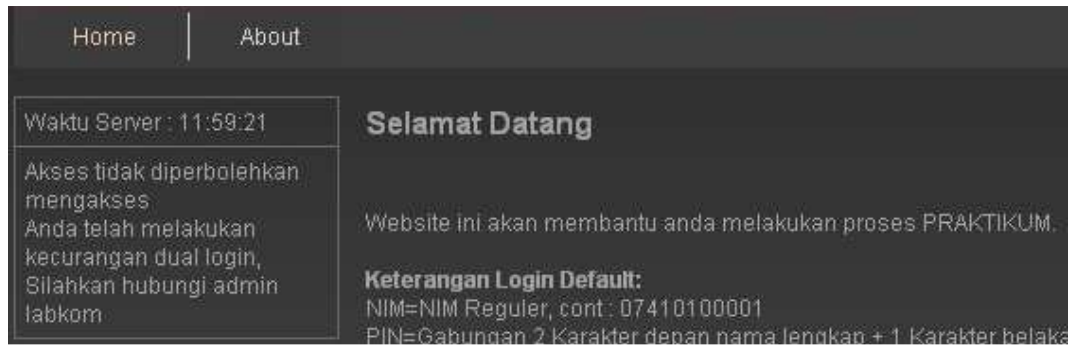
PDC-Labkom akan *request login* pada *web service* berdasarkan nim, ip komputer, kode sesi, kode hari, dan tanggal praktikum dari komputer *client*.

4. Akses Diterima atau Ditolak

Setelah PDC-Labkom *request login*, maka *web-service* pertama akan memeriksa kedalam tabel pelanggaran. Jika nim tersebut ada di dalam tabel pelanggaran dan status praktikan masih tutup, maka praktikan tersebut tidak mendapatkan akses untuk praktikum, untuk akses ditolak dapat dilihat pada Gambar 4.27. Jika nim tidak tercatat melakukan pelanggaran atau statusnya buka, maka akan diperiksa dengan tabel *logging*. Apakah komputer tersebut pada ip komputer, kode sesi, kode hari, dan tanggal yang sama sudah ada praktikan yang memakai. Jika komputer sudah dipakai, praktikan tersebut akan dicatat pada tabel pelanggaran dan tidak diberikan akses praktikum. Jika belum ada yang memakai, maka praktikan tersebut akan diberikan akses untuk melanjutkan praktikum untuk akses diterima dapat dilihat pada Gambar 4.28.



Gambar 4.27. Halaman *Home* (Praktikan)



Gambar 4.28. Halaman Akses Praktikan Ditolak

4.3.2 Analisa hasil uji coba sistem

A. analisis fitur utama sitem

Pengujian fitur utama sistem berdasarkan uji *blackbox* dinyatakan sudah cukup baik karena semua target sudah bisa terpenuhi. Kemudian uji dengan menggunakan kuesioner menyatakan bahwa calon pengguna sistem sudah merasa cukup puas terhadap kemampuan penyediaan informasi oleh sistem. Hasil kuesioner dengan jumlah responden 15 orang dirangkum sebagai berikut :

1. Sampel mahasiswa (14 orang)
2. Admin Labkom

Tabel 4.9. Tabel kuesioner

No	Kriteria	Hasil				
		Sangat Kurang	Kurang	Cukup	Baik	Sangat Baik
1	Proses SSO					
2	Kemudahan Navigasi Program					
3	Informasi yang ditampilkan					

Admin Labkom (1 orang):

Proses SSO :

Kemudahan Navigasi Program :

Informasi yang Ditampilkan :

Jadi, hasil secara keseluruhan untuk tiap kriteria dapat dilihat pada Tabel

4.10. Untuk detail angket dapat dilihat pada lampiran 5.

Tabel 4.10. Tabel rangkuman hasil kuesioner

No	Kriteria	Hasil (orang)				
		Sangat Kurang	Kurang	Cukup	Baik	Sangat Baik
1	Proses SSO			1	10	3
2	Kemudahan Navigasi Program			2	12	
3	Informasi yang ditampilkan		1	8	3	2

Analisis untuk hasil tersebut adalah sebagai berikut :

1) Proses SSO

Hasil penilaian ini cukup kelas, yaitu lebih dari cukup, dan mayoritas responden .

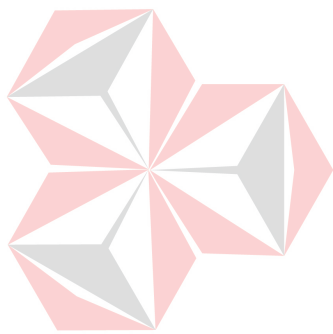
2) Kemudahan Navigasi Program

Hasil penilaian untuk kriteria ini cukup jelas, yaitu lebih dari cukup, dan mayoritas responden (0) menyatakan bahwa kemudahan navigasi program sudah baik.

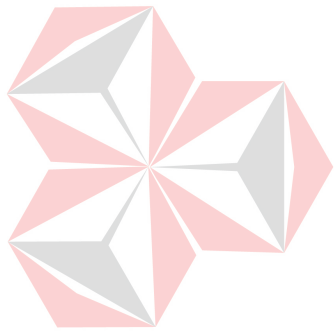
3) Informasi yang Ditampilkan

Hasil penilaian untuk kriteria ini juga cukup jelas, yaitu lebih dari cukup, dan mayoritas responden (0) menyatakan bahwa informasi yang ditampilkan sudah baik. Karena kriteria ini merupakan poin penilaian responden yang terpenting,

dan hasilnya baik, maka kemampuan sistem menyampaikan informasi yang dibutuhkan *user* dapat dinyatakan sudah cukup baik.



UNIVERSITAS
Dinamika



UNIVERSITAS
Dinamika

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari pembuatan implementasi sistem autentifikasi terintegrasi pada *domain controller* dan *application server* labkom STIKOM Surabaya adalah sebagai berikut:

1. Implementasi SSO telah mampu menangani kecurangan multi *account* praktikan di Labkom STIKOM Surabaya. Dengan adanya implementasi SSO ini praktikan hanya dapat *login* pada 1 komputer di hari, sesi, dan tanggal yang sama.
2. Sistem yang dibangun telah mampu melakukan pencatatan histori praktikan dan histori pencatatan pelanggaran Labkom STIKOM Surabaya. Admin Labkom dapat melihat laporan-laporan yang dapat melihat praktikan yang melakukan kecurangan *multi account*.
3. *Web service* SSO telah mampu terintegrasi dengan PDC-Labkom dengan baik. *Web service* SSO akan ditaruh pada server Labkom, sehingga *client* Labkom yang menggunakan aplikasi PDC-Labkom dapat langsung mengakses *login* yang akan divalidasi oleh *web service* SSO.

5.2 Saran

Sistem dapat dikembangkan sebagai sarana autentifikasi untuk aplikasi lain yang ada di Labkom Surabaya, seperti aplikasi Hercules yang digunakan Labkom untuk sertifikasi dan aplikasi Poseidon yang digunakan Labkom untuk ujian UTS dan UAS.

DAFTAR PUSTAKA

- Andi. 2004. *Windows Server 2003 Web Edition*. Yogyakarta: C.V Andi Offset.
- Dani, J. 2008. *Pengembangan Kebijakan Keamanan Informasi Pada Perusahaan Jasa Layanan Kurir*. 19 Januari 2011. URL: <http://digilib.ui.ac.id/opac/themes/libri2/detail.jsp?id=126677&lokasi=lokal>.
- Danny, R. & Tommy, R. 2002. *ASP.NET : Your Visual Blueprint for Creating Web Application on the .NET framework*. Inc: Hungry Mind.
- Hursti, J. 1997. *Single Sign On*. Department of Computer Science Helsinki University Of Technology.
- Kendall, K. E. & Kendall, J. E. 2003. *Analisis dan Perancangan Sistem Jilid 1*. Jakarta: Prehallindo.
- Malik, J. J. 2009. *Best Tools Hacking & Recovery Password*. Yogyakarta: C.V Andi Offset.
- Marlinda, L. 2004. *Sistem Basis Data*. Yogyakarta: Andi Offset.
- Mfatihhurrizqi. 2010. *Praktikum dan Download*. 10 Februari 2011. URL: <http://Mfatihhurrizqi.blogspot.com/2010/06/praktikum-dan-download.html>
- Neuschel, R. 1976. *Management System for Profit and Growth*. New York: McGraw-Hill.
- Noprianto. 2005. *Beragam Jenis Application Server*. 23 Mei 2011. URL: http://www.google.co.id/url?sa=t&source=web&cd=1&ved=0CCMQFjAA&url=http%3A%2F%2Fkambing.ui.ac.id%2Fbebas%2Fv22%2FInfoLinux2005%2FPDFLINUX0505%2F34_Utama_05.pdf&rct=j&q=application%20server%20filetype%3Apdf&tbs=ctr%3AcountryID&ei=xi3bTZ26HcbrrQed6N3rDg&usg=AFQjCNG4HK-TyHSe5rlCQ0BnNUFio4yWxQ&cad=rja
- Rafiudin, R. 2005. *Konfigurasi Sekuriti Jaringan Cisco*. Jakarta: PT Elex Media Komputindo.
- Yuswanto, & Subari. 2005. *Mengelolah Database dengan SQL Server 2000*. Jakarta: Prestasi Pustaka.