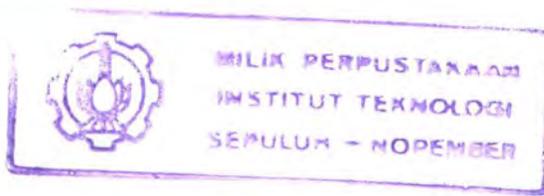


22173/H/05



**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK  
SISTEM BOOKING KOMPUTER di LABORATORIUM  
BERBASIS ACTIVE DIRECTORY SERVER  
MELALUI WEB SERVICE**

**TUGAS AKHIR**



R.SIF  
005.1  
MSI  
P-1  
2005

Disusun Oleh:  
**M. SINANTRYA W**  
**NRP. 5100 100 053**

<b>PERPUSTAKAAN ITS</b>	
<b>Tgl. Terima</b>	5-4-2005
<b>Terima Dari</b>	H
<b>No. Agenda Prp.</b>	271598

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2005**

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK  
SISTEM BOOKING KOMPUTER di LABORATORIUM  
BERBASIS ACTIVE DIRECTORY SERVER  
MELALUI WEB SERVICE**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer**

**Pada**

**Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Mengetahui/Menyetujui  
Dosen Pembimbing**

  
**(Rebrilyan Samopa, S.Kom., M.Kom.)**  
**NIP. 132 206 858**

**SURABAYA  
Januari 2005**

## ABSTRAK

Sering terjadi mahasiswa yang ingin mengerjakan tugas di laboratorium tidak mendapatkan komputer untuk digunakan, karena komputer sedang digunakan semua. Sehingga user tersebut tidak dapat menggunakan komputer.

Dibutuhkan suatu sistem pemesanan yang dapat diakses oleh user yang mempunyai login. Dengan menggunakan Active Directory yang terdapat pada Windows 200x sistem tersebut dapat dibuat, karena pada Active Directory terdapat obyek user dan obyek komputer yang masing – masing mempunyai properti atau atribut yang dapat diakses dengan menggunakan bahasa pemrograman tertentu, yang terpenting pada Active Directory terdapat atribut berupa *logonhours* dan *workstation* yang dapat membatasi dan mengatur pemakaian komputer oleh user tertentu

Dengan menggunakan ASP dan VBScript, sistem tersebut dapat berjalan dalam aplikasi berbasis web, sehingga seorang user yang ingin menggunakan komputer, dapat melihat jadwal komputer terlebih dahulu dari luar kampus. Dengan mengakses melalui ADO dan melakukan *binding*, maka suatu atribut dari obyek dapat diakses dan diupdate.

**Kata Kunci :** *Active Directory, logonhours, workstation*

## **KATA PENGANTAR**

Dengan memanjatkan puji syukur ke hadirat Allah SWT atas karunia dan rahmat-Nya penulis dapat melaksanakan dan menyelesaikan penyusunan laporan Tugas Akhir (TA) ini, yang berjudul:

### **PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK SISTEM BOOKING KOMPUTER di LABORATORIUM BERBASIS ACTIVE DIRECTORY SERVER MELALUI WEB SERVICE**

Tugas Akhir ini dibuat guna memenuhi persyaratan akademik dalam rangka ujian akhir bagi mahasiswa Strata I (S1) Jurusan Teknik Informatika , Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam menyelesaikan Tugas Akhir ini, penulis merasakan banyak pihak yang telah membantu baik pada saat pengerjaan Tugas Akhir, hingga penyusunan laporan Tugas Akhir. Pada kesempatan ini, penulis menyampaikan rasa terima kasih kepada :

1. Allah SWT, atas karunia, rahmat dan ijin-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Kedua Orang Tua penulis Ibu Farida, Bapak Indri, atas doa dan dukungan moral maupun materi yang diberikan kepada penulis.
3. Adik penulis Meidyta dan Faradilla, atas doa dan dukungan serta bantuannya sehingga buku ini dapat selesai.

4. Bapak Febrilyan Samopa, M.Sc selaku Dosen pembimbing Tugas Akhir ini atas kesabaran dan kesediaan waktunya untuk membimbing penulis dalam penyelesaian Tugas Akhir ini.
5. Bapak Wahyu Suadi, M.Kom, selaku dosen wali yang telah memberikan dorongan dan semangat kepada penulis.
6. Seluruh Bapak Ibu Dosen T. Informatika yang telah memberikan ilmu pengetahuan kepada penulis selama masa – masa perkuliahan.
7. Mas Bambang selaku administrator Lab- Prog Sistem Informasi yang telah memberikan banyak bantuan kepada penulis selama pengerjaan Tugas Akhir.
8. Angga, Joko, Ali Sofyan, Ulum, Okhi, Ashar, Try, Deka, Laks, Mbak Ulfa, Puspa atas bantuan dan petunjuknya selama penulis mengerjakan Tugas Akhir.
9. Mas Juned, Pak Hoirul, Dewa, Taqi selaku administrator Lab-RPL tempat penulis mengerjakan Tugas Akhir, terima kasih atas kerjasamanya.
10. Laks, Gunawan, Rudi, Ichsan, selaku sahabat dekat penulis di T.Informatika.
11. Suryo Cumi, Indra Kiwut, Dimas Lambe dan Tommy Koi, selaku sahabat dekat penulis sejak SMU dan yang selalu memberi semangat dan motivasi untuk segera lulus.
12. Teman – teman C10, dan seluruh warga TC yang tidak dapat disebutkan satu persatu terima kasih atas bantuannya selama ini.

Untuk pihak-pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan dukungan dan semangat baik secara langsung maupun tidak langsung penulis mohon ma'af dan terima kasih.

Penulis menyadari sepenuhnya bahwa di dalam penyusunan laporan ini masih terdapat kekurangan – kekurangan baik dari segi materi maupun segi tata bahasa penulisan laporan ini, untuk itu penulis memohon dengan penuh kerendahan hati dan sikap terbuka atas kritik dan saran yang bersifat membangun.

Akhirnya, semoga apa yang telah penulis susun dapat bermanfaat bagi masyarakat pada umumnya dan mahasiswa ITS Surabaya pada khususnya.

Surabaya, Januari 2005

M Sinantrya W

## DAFTAR ISI

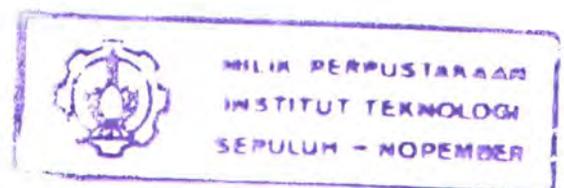
<b>ABSTRAK</b> .....	<b>iii</b>
<b>KATA PENGANTAR</b> .....	<b>iv</b>
<b>DAFTAR ISI</b> .....	<b>vii</b>
<b>DAFTAR GAMBAR</b> .....	<b>x</b>
<b>DAFTAR TABEL</b> .....	<b>xiv</b>
<b>BAB 1 PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Permasalahan.....	2
1.3. Tujuan dan Manfaat .....	2
1.4. Batasan Permasalahan.....	3
1.5. Metodologi Tugas Akhir .....	3
1.6. Sistematika Penulisan .....	4
<b>BAB 2 DASAR TEORI</b> .....	<b>6</b>
2.1. Active Directory .....	6
2.1.1 Konsep dan Fungsi Active Directory .....	6
2.1.2 Struktur Active Directory .....	8
2.2. Active Directory Service Interface .....	10
2.2.1 ADSI Objects.....	10
2.2.2 ADSI Provider .....	11
2.2.3 Schema Management .....	12
2.2.4 Bahasa Pemrograman yang Mendukung.....	17
2.3. Mengakses Active Directory .....	19
2.3.1 <i>Querying</i> Active Directory .....	26
<b>BAB 3 PERANCANGAN PERANGKAT LUNAK</b> .....	<b>30</b>
3.1. Deskripsi Umum.....	30
3.2. Perancangan Data .....	31
3.2.1 Perancangan Data Flow Diagram .....	31
3.2.2 Perancangan Basis Data .....	36
3.2.2.1 Logical Data Model.....	38

3.3. Perancangan Proses User .....	39
3.4. Perancangan Antar Muka.....	41
<b>BAB 4 IMPLEMENTASI PERANGKAT LUNAK.....</b>	<b>45</b>
4.1. Implementasi Data.....	45
4.1.1 Implementasi Basis Data .....	45
4.1.2 Koneksi Basis Data .....	49
4.1.2.1 Koneksi Active Directory .....	49
4.1.2.2 Koneksi Microsoft Acces .....	49
4.2. Implementasi Proses User .....	50
4.2.1 Login .....	50
4.2.2 Proses Untuk Administrator .....	51
4.2.2.1 Reset All User .....	51
4.2.2.2 Reset All Computer.....	52
4.2.2.3 Reset Schedulle.....	53
4.2.2.4 Reset Day.....	54
4.2.2.5 Reset User Hours.....	57
4.2.2.6 Memilih Komputer Untuk User .....	58
4.2.3 Proses Untuk User.....	59
4.2.3.1 Memesan Komputer .....	59
4.2.3.2 Melihat Jadwal yang Dimilikinya .....	63
4.2.3.3 Melihat Pengguna Komputer .....	64
4.2.4 Proses Peringatan .....	64
4.3. Implementasi Antar Muka.....	65
4.3.1 Form Login .....	65
4.3.2 Daftar Komputer .....	65
4.3.3 Jadwal Komputer .....	65
4.3.4 Pengguna .....	66
4.3.5 Jadwal User.....	66
4.3.6 Utama Admin.....	67
4.3.7 Daftar User-Admin .....	67
<b>BAB 5 UJI COBA DAN EVALUASI.....</b>	<b>68</b>

5.1. Lingkungan Pelaksanaan Uji Coba.....	68
5.2. Data Uji Coba.....	68
5.2.1 Uji Coba Administrator Pertama .....	68
5.2.1.1 Reset All User .....	69
5.2.1.2 Reset All Computer.....	69
5.2.1.3 Reset Schedule .....	70
5.2.1.4 Memilih Komputer Untuk User .....	71
5.2.2 Uji Coba User Pertama.....	71
5.2.3 Uji Coba User Kedua.....	72
5.2.4 UjiCoba Admin Kedua.....	72
5.2.4.1 Menghapus Jadwal User.....	73
5.2.4.2 Menambah Jam User .....	73
5.2.5 Uji Coba Proses Peringatan .....	75
5.2.6 Uji Force Log Off .....	75
<b>BAB 6 KESIMPULAN DAN SARAN.....</b>	<b>77</b>
6.1. Kesimpulan.....	77
6.2. Saran.....	77
<b>DAFTAR PUSTAKA .....</b>	<b>79</b>

## DAFTAR GAMBAR

Gambar 2.1 Hirarki Struktur Active Directory .....	8
Gambar 2.2 Contoh Sebuah Tree.....	9
Gambar 2.3 Sistem Kerja ADSI Provider.....	12
Gambar 2.4 Layout umum ADSI.....	13
Gambar 2.5 Schema Container .....	14
Gambar 2.6 Hubungan antar Obyek.....	15
Gambar 2.7 Interface ADSI Untuk Daftar User.....	15
Gambar 2.8 Interface Properti User Account pada ADSI .....	16
Gambar 2.9 Interface Properti User Logon Hours pada ADSI.....	16
Gambar 2.10 Interface Properti User Logon Workstation pada ADSI .....	17
Gambar 2.11 Hubungan Bahasa Pemrograman dan ADSI.....	19
Gambar 2.12 Contoh skrip dalam VBScript bukan untuk web.....	24
Gambar 2.13 Contoh skrip dalam ASP VBScript.....	26
Gambar 2.14 Code untuk koneksi ke Active Directory.....	26
Gambar 2.15 Contoh code query terhadap Active Directory .....	27
Gambar 3.1 Context Free Diagram .....	32
Gambar 3.2 DFD level 1.....	33
Gambar 3.3 Sub-Proses Pengesetan Data Base.....	34
Gambar 3.4 SubProses Pengaturan Jadwal.....	35
Gambar 3.5 Entity Relationship Diagram.....	36
Gambar 3.6 Logical Data Model.....	38
Gambar 3.7 Bentuk Dasar Form Login .....	41
Gambar 3.8 Bentuk Dasar Form Daftar Komputer.....	41
Gambar 3.9 Bentuk Dasar Form Jadwal Komputer .....	42
Gambar 3.10 Bentuk Dasar Form Pengguna .....	42
Gambar 3.11 Bentuk Dasar Form Jadwal User.....	43
Gambar 3.12 Bentuk Dasar Form Utama Admin.....	43
Gambar 3.13 Bentuk Dasar Form Daftar User-Admin .....	44
Gambar 4.1 Physical Data Model .....	45



Gambar 4.2 Tipe Data Tabel kompie .....	46
Gambar 4.3 Struktur Tabel Kompie .....	46
Gambar 4.4 Tipe data tabel User .....	47
Gambar 4.5 Struktur Tabel User .....	47
Gambar 4.6 Algoritma Untuk Tabel Penjadwalan .....	48
Gambar 4.7 Tipe data tabel penjadwalan .....	48
Gambar 4.8 Struktur Tabel Penjadwalan .....	48
Gambar 4.9 Tipe data tabel dipakai .....	49
Gambar 4.10 Tipe data tabel dipakai .....	49
Gambar 4.11 Koneksi Active Directory .....	50
Gambar 4.12 Koneksi Microsof Access .....	50
Gambar 4.13 Algoritma Login Active Directory .....	51
Gambar 4.14 Query User pada Active Directory .....	52
Gambar 4.15 Query untuk menghapus user .....	52
Gambar 4.16 Algoritma dan Query Insert user .....	52
Gambar 4.17 Query Komputer pada Active Directory .....	53
Gambar 4.18 Query untuk menghapus komputer .....	53
Gambar 4.19 Algoritma dan Query Insert Komputer .....	53
Gambar 4.20 Query untuk menghapus pejadwalan .....	54
Gambar 4.21 Query Menghitung Jumlah Komputer .....	54
Gambar 4.22 Query Insert Penjadwalan .....	54
Gambar 4.23 Query Update Untuk Hari Sabtu .....	55
Gambar 4.24 Query Update Untuk Hari Minggu .....	55
Gambar 4.25 Query Update Untuk Hari Senin .....	56
Gambar 4.26 Query Update Untuk Hari Selasa .....	56
Gambar 4.27 Query Update Untuk Hari Rabu .....	56
Gambar 4.28 Query Update Untuk Hari Kamis .....	57
Gambar 4.29 Query Update Untuk Hari Jumat .....	57
Gambar 4.30 Query Update Untuk Mengganti jam user .....	58
Gambar 4.31 Untuk menghapus dan mengganti data tabel dipakai .....	58
Gambar 4.32 Untuk menyimpan data pada userworkstation .....	58

Gambar 4.33 Query Komputer pada Active Directory .....	59
Gambar 4.34 Query untuk menampilkan Jadwal Komputer .....	60
Gambar 4.35 Query untuk mengupdate jadwal dan user.....	60
Gambar 4.36 Binding obyek user.....	61
Gambar 4.37 Mengambil dan mengubah loginhours .....	61
Gambar 4.38 Case pertama.....	62
Gambar 4.39 Case kedua.....	62
Gambar 4.40 Case terakhir .....	62
Gambar 4.41 Pembuatan file Stream untuk Logon Hours.....	63
Gambar 4.42 Put Logon Hours .....	63
Gambar 4.43 Query untuk mengambil jadwal user.....	64
Gambar 4.44 Kode Untuk Proses peringatan.....	64
Gambar 4.45 Form Login .....	65
Gambar 4.46 Form Daftar Komputer .....	65
Gambar 4.47 Form Jadwal Komputer .....	66
Gambar 4.48 Form Pengguna .....	66
Gambar 4.49 Form Jadwal User.....	66
Gambar 4.50 Form Utama Admin.....	67
Gambar 4.51 Form Daftar User Admin.....	67
Gambar 5.1 Permintaan nilai default jam .....	69
Gambar 5.2 Hasil tabel user.....	69
Gambar 5.3 Hasil tabel kompie.....	70
Gambar 5.4 Hasil tabel penjadwalan.....	70
Gambar 5.5 Hasil pengosongan loginhours .....	70
Gambar 5.6 Komputer pada userworkstation test .....	71
Gambar 5.7 Hasil pengiriman waktu pada loginhours test.....	71
Gambar 5.8 Hasil apabila user memilih jadwal user lain .....	72
Gambar 5.9 Hasil pengiriman waktu pada loginhours test2 .....	72
Gambar 5.10 Jadwal sebelum dihapus .....	73
Gambar 5.11 Jadwal sesudah dihapus.....	73
Gambar 5.12 Jumlah jam user sebelum dilakukan operasi.....	74

Gambar 5.13 Jumlah jam baru user test2.....	74
Gambar 5.14 Jumlah jam baru user test.....	74
Gambar 5.15 Hasil Update jam user.....	74
Gambar 5.16 Hasil Pengiriman pesan .....	75

## DAFTAR TABEL

Tabel 2.1 Atribut Umum User .....	27
Tabel 2.2 Atribut alamat user.....	27
Tabel 2.3 Atribut telepon user.....	28
Tabel 2.4 Atribut organisasi user .....	28
Tabel 2.5 Atribut account user.....	29
Tabel 2.6 Atribut umum komputer.....	29
Tabel 2.7 Atribut sistem operasi komputer.....	29
Tabel 4.1 Isi variabel konversi logonhours.....	61
Tabel 5.1 Lingkungan Pengujian Aplikasi pada Server .....	68
Tabel 5.2 Lingkungan Pengujian Aplikasi pada Client.....	68



BAB 1

PENDAHULUAN

# **BAB 1**

## **PENDAHULUAN**

Pada bab ini akan dibahas mengenai latar belakang dan tujuan dari pembuatan tugas akhir, menentukan permasalahan yang akan dipecahkan beserta batasan dari penyelesaiannya. Kemudian metodologi yang digunakan dalam penyelesaian permasalahan serta sistematika dalam penulisannya.

### **1.1. LATAR BELAKANG**

Bertambahnya jumlah komputer yang ada di laboratorium, tidak sebanding dengan jumlah mahasiswa yang ada. Mahasiswa baru yang masuk ke Fakultas Teknologi Informasi rata – rata pertahunnya 100 orang, sedangkan jumlah komputer yang berada dalam laboratorium cenderung tetap.

Sering terjadi mahasiswa yang ingin mengerjakan tugas di laboratorium tidak mendapatkan komputer untuk digunakan, karena komputer sedang digunakan semua. Sehingga user tersebut tidak dapat menggunakan komputer.

Dibutuhkan sebuah sistem yang dapat digunakan oleh seorang user untuk melakukan pemesanan terhadap suatu komputer tertentu dan pada waktu tertentu, sehingga user dapat menggunakan komputer sebaik – sebaiknya.

Setiap komputer di dalam laboratorium terhubung dengan sebuah server yang terinstal sistem operasi windows server 200x. Pada Windows server 200x setiap informasi mengenai obyek yang ada dalam sebuah jaringan dan membantu untuk membuat informasi yang dapat digunakan oleh user disimpan dalam sebuah direktori khusus yang disebut Active Directory Server.

Dalam Active Directory terdapat obyek user dan obyek komputer yang masing – masing mempunyai properti atau atribut yang dapat diakses dengan menggunakan bahasa pemrograman tertentu.

## **1.2. PERMASALAHAN**

Berdasarkan uraian latar belakang yang telah dijelaskan sebelumnya, permasalahan yang akan diselesaikan dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana cara pengaksesan ke dalam Active Directory Server melalui web.
2. Bagaimana membuat jadwal yang kosong agar user dapat memesan komputer.
3. Bagaimana sistem mengetahui jadwal user.

## **1.3. TUJUAN DAN MANFAAT**

Tujuan tugas akhir ini adalah membuat perangkat lunak *system booking* komputer berbasis Active Directory Server. Sistem tersebut merupakan suatu perangkat lunak yang dapat :

- a. Melakukan login ke server.
- b. Digunakan oleh user untuk memesan sebuah komputer pada suatu interval waktu tertentu.
- c. Mengatur jam pemakaian komputer oleh seorang user agar tidak disalahgunakan.

Manfaat yang diperoleh adalah seluruh user dapat menggunakan komputer secara bergantian sesuai dengan jadwal yang telah di pesannya.

#### 1.4. BATASAN PERMASALAHAN

Dalam membangun perangkat lunak sistem booking komputer dengan berbasis Active Directory Server ini ditentukan batasan – batasan sebagai berikut:

1. Perangkat lunak tersebut tidak melakukan pembuatan user account baru.
2. Perangkat lunak tersebut tidak melakukan update terhadap data pada obyek user.
3. Perangkat lunak tersebut tidak melakukan *scheduling* terhadap user secara otomatis.
4. Tidak terdapat perbedaan hirarki antara mahasiswa, dosen dan karyawan.
5. Hanya dapat digunakan oleh user yang memiliki *account* login.
6. Bahasa pemrograman yang digunakan adalah VBScript dengan Web Server ASP (*Active Server Provider*).

#### 1.5. METODOLOGI TUGAS AKHIR

Metodologi yang digunakan dalam pembuatan Tugas Akhir ini adalah :

1. Studi literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pembuatan perangkat lunak. Termasuk diantaranya adalah pencarian literatur, diskusi dan pemahaman terhadap topik Tugas Akhir.

Pada tahap ini juga akan dipelajari metodologi dan algoritma yang akan digunakan dalam pembuatan perangkat lunak sehingga membantu pada tahap perancangan dan pembuatan perangkat lunak.

## 2. Desain Perangkat Lunak

Tahap ini adalah tahap pembuatan desain perangkat lunak yang akan dibuat.

## 3. Pembuatan Perangkat Lunak

Dalam tahap ini, dilakukan implementasi berdasarkan rancangan yang telah dibuat pada tahap sebelumnya.

## 4. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba perangkat lunak untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya perangkat lunak dan mengadakan perbaikan jika ada kekurangan.

## 5. Penyusunan Buku Tugas Akhir

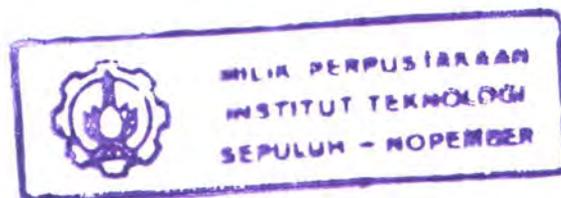
Pada tahap terakhir ini disusun buku sebagai dokumentasi dari pelaksanaan Tugas Akhir. Dokumentasi ini dibuat agar memudahkan orang lain yang ingin mengembangkan sistem informasi ini lebih lanjut.

### 1.6. SISTEMATIKA PENULISAN

Sistematika penulisan Tugas Akhir ini tersusun dalam enam bab, dengan sistematika sebagai berikut:

BAB I berisi pendahuluan, yang menguraikan tentang latar belakang, permasalahan, batasan masalah, tujuan dan manfaat Tugas Akhir, metodologi, dan sistematika penulisan Tugas Akhir.

BAB II berisi penjelasan teori dasar dari Tugas Akhir, yaitu menjabarkan antara lain Active Directory, Active Directory Service Interface, Bahasa pemrograman yang digunakan.



BAB III menjelaskan tentang tahapan proses perancangan perangkat lunak mulai dari deskripsi secara umum sampai perancangan data, perancangan aplikasi dan perancangan antarmuka.

BAB IV berisi implementasi dari perangkat lunak

BAB V berisi hasil uji coba perangkat lunak dengan melakukan uji coba terhadap data – data masukan yang dipakai.

BAB VI berisi kesimpulan yang dapat diambil dari hasil uji coba perangkat lunak. Serta disarankan kemungkinan pengembangan lebih lanjut yang dapat dilakukan berkaitan dengan perangkat lunak ini.



**BAB 2**

**DASAR TEORI**

## BAB 2

### DASAR TEORI

#### 2.1. ACTIVE DIRECTORY

Hal yang menjadi kunci utama dari perangkat lunak ini adalah Active Directory. Active Directory adalah sebuah direktori khusus yang menyimpan informasi mengenai obyek yang ada dalam sebuah jaringan dan membantu untuk membuat informasi tersebut dapat dipergunakan oleh user.

Active Directory (AD) merupakan fitur terpenting Windows 200x yang membedakan dari arsitektur Windows NT. Active Directory adalah pengganti teknologi NT SAM database pada Windows NT, yang berisi konfigurasi *security*, *user*, *group*, dan komputer, dengan banyak kelebihan lain yang dimiliki AD.

##### 2.1.1 Konsep dan Fungsi Active Directory

Active Directory adalah *directory service* yang menyimpan konfigurasi jaringan baik *user*, *group*, komputer, hardware, serta berbagai *policy* keamanan dalam satu basis data terpusat. Peranan AD dalam jaringan dapat diumpamakan sebagai buku telepon, yang menyimpan daftar alamat dan informasi penting untuk mengenali berbagai obyek dalam jaringan.

Peran utama AD adalah menyediakan sarana untuk melakukan administrasi jaringan secara terpusat baik di level domain maupun lintas domain, selama antar domain tersebut masih berada dalam satu *forest*.

Kehadiran AD semakin memudahkan administrator dalam mengelola jaringan sehingga tidak diperlukan kehadiran administrator secara fisik, karena

konfigurasi user, *group*, dan komputer dapat dilakukan secara remote. Hal tersebut dimungkinkan dengan adanya Global Catalog (GC) yang menyimpan konfigurasi komputer dan obyek jaringan, dimana GC tersebut dapat diakses dari manapun di dalam jaringan dengan memanfaatkan infrastruktur AD.

Kemudahan dan fitur yang ditawarkan AD antara lain adalah :

### **1) *Simplified Administration***

AD menyediakan “single point” dalam hal administrasi semua sumber daya jaringan. Seorang administrator dapat melakukan login dari komputer manapun di dalam jaringan dan melakukan konfigurasi terhadap obyek dan setiap komputer dalam jaringan.

### **2) *Scalability***

AD mampu mengelola sampai dengan jutaan obyek, dibandingkan arsitektur Windows NT yang “hanya” mampu menangani maksimal 40000 obyek dalam satu domain.

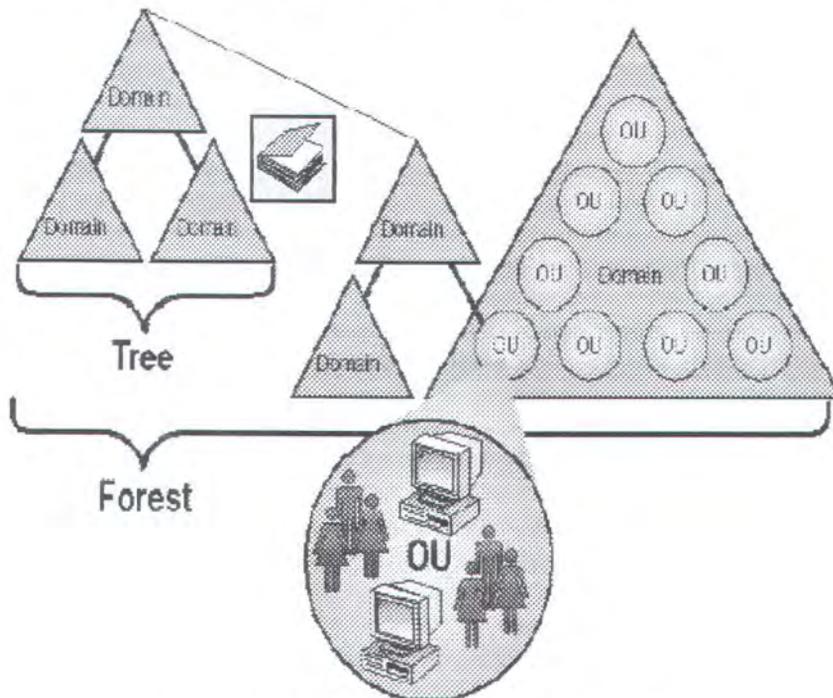
### **3) *Open Standard***

AD kompatibel dan mendukung berbagai protokol dan teknologi standar yang ada, antara lain LDAP dan LDIF, sehingga AD dapat berkomunikasi dengan Novell Directory Service dan teknologi lain yang menggunakan LDAP. Support terhadap HTTP memungkinkan AD diakses dari web browser dan berbagai bahasa pemrograman pengakses data. Windows 2000 juga mengadopsi Kerberos 5 sebagai protokol otentifikasinya, sehingga kompatibel dengan berbagai produk yang menggunakan protokol sejenis. Sistem penamaan domain dalam AD menggunakan standar DNS name,

sehingga nama domain Windows 2000 merupakan standar penamaan domain yang digunakan di internet, maka lebih mudah melakukan koneksi dengan internet.

### 2.1.2 Struktur Active Directory

AD terdiri dari berbagai obyek, yang merupakan representasi obyek-obyek yang terdapat di dalam jaringan baik hardware, user, maupun domain. Hirarki struktur AD tersebut dapat dilihat pada gambar 2.1



Gambar 2.1 Hirarki Struktur Active Directory

**Object** : adalah istilah yang digunakan untuk menyebut suatu unit tertentu yang terdapat di dalam jaringan, misalnya user, group, printer, ataupun *shared folder*.

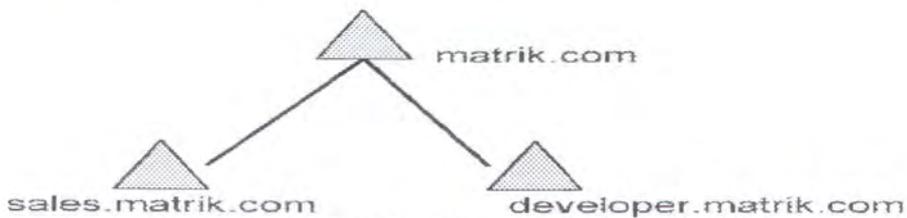
**Container** : merupakan “wadah” yang di dalamnya terisi berbagai macam obyek.

**Organizational Unit (OU)** : adalah representasi *Container* yang didalamnya berisi berbagai macam obyek. OU merupakan kesatuan terkecil dimana

pengaturan *Group Policy* dalam AD dapat diterapkan. Biasanya OU tersebut mencerminkan kesatuan organisasi tertentu dalam jaringan, misalnya dapat didefinisikan OU untuk *Sales*, *Marketing*, *Direksi*, dan sebagainya.

**Domain** : merupakan kesatuan jaringan terkecil, yang didalamnya berisi berbagai obyek dan OU. Domain merupakan *security boundary*, sehingga seluruh obyek dalam satu domain berada dalam otoritas *security* yang sama. Sebuah organisasi dapat memiliki lebih dari satu domain dalam jaringannya, tergantung pada kebutuhan bisnis maupun *policy* keamanannya.

**Tree** : adalah gabungan dari beberapa domain yang masing-masing masih berada dalam satu induk *namespace*. Misalnya dibentuk suatu *tree* dengan induk domain *matrik.com*, dan di level bawahnya terdapat dua domain bernama *sales.matrik.com* dan *developer.matrik.com*. Contoh dari bentuk *tree* tersebut dapat dilihat pada gambar 2.2



Gambar 2.2 Contoh Sebuah Tree

**Forest** : beberapa *tree* dapat bergabung menjadi sebuah *forest* dan masing-masing domain tersebut menggunakan *namespace* yang berbeda. Domain yang berada dalam satu *forest* menggunakan global katalog yang sama, sehingga informasi konfigurasi dan obyek jaringan antar domain dalam satu forest dapat saling pertukarkan dan diakses secara terpusat.

Active Directory menyediakan sebuah tujuan untuk mengatur user, *account*, klien, server dan aplikasi berbasis windows. Juga membantu organisasi mengintegrasikan sistem tanpa menggunakan windows dengan aplikasi berbasis windows dan peralatan windows yang kompatibel, hal tersebut menggabungkan direktori dan meringankan pengaturan seluruh jaringan sistem operasi. Perusahaan juga dapat menggunakan Active Directory untuk memperluas sistem secara aman ke internet. Active Directory dapat meningkatkan nilai investasi jaringan yang telah ada dan menurunkan total biaya dari penggunaan komputer dengan membuat sistem operasi jaringan pada windows lebih mudah diatur, aman dan saling beroperasi.

## **2.2. ACTIVE DIRECTORY SERVICE INTERFACE**

ADSI adalah sebuah set dari COM *programming interface* yang mempermudah *customers* dan *Independent Software Vendors (ISVs)* untuk membuat aplikasi yang terdaftar, akses dan mengatur *multiple* directory service dengan sebuah tatap muka yang telah di set dan ditentukan.

### **2.2.1 ADSI Objects**

*Active Directory Service Interface Object* adalah Obyek COM yang merepresentasikan persistensi obyek dalam directory service. Sebuah obyek Active Directory Service Interface dapat direkayasa menggunakan satu atau lebih *interface* COM. Obyek ADSI dibagi dalam dua group : Directory Service Leaf Object dan Directory Service Container Object. Sebuah *Container Object* dapat mengandung obyek ADSI lain. Sebuah *Leaf Object* tidak dapat mengandung

obyek ADSI. Pembagian dari tipe sebuah obyek ke sebuah *host* dan satu atau lebih *object dependent* melakukan sebuah pengelompokan logis dari properti dan metode. Pembagian ini tidak merefleksikan struktur dari directory. Hubungan *host* dan *object dependent* harus tidak keliru dengan kontainer dan isi relationship yang sebelumnya adalah karakteristik ADSI.

### 2.2.2 ADSI Provider

Sebuah ADSI provider mengandung implementasi dari obyek ADSI dan *object dependent* untuk keterangan *namespace*. ADSI provider mempunyai beberapa *namespace* yang dapat digunakan untuk melakukan akses ke dalam Active Directory, diantaranya adalah :

#### 1. LDAP

*Namespace* LDAP didapatkan pada :

- Windows 2000 Active Directory
- Exchange Directory Services
- Netscape Directory Server
- Site Server

#### 2. WinNT

*Namespace* WinNT didapatkan pada :

- Windows NT 4.0
- Windows 2000 Active Directory

#### 3. NetWare Bindery

- Netware Bindery Servers
- Other “Compatible” Servers

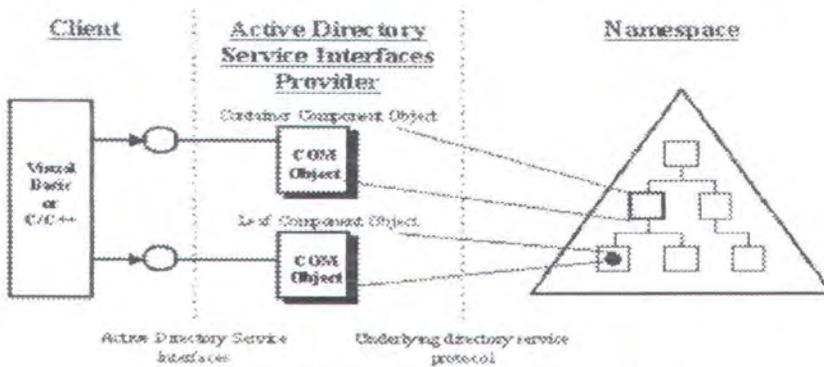
#### 4. Novell NDS

- NDS

#### 5. Internet Information Server

- Internet Information Server 4.0

Gambar 2.3 menunjukkan bagaimana klien hanya untuk mendapatkan dan menggunakan interface pada sebuah obyek, dan tidak secara mendetail mengenai tempat dan bagaimana perangkat lunak dari sebuah obyek diimplementasikan.



Gambar 2.3 Sistem Kerja ADSI Provider

### 2.2.3 Schema Management

ADSI menyediakan pembentukan obyek sehingga manipulasi *directory service* dapat diseragamkan dengan *namespace*. Bagaimanapun juga, sebuah obyek ADSI dalam direktori apapun dapat mempunyai lebih dari satu fungsi dari yang telah dispesifikasi dalam ADSI. Sebuah direktori juga dapat mengandung obyek yang tidak didefinisikan sama sekali oleh ADSI. Sebagai tambahan, ada *directory service* yang dapat diperluas yang memperbolehkan skema dasarnya untuk dimodifikasi dan obyeknya dapat dirubah oleh administrator dan vendor perangkat lunak. Perluasan obyek dipegang oleh *Schema Management ADSI Object*. Obyek ini digunakan untuk :

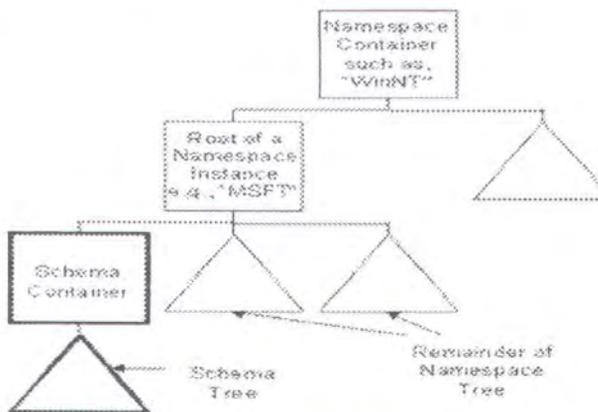
- Mencari definisi obyek;
- Memperluas definisi obyek.

*Schema Management Object* dapat digunakan untuk mencari dan memodifikasi skema dari sebuah *namespace*. Obyek tersebut adalah :

- *Schema Container Object*, yang mengandung skema yang telah ada
- *Class Container Object*, yang mendefinisikan sebuah kelas obyek kelas
- *Property Object*, menjelaskan sebuah properti.
- *Syntax Object*, menjelaskan penulisan yang dapat digunakan dalam definisi sebuah properti.

Obyek – obyek tersebut berbeda dari *Directory Service Object* seperti komponen user, yang pada propertinya tidak dibagi dalam set fungsional.

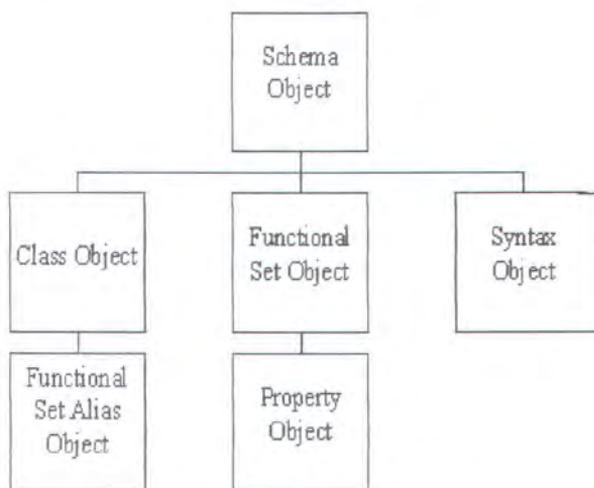
*Schema Container Object* digunakan untuk menambahkan set definisi obyek ke bagian *directory tree*. Pada umumnya, setiap instansi dari direktori memiliki skema sendiri – sendiri. ADSI mempresentasikan hal tersebut dengan menempatkan *Schema Container* sebagai anak dari *directory root*. Bentuk umum dari ADSI ada pada gambar 2.4.



Gambar 2.4 Layout umum ADSI

ADSI tidak membatasi *Schema Container* pada level dari sebuah *tree*. Direktori yang kompleks memperbolehkan banyak skema untuk berada dalam instansi sebuah direktori. Dalam kasus tersebut, *Schema Container* mungkin dapat ditemukan pada bagian lain dari sebuah *tree*.

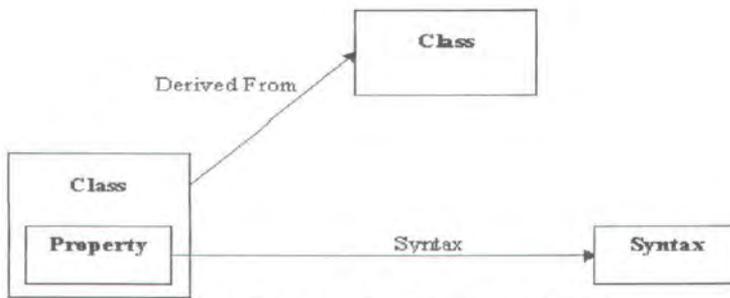
*Schema Container* sendiri adalah sebuah *tree* yang mengandung *Class*, set fungsi, properti dan definisi *syntax*. *Class* baru dan set fungsi dapat di buat dalam kontainer untuk memperluas skema. Set fungsi ditentukan terpisah dari kelas sehingga dapat digunakan dalam pendefinisian berbagai kelas. Contoh dari *Schema Container* bisa dilihat pada gambar 2.5



**Gambar 2.5** *Schema Container*

*Class Container Object* digunakan untuk mendefinisikan kelas sebuah obyek yang dapat dibuat di dalam direktori. Kelas baru bisa didapatkan dari kelas yang telah dengan menggunakan model ADSI.

Gambar 2.6 mengilustrasikan bagaimana sebuah *Class Container Object* berhubungan dengan *class object* yang lain, *property object*, dan *syntax object* untuk membuat sebuah pendefinisian kelas.

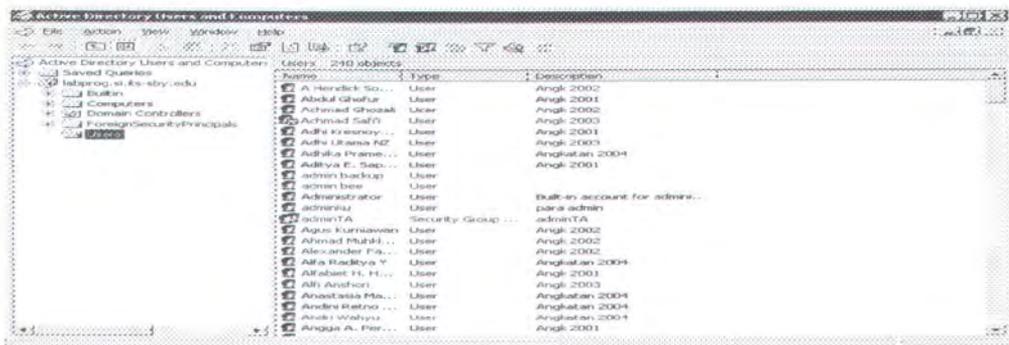


Gambar 2.6 Hubungan antar Obyek

*Property Object*, merupakan bermacam – macam atribut dari sebuah obyek. Pada umumnya setiap obyek user dan komputer dalam ADSI mempunyai atribut – atribut yang menjelaskan user maupun komputer secara lebih detail.

Untuk dapat mengakses *Active Directory Service Interface* diperlukan seorang user dengan hak administrator, *Active Directory Service Interface* mempunyai fungsi dan tampilan yang memudahkan seorang administrator untuk melakukan pengelolaan terhadap user maupun komputer

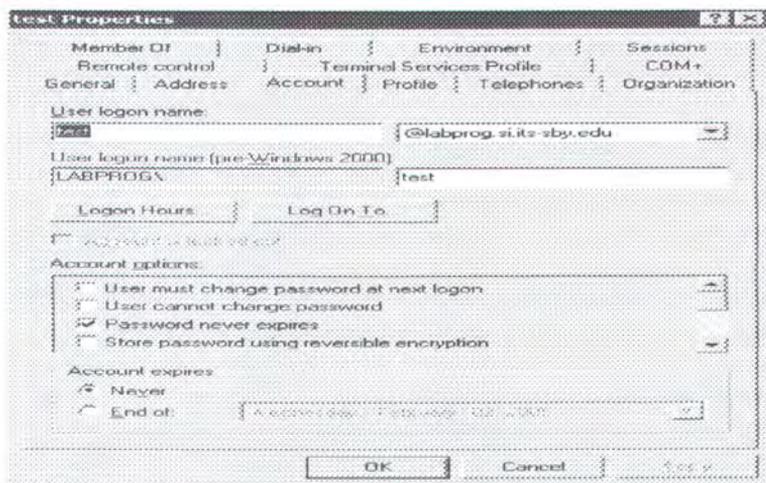
Untuk melakukan pengelolaan terhadap user seorang administrator dapat melakukannya melalui tampilan seperti dalam gambar 2.7.



Gambar 2.7 Interface ADSI Untuk Daftar User

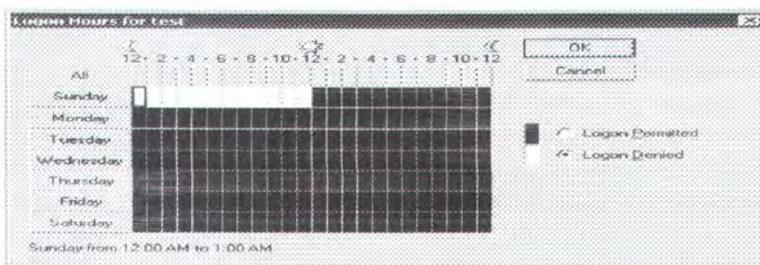
Dengan memilih seorang user maka akan muncul sebuah menu properti seperti yang dapat dilihat pada gambar 2.8, seorang admin dapat memilih sebuah menu agar dapat di kelola. Menu yang berhubungan erat dengan pembuatan

perangkat lunak pada tugas akhir ini adalah menu *account*, maka gambar 2.8 menampilkan bagian menu *account*, yang terdapat data login untuk user tersebut, *Logon Hours* dan *Log On To*.



Gambar 2.8 Interface Properti User Account pada ADSI

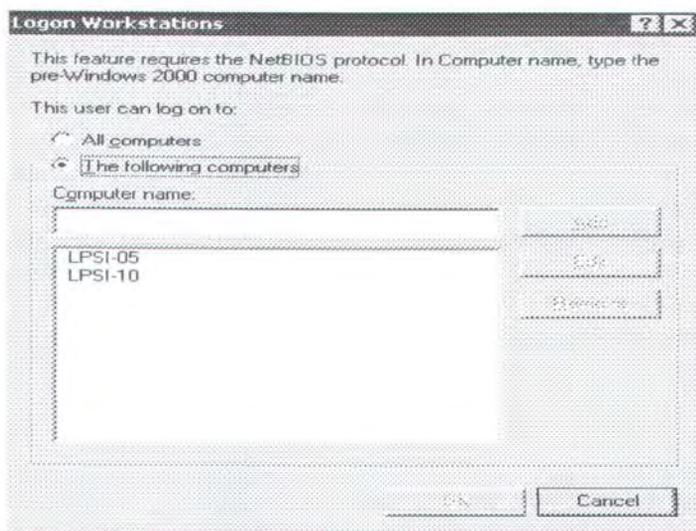
Pada gambar 2.9 menunjukkan tampilan daftar *Logon Hours* dari seorang user yang terdiri dari tujuh hari dan tiap hari terdapat waktu 24 jam yang dibagi menjadi 1 jam setiap kolomnya. Bagian yang berwarna putih menunjukkan bahwa user tersebut tidak dapat melakukan login dalam hari dan waktu tersebut, sedangkan yang berwarna biru menunjukkan hari dan waktu user dapat melakukan login. Urutan *logonhours* dimulai dari hari Minggu pada pukul 07.00-08.00.



Gambar 2.9 Interface Properti User Logon Hours pada ADSI

Pada gambar 2.10 menampilkan daftar komputer yang dapat digunakan oleh user bersangkutan, apabila user tersebut dapat menggunakan komputer yang

mana saja, maka pilihan *All Computer* yang diberi tanda, dan apabila user hanya dapat menggunakan komputer tertentu saja maka administrator harus memasukkan nama – nama komputer yang boleh dipakai oleh user.



Gambar 2.10 Interface Properti User Logon Workstation pada ADSI

#### 2.2.4 Bahasa Pemrograman yang Mendukung

Aplikasi klien ADSI dapat dibuat dalam berbagai bahasa pemrograman. Untuk tugas-tugas administratif pada umumnya, ADSI menetapkan interface dan object yang dapat diakses melalui banyak bahasa secara otomatis, seperti Microsoft® Visual Basic®, dan Microsoft Visual Basic Scripting Edition (VBScript), dan Java, untuk meningkatkan performance dan efisiensi penggunaan bahasa seperti C dan C++.

Integrasi yang lancar dengan Active Server Pages dan VBScript memudahkan untuk membuat aplikasi yang berbasis internet yang dapat mengakses Active Directory Service. Untuk integrasi dengan aplikasi OLE DB, ADSI menyediakan sebuah provider OLE DB dengan mendukung sebuah subset

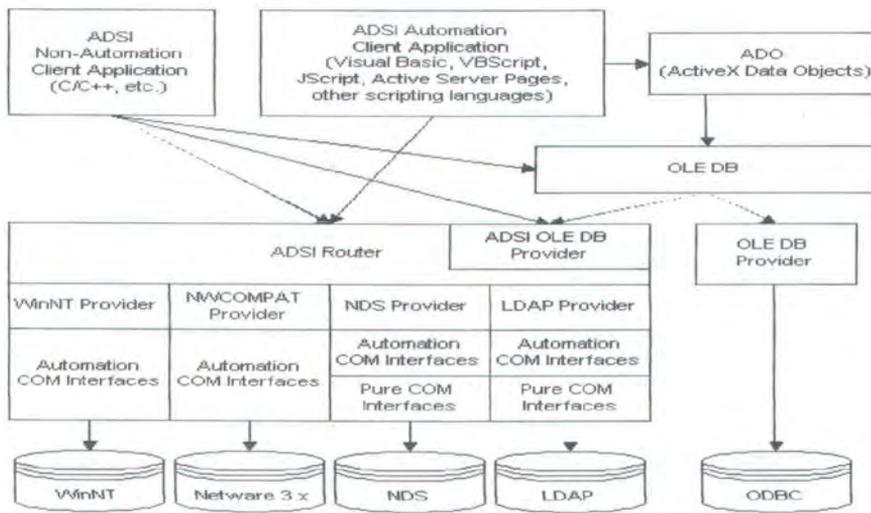
dari interface hasil query OLE DB. Provider OLE DB mendukung akses “read-only” ke Active Directory dan NDS.

Untuk aplikasi internet, menggunakan file Active Server Page (ASP) dapat menulis dan memanipulasi object – object ADSI pada server dan menampilkan hasilnya dalam sebuah halaman web. Dalam *Microsoft Management Console*, administrasi directory service secara snap-ins dapat menggunakan ADSI untuk menemukan kepentingan yang ada dalam directory service. Secara jelas, *Active Directory Service Interface* menyediakan akses secara luas dan bermacam – macam set directory service

Untuk mengakses struktur yang menggunakan APIs, arsitektur ADSI menentukan low-level interface yang tidak mendukung otomatisasi COM yang dapat diakses dari bahasa seperti C dan C++. Interface ini lebih kecil daripada pembungkus COM untuk protokol jaringan ke *directory service*.

Menulis *code* untuk interface yang ditampilkan memperbolehkan aplikasi yang dibuat untuk mencapai directory service untuk semua provider ADSI yang terinstall dan mengintegrasikan data yang dihasilkan. Dengan sedikit perubahan atau tidak pada kode, aplikasi dapat terus mengakses directory service tambahan pada jaringan, sebagai provider ADSI baru yang terinstall

Gambar 2.11 menunjukkan bagaimana ADSI sesuai dengan lingkungan aplikasi. Apakah aplikasi tersebut ditulis dalam Visual Basic, C/C++, VBScript, Microsoft® Jscript®, atau aplikasi Web dengan menggunakan Active Server Pages, Active Directory Service Interfaces mendukung akses dasar directory service yang bersih dan mudah digunakan tanpa harus menggunakan APIs



Gambar 2.11 Hubungan Bahasa Pemrograman dan ADSI

Seperti yang dapat dilihat pada gambar di atas, klien yang tidak mendukung otomatisasi COM mempunyai akses ke seluruh ADSI, termasuk kedua interface COM dengan konvensi penamaan **IDirectoryXXX** interface otomatisasi COM dengan konvensi penamaan **IADsXXX**. Karena klien seringkali meminta informasi dari directory service, Model query yang flexybel dari ADSI melalui OLE DB dan IdirectorySearch lebih effective.

### 2.3. MENGAkses ACTIVE DIRECTORY

Akses ke informasi yang disimpan dalam Active Directory dapat menggunakan dengan *utility management* standar Active Directory. Sangatlah sesuai dan mudah digunakan; akan tetapi ada situasi jika ingin mengakses secara cepat informasi *account user* atau *group*. Tentu saja, dapat menggunakan *terminal service* untuk menyelesaikannya, tetapi hal tersebut ada batasan yang harus disadari.

- Tanpa mempunyai terminal service yang berjalan pada mode aplikasi, hanya dibatasi 2 koneksi bersamaan per server.

- Secara *default*, ijin untuk mengakses *terminal service* pada mode administrasi dibatasi untuk administrator lokal dari server yang dituju. Artinya harus dipastikan siapapun yang mengakses server adalah local administrator atau group non-admin yang diijinkan untuk melakukan koneksi.

Batasan tersebut dapat dihilangkan dengan mendukung akses ke AD melalui web berbasis ASP. ASP adalah teknologi internet Information Server yang mengizinkan pembangkitan dinamik HTML. Skrip, biasanya ditulis dalam VB Script atau JScript, yang dieksekusi ketika diakses oleh klien yang menjalankan browser. Hasil eksekusi akan dikirimkan ke web klien dalam format html dan ditampilkan pada *browser window*. Skrip akan berjalan pada sebuah web server dalam beberapa aspek tidak seperti yang ada pada local, perbedaan yang perlu diambil pada *account* :

- Skrip yang dieksekusi secara local akan berjalan oleh seorang user pada suatu waktu, sementara yang berjalan pada sebuah server biasanya dieksekusi oleh multiple user di waktu yang sama. Hal ini meningkatkan performansi load dan contention ketika mengakses resource pada web server.
- Hasil dari script asp ditampilkan pada browser window, sementara yang berjalan secara local biasanya menggunakan alat lain untuk output
- Dengan mekanisme otentikasi akan lebih kompleks ketika mengesekusi skrip berbasis web

Jelasnya, banyak sekali persoalan yang harus disadari pada kedua kasus. Khususnya kita akan melihat pada eksekusi skrip yang menggunakan ADSI untuk mengakses informasi Active Directory. Pertama, akan ditunjukkan skrip yang berjalan diluar dari ASP(Lokal) dan kemudian akan dijelaskan cara untuk merubah ke form berbasis web.

Skrip pada gambar 2.12, ListADGroups.vbs, menggunakan Active Data Objects (dan OLE DB *Directory Services* provider yang disebut ADsDSOObject) untuk mengekstrak sebuah list group dari active directory. Melibatkan pembuatan object ADO sebagai berikut :

- Sebuah *Connection Object* disimpan pada variabel oCon
- Sebuah *Command object* disimpan pada variabel oCmd
- Sebuah *RecordSet Object* disimpan dalam variabel oRecordSet

Untuk setiap obyek tersebut, akan diset properti yang tepat. Khususnya untuk *Connection Object* :

- Properti provider menetapkan OLE DB provider untuk melakukan koneksi ke Active Directory (ADsDSOObject)
- Variabel sUser adalah nama dari domain yang akan digunakan untuk melakukan koneksi ke Active Directory
- Variabel sPassword adalah password untuk *account* sUser

Untuk *Command Object* :

- Properti ActiveConnection telah diset sebelumnya pada saat membuat obyek oCon
- Properti CommandText terdiri dari parameter *search semicolon* terpisah

Keluaran dari hasil eksekusi *ADO Command Object* disimpan dalam *RecordSet*. Untuk mendapatkan hasilnya, skrip melalui *record* demi *record* dan menampilkan isi masing – masing.

Setelah semua hasil terdata, *Recordset* dan *Connection Objects* ditutup dan di set ke *Nothing*, yang akan membebaskan dari memori.

```
'////////////////////////////////////
'////////////////////////////////////
'/// Name: ListADGroups.vbs
'/// Version:1.0
'/// Date: 09/17/02
'/// Purpose:listing information about Active Directory groups
'/// OS:Windows 2000, XP
'/// Reqs:Account with permissions to read Active Directory
groups properties
'/// Syntax:cscript /nologo ListADGroups.vbs
'////////////////////////////////////
'////////////////////////////////////

Option Explicit
On Error Resume Next

'////////////////////////////////////
'/// Variable Declarations

Dim oRootDSE, oCon, oCmd, oRecordSet
Dim sDomainADsPath, sUser, sPassword, sGroup, sProperties
Dim aDescription, aMember, iCount

'////////////////////////////////////
'/// Extract domain name of the logged on user account

Set oRootDSE = GetObject("LDAP://RootDSE")
sDomainADsPath = "LDAP://" &
oRootDSE.Get("defaultNamingContext")
Set oRootDSE = Nothing

'////////////////////////////////////
'/// Create, configure, and open ADO Connection object
```

```

Set oCon          = CreateObject("ADODB.Connection")

oCon.Provider     = "AdsDSOObject"

sUser            = "UserName"
sPassword       = "Password"

oCon.Open "ADProvider", sUser, sPassword

'////////////////////
'/// Create and configure ADO Command object

Set oCmd         = CreateObject("ADODB.Command")
Set oCmd.ActiveConnection = oCon

sProperties      = "name,AdsPath,description,member"
sGroup          = "*"

oCmd.CommandText = "<" & sDomainAdsPath &
">;(&(objectCategory=group) (name=" & sGroup & "));" & sProperties
& ";subtree"
oCmd.Properties("Page Size") = 100

'////////////////////
'/// Populate ADO RecordSet object with AD Group info

Set oRecordSet = oCmd.Execute

'////////////////////
'/// Display results by listing all records in Recordset

WScript.Echo "Global Groups for the domain " &
Replace(Mid(sDomainAdsPath,11), ",DC=", ".")

While Not oRecordSet.EOF
    WScript.Echo "Name: " & vbTab & oRecordSet.Fields("name")
    WScript.Echo "AdsPath: " & vbTab &
oRecordSet.Fields("AdsPath")
    aDescription = oRecordSet.Fields("description")
    If Not IsNull(aDescription) Then
        WScript.Echo "Description: " & vbTab & aDescription(0)
    End If
    aMember = oRecordSet.Fields("member")
    WScript.Echo "Members: "
    If Not IsNull(aMember) Then
        For icount = 0 to UBound(aMember)
            WScript.Echo vbTab & vbTab & aMember(iCount)
        Next
    End If
    oRecordSet.MoveNext
Wend

'////////////////////
'/// Close Recordset and Connection objects

oRecordSet.Close

```

```

oCon.Close

'////////////////////////////////////
'/// Clean up

Set oRecordSet = Nothing
Set oCon = Nothing

```

**Gambar 2.12** Contoh skrip dalam VBScript bukan untuk web

Untuk mengadaptasikan skrip tersebut ke lingkungan Web Server, perubahan berikut haruslah di lakukan :

- Skrip haruslah berada dalam tanda <% dan %>, yang menandakan bagian dari HTML yang harus diproses sebelum dikirim kembali ke klien.
- Masukkan Createobject harus dirubah ke Server.CreateObject (Server adalah obyek dalam ASP), yang menghilangkan permasalahan ketika *multiple* user menjalankan skrip yang sama
- Statemen Response.Write harus digunakan untuk membangkitkan *code* HTML

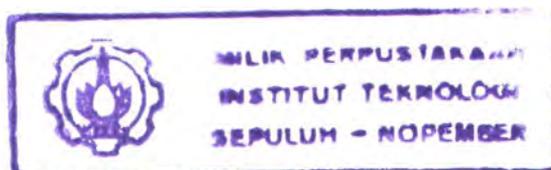
Halaman ASP pada gambar 2.13 mengandung semua perubahan tersebut. Dengan format yang sederhana untuk alasan kemudahan. Skrip membangkitkan tabel yang terdiri dari 4 kolom : group, nama, AdsPath, deskripsi dan daftar anggota.

```

<%@ Language="VBScript" %>
<%
Option Explicit
%>
<HTML>
<HEAD>
<TITLE>Listing of Domain Groups</TITLE>
</HEAD>
<%
Dim oRootDSE, oCon, oCmd, oRecordSet
Dim sDomainADsPath, sUser, sPassword, sGroup, sProperties
Dim aDescription, aMember, iCount

Set oRootDSE = GetObject("LDAP://RootDSE")

```



```

sDomainADsPath      = "LDAP://" &
oRootDSE.Get("defaultNamingContext")
Set oRootDSE        = Nothing

Set oCon            = Server.CreateObject("ADODB.Connection")

sUser               = "UserName"
sPassword           = "Password"

oCon.Provider       = "AdsDSOObject"

oCon.Open "ADProvider", sUser, sPassword

Set oCmd            = Server.CreateObject("ADODB.Command")
Set oCmd.ActiveConnection = oCon

sProperties         = "name,ADsPath,description,member"
sGroup              = "*"

oCmd.CommandText = "<" & sDomainADsPath &
">;(&(objectCategory=group)(name=" & sGroup & "));" & sProperties
& ";subtree"
oCmd.Properties("Page Size") = 100

Set oRecordSet = oCmd.Execute

Response.Write("<strong> Global Groups for the domain: " &
Replace(Mid(sDomainADsPath,11), ",DC=", ".") & "</strong>")

Response.Write("<table border='1'>")
Response.Write("<tr><th>Name</th><th>ADsPath</th><th>Description<
/th><th>Members</th></tr>")
Response.Write("<font size=-2>")
While Not oRecordSet.EOF
    Response.Write("<tr><td>" & oRecordSet.Fields("name") &
"</td>")
    Response.Write("<td>" & oRecordSet.Fields("ADsPath") &
"</td>")
    aDescription = oRecordSet.Fields("description")
    Response.Write("<td> ")
    If Not IsNull(aDescription) Then Response.Write
aDescription(0)
    Response.Write("</td>")
    aMember = oRecordSet.Fields("member")
    Response.Write("<td><select size = '5'> ")
    If Not IsNull(aMember) Then
        For icount = 0 to UBound(aMember)
            Response.Write("<option>" & aMember(icount))
        Next
    End If
    Response.Write("</td></tr>")
    oRecordSet.MoveNext
Wend
Response.Write("</font>")
Response.Write("</table>")

```

```

oRecordSet.Close
oCon.Close

Set oRecordSet = Nothing
Set oCon = Nothing
%>
</BODY>
</HTML>

```

**Gambar 2.13** Contoh skrip dalam ASP VBScript

Seperti yang terlihat, skrip tidak banyak berubah, perbedaan utamanya adalah bahwa *output* perlu diformat berdasarkan spesifikasi HTML.

### 2.3.1 *Querying* Active Directory

Untuk melakukan *query* Active Directory maka diperlukan ADSI provider dan ADO. Jika menggunakan Windows 2000, maka ADSI sudah terinstall.

Untuk melakukan koneksi ke Active Directory, maka diperlukan ADO *Connection Object* dan melakukan seting provider dengan ADSDSOObject dengan *code* dalam gambar 2.14

```

Dim Conn, strRS, RS, strConn
Set Conn = Server.CreateObject("ADODB.Connection")
Set RS = Server.CreateObject("ADODB.Recordset")
Conn.Provider = "ADsDSOObject"
Conn.Open "Active Directory Provider"

```

**Gambar 2.14** *Code* untuk koneksi ke Active Directory

Untuk melakukan *query*, koneksi ke dalam Active Directory haruslah berhasil lebih dahulu. Perbedaan dalam *query* biasa dengan *query* dalam Active Directory adalah jika *query* dalam Active Directory tidak dapat menggunakan `SELECT *`.

Gambar 2.15 adalah contoh bagaimana melakukan *query* terhadap data nama user dan deskripsi dari obyek user dalam Active Directory.

Dikarenakan dalam Active Directory perintah SELECT \* tidak diboleh dilakukan, maka untuk mengakses isi atribut dari obyek user dan komputer dibutuhkan daftar properti yang dapat diakses.

```
strRS = "SELECT displayName, description, FROM 'LDAP://' &
strDomain &" WHERE objectClass = 'user' ORDER by displayName "
RS.Open strRS, Conn,1,1
```

**Gambar 2.15** Contoh *code query* terhadap Active Directory

Untuk mengakses properti user, dengan daftar atribut yang bersifat umum seperti nama, dapat dilihat pada tabel 2.1, untuk mengaksesnya menggunakan isi dari kolom LDAP Name

**Tabel 2.1** Atribut Umum User

Property	LDAP Name	Syntax (Characters)
First name	givenName	Text (64)
Initials	initials	Text (6)
Last name	sn (Surname)	Text (64)
Display name	displayName	Text (256)
Description	description	Text (1024)
Office	physicalDeliveryOfficeName	Text (128)
Telephone number	telephoneNumber	Text (64)
Phone Number (Others)	otherTelephone	Text (64)
E-mail	mail	Text (256)
Web page	wWWHomePage	Text (2048)
Web Page Address (Others)	url	Text

Sedangkan untuk mengakses atribut dari Tab Alamat dapat digunakan daftar atribut dalam tabel 2.2.

**Tabel 2.2** Atribut alamat user

Property	LDAP Name	Syntax (Characters)
Street	streetAddress	Text (1024)
P.O. Box	postOfficeBox	Text (40)
City	l (Locality-Name)	Text (128)
State/province	St (State-Or- Province-Name)	Text (128)
Zip/Postal Code	postalCode	Text (40)
Country/region	Co (Text-Country)	Text (128)
	c (Country-Name)	Text (3)
	countryCode	Integer

Untuk mengakses atribut user dari Tab telepon dapat digunakan daftar atribut pada tabel 2.3

**Tabel 2.3** Atribut telepon user

Property	LDAP Name	Syntax (Characters)
Home	homePhone	Text (64)
Home Phone (Others)	otherHome Phone	Text (64)
Pager	pager	Text (64)
Pager Number (Others)	otherPager	Text (64)
Mobile	mobile	Text (64)
Mobile Number (Others)	otherMobile	Text (64)
Fax	facsimile-TelephoneNumber	Text (64)
Fax Number (Others)	OtherFacsimile-TelephoneNumber	Text (64)
IP phone	ipPhone	Text
IP Phone Number (Others)	otherIpPhone	Text
Notes	Info	Text (1024)

Untuk mengakses atribut organisasi user dapat digunakan daftar atribut pada tabel 2.4

**Tabel 2.4** Atribut organisasi user

Property	LDAP Name	Syntax (Characters)
Department	department	Text (64)
Company	company	Text (64)
Manager	manager	DN; Memilih user dari daftar
Direct reports	directReports	DN

Untuk mengakses atribut *account* user dapat digunakan daftar atribut pada tabel 2.5

**Tabel 2.5** Atribut *account* user

Property/Setting	LDAP Name	Syntax	Description
User logon name (pre-Windows 2000)	sAMAccountName	Text	Digunakan sebagai Account Untuk Login oleh user
Logon Hours	logonHours	(Binary)	Hari dalam seminggu dan jam yang menunjukkan diijinkannya user untuk logon
Log On To/Logon Workstations	userWorkstations	Text (1024)	Daftar Komputer yang dapat digunakan oleh user
Account options	userAccountControl	Yes/No	
Account expires	accountExpires	Date	

Selain untuk user dalam Active Directory juga terdapat cara untuk mengakses atribut komputer. Untuk mengakses atribut komputer yang bersifat umum dapat menggunakan daftar atribut pada tabel 2.6

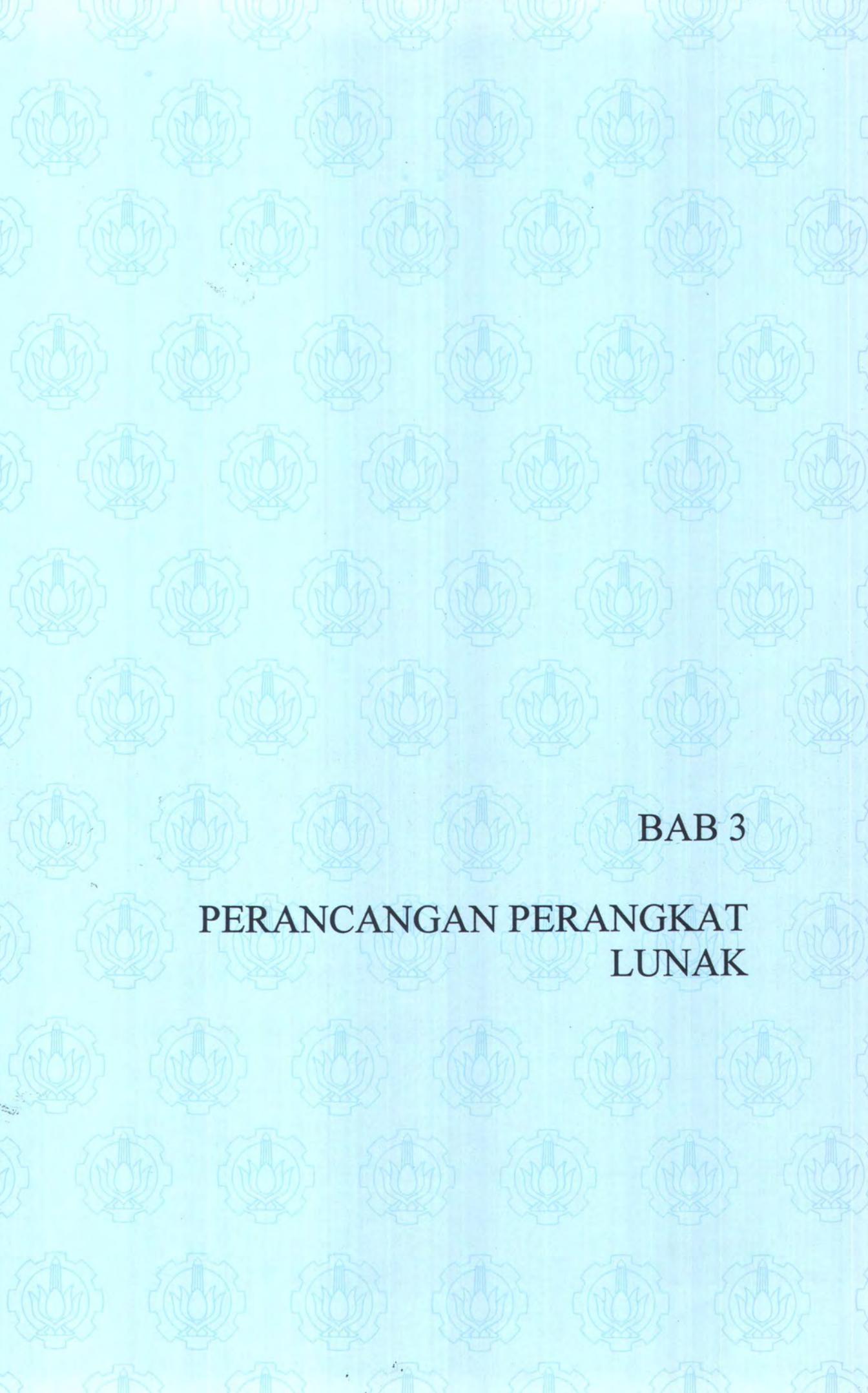
**Tabel 2.6** Atribut umum komputer

Property	LDAP Name	Syntax
Computer name (pre-Windows 2000)	sAMAccountName	Text (256)
DNS name	dNSHostName	Text (2048)
Description	description	Text (1024)
Trust computer for delegation	userAccount-Control	Yes/no

Untuk mengetahui tentang sistem operasi yang digunakan dalam komputer tertentu dapat menggunakan daftar atribut pada tabel 2.7

**Tabel 2.7** Atribut sistem operasi komputer

Property	LDAP Name	Syntax
Name	operating-System	Text
Version	operating-SystemVersion	Text
Service Pack	operating-System-ServicePack	Text



**BAB 3**

**PERANCANGAN PERANGKAT  
LUNAK**

## BAB 3

### PERANCANGAN PERANGKAT LUNAK

#### 3.1. DESKRIPSI UMUM

Perangkat lunak yang dibuat dalam tugas akhir ini adalah perangkat lunak yang dapat melakukan booking komputer yang terhubung dalam jaringan dengan server yang menggunakan Active Directory.

Dengan menggunakan *logonhours* dan *workstation* user dalam Active Directory, maka seorang user dapat diatur kapan dia melakukan dapat melakukan login berdasarkan hari dan jam, dan juga pada komputer mana dia dapat melakukan login.

Dengan menggunakan *logonhours* dan *workstation* user dalam Active Directory, perangkat lunak dapat dibuat. Sistem akan mengambil kedua data tersebut dan kemudian seorang user akan memilih jadwal yang ada pada tampilan tatap muka. Dari pilihan jadwal tersebut akan dikirimkan suatu nilai dan nama komputer ke Active Directory, yang akan menentukan jadwal user untuk memakai komputer yang dipilih.

Dalam penggunaannya, perangkat lunak ini menggunakan metode *first come first serve*, tidak ada perbedaan tingkatan user dalam penggunaan perangkat lunak.

Untuk menggunakan perangkat lunak dalam Tugas Akhir ini, seorang user harus merupakan anggota dari obyek user dalam Active Directory yang digunakan, sehingga dia dapat melakukan login ke dalam perangkat lunak seperti login dalam komputer seperti biasa.

User dapat memilih komputer pada waktu tertentu selama komputer tersebut tidak digunakan oleh user lain, apabila user tersebut memilih komputer yang telah digunakan akan ditampilkan user pengguna dari komputer yang dimaksud atau membatalkan apabila user tersebut yang membatalkannya.

Apabila user yang login mempunyai hak setara dengan administrator, user tidak dapat memesan komputer. User hanya dapat mengelola pemakaian sistem, menghapus pemakaian komputer pada hari sebelumnya, mengeset ulang user, komputer dan jadwal.

### **3.2. PERANCANGAN DATA**

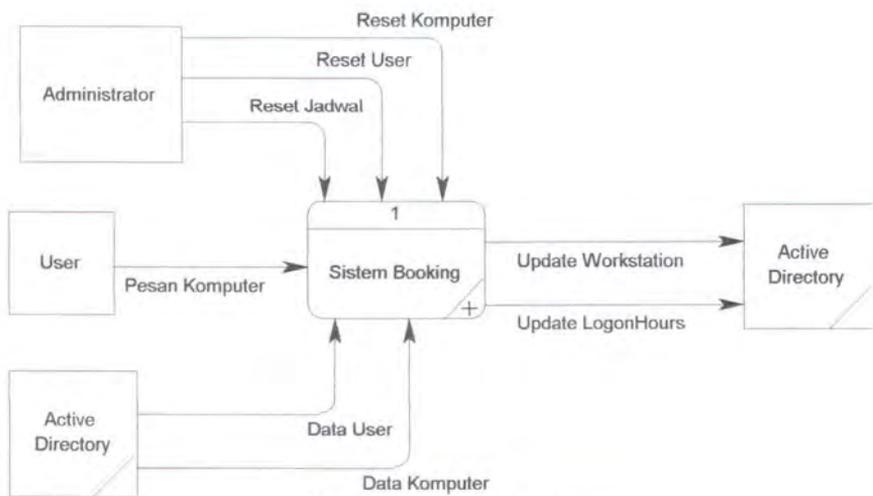
Perancangan data bertujuan untuk menentukan kebutuhan aplikasi di buat . Bentuk data apa saja yang dibutuhkan serta seberapa besar data yang dibutuhkan agar aplikasi dapat berjalan secara optimal.

Pada Perancangan data ini basis data yang digunakan adalah Microsoft Access 2000 yang berfungsi sebagai penyimpan data yang akan membantu untuk pembentukan jadwal penggunaan Komputer.

Dengan adanya database Microsoft Access, maka sistem ini menggunakan dua buah koneksi basis data yaitu koneksi ke *Active Directory Server* dan koneksi ke Microsoft Access.

#### **3.2.1 Perancangan Data Flow Diagram**

Untuk menggambarkan proses – proses yang berjalan dalam sistem, secara umum dapat dilihat pada gambar 3.1 yang merupakan Context Free Diagram atau DFD level 0 dari sistem.



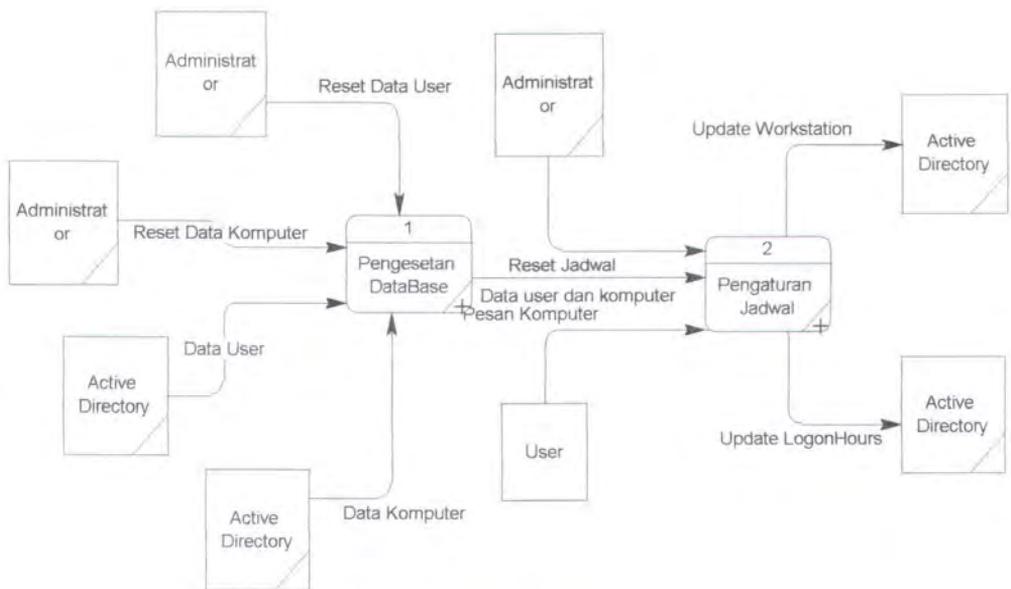
**Gambar 3.1** Context Free Diagram

Pada perangkat lunak yang dibuat pada Tugas Akhir ini terdapat beberapa external entiti, seperti yang dapat dilihat pada gambar 3.1. External entiti yang berperan dalam sistem adalah Administrator, User dan Active Directory.

Seorang administrator mempunyai tugas untuk mengelola sistem, administrator berhak untuk melakukan *setting*, *update* dan *reset* terhadap jadwal. Seorang user hanya dapat melakukan pemesanan terhadap komputer yang belum di pesan oleh user lain. Active Directory sebagai obyek dalam perangkat lunak ini, mempunyai tugas untuk memberikan data user dan data komputer sebagai acuan untuk basis data yang akan dibuat, selain itu Active Directory juga menerima data hasil sistem yang digunakan untuk meng-*update* obyek user pada atribut logonhours dan workstations.

Gambar 3.2 menggambarkan kerja sistem dengan menggunakan DFD level1. Pada proses setting basis data, dibutuhkan masukkan data dari administrator dan Active Directory. Jika seorang administrator memberikan *input* Reset Komputer maka Proses membutuhkan data komputer dari Active Directory

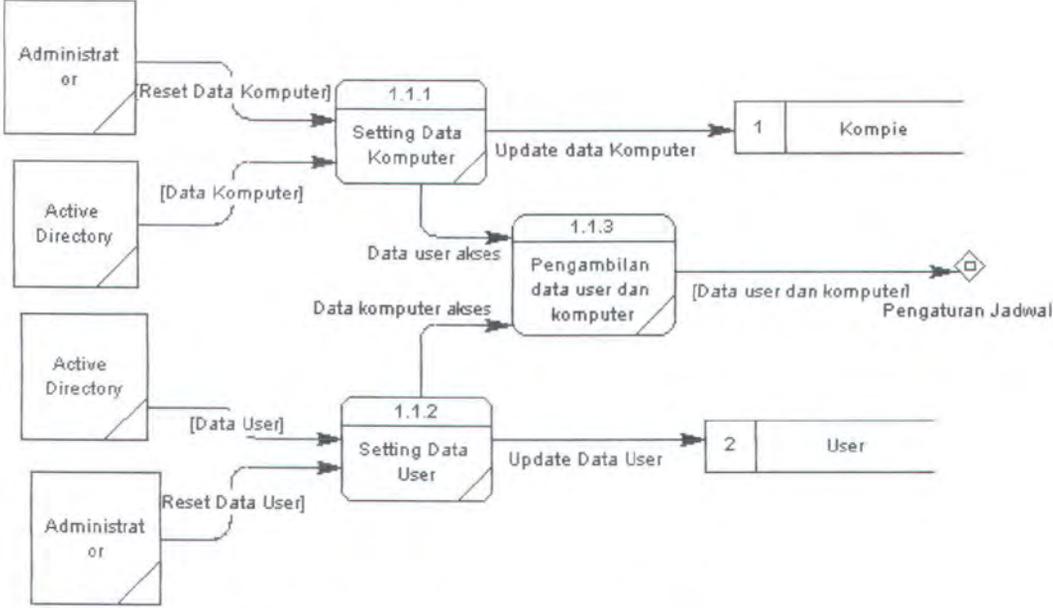
yang akan digunakan untuk memperbarui data komputer apabila dalam Active Directory terdapat komputer baru atau penghapusan komputer yang telah ada. Begitu juga apabila administrator memberikan *input* Reset User maka proses akan membutuhkan data user dari Active Directory untuk memperbarui data user pada basis data apabila ada user baru ataupun penghapusan user. Hasil dari proses setting basis data adalah data user dan komputer yang akan digunakan dalam proses berikutnya.



**Gambar 3.2** DFD level 1

Proses selanjutnya adalah proses pengaturan jadwal. Proses ini untuk melakukan *update* terhadap jadwal, apabila ada user yang memberikan *input* untuk memesan komputer. Proses ini juga untuk melakukan *reset* terhadap jadwal apabila administrator memberikan *input* Reset Jadwal, maka seluruh jadwal akan menjadi kosong. Proses ini akan menghasilkan dua buah output yang akan digunakan Active Directory untuk melakukan *update* terhadap *logon hours* dan *workstation* yang dimiliki oleh user tertentu.

Pada gambar 3.3 menggambarkan subproses dari setting basis data, proses ini memiliki beberapa subproses.



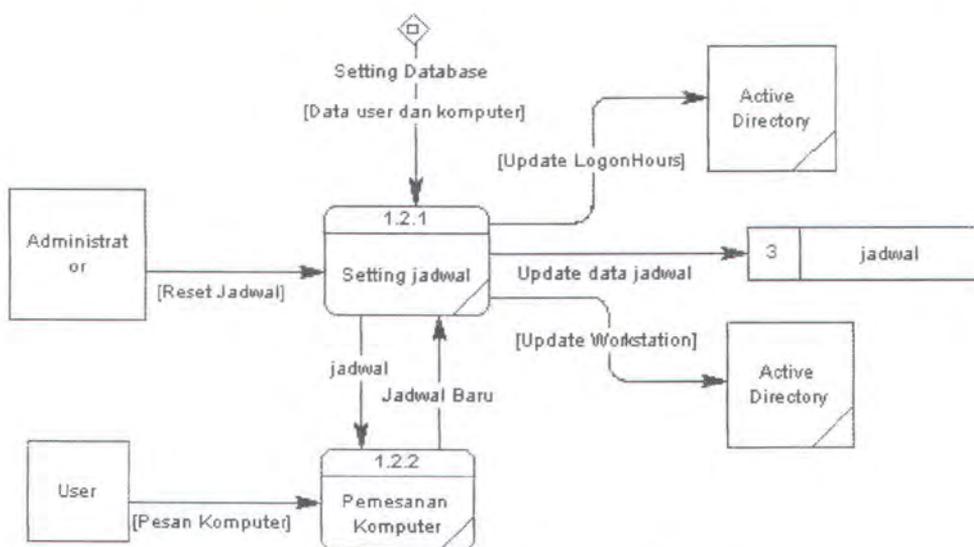
Gambar 3.3 Sub-Proses Pengesetan Data Base

Proses setting data komputer yang akan berjalan apabila seorang administrator memberikan *input* Reset Komputer maka proses membutuhkan data komputer dari Active Directory yang akan digunakan untuk memperbaiki data komputer apabila dalam Active Directory terdapat komputer baru atau penghapusan komputer yang telah ada.

Begitu juga dengan proses setting user, proses ini akan berjalan apabila administrator memberikan *input* Reset User maka proses akan membutuhkan data user dari Active Directory untuk memperbaiki data user pada basis data apabila ada user baru ataupun penghapusan user.

Selanjutnya data user dan data komputer yang sudah di *update* dalam basis data akan diambil dalam proses pengambilan data user dan komputer, yang hasilnya akan digunakan untuk proses setting jadwal.

Untuk subproses pengaturan jadwal digambarkan pada gambar 3.4. Hasil *output* dari proses setting basis data berupa data user dan komputer akan digunakan sebagai *input* dalam proses setting jadwal yang merupakan subproses pengaturan jadwal. Proses setting jadwal akan dilakukan apabila proses data komputer dan user telah didapatkan. Seorang administrator dapat melakukan *reset* jadwal dengan memberikan *input* Reset Jadwal, yang akan membuat seluruh data komputer pada jadwal dalam basis data akan menjadi kosong.



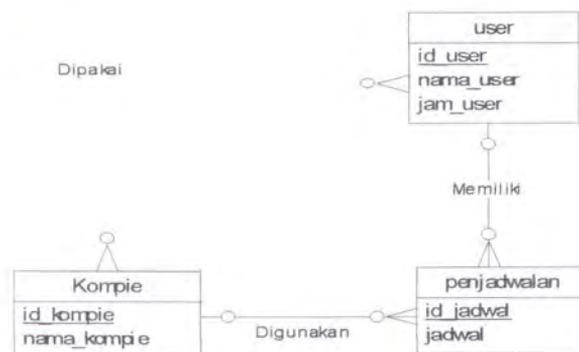
**Gambar 3.4** SubProses Pengaturan Jadwal

Sedangkan user dalam proses pengaturan jadwal berperan pada proses pemesanan komputer, apabila ada seorang user memesan komputer maka akan merubah susunan jadwal, jadwal baru tersebut akan dikirim ke proses setting jadwal yang kemudian akan menghasilkan output data *logon hours* dan

*workstation* yang dikirim ke Active Directory dan output jadwal yang akan melakukan *update* pada basis data.

### 3.2.2 Perancangan Basis Data

Untuk membuat suatu basis data yang optimal dan dapat digunakan dalam sistem yang sesuai dengan desain perangkat lunak yang telah dibuat. Gambar 3.5 merupakan Entity Relationship Diagram yang akan digunakan sebagai dasar dari pembuatan basis data.



**Gambar 3.5** Entity Relationship Diagram

Dalam ERD tersebut terdapat beberapa entiti dan masing – masing entiti mempunyai relasi dengan entiti lainnya, berikut adalah penjelasan mengenai entiti dan relasi dalam ERD pada gambar 3.5 :

#### ➤ Entity

- **Kompie**

Entity yang merepresentasikan komputer, yang berisi nama – nama komputer yang terhubung dalam jaringan yang berada dalam obyek komputer dalam *Active Directory Server*, yang telah diambil secara langsung dari Active Directory Server.

- User
 

Entity yang merepresentasikan user, yang berisi atribut `login_name` dari obyek user yang ada dalam *Active Directory Server*, yang telah diambil secara langsung dari *Active Directory Server*. Pada tabel ini terdapat atribut tambahan yang tidak ada dalam *Active Directory Server* yaitu `jam_user`. Atribut ini berfungsi memberikan keterangan sisa jam yang dapat user gunakan untuk mem-*booking* komputer.
- Penjadwalan
 

Entity yang melakukan pendataan terhadap jadwal penggunaan komputer dan jadwal user, yang diperoleh dari hasil penggabungan data dari tabel user dan komputer.

➤ Relasi

- Digunakan (Kompie – Penjadwalan | Relasi One to Many)
 

Sebuah Komputer digunakan pada 1 atau banyak jadwal, setiap jadwal terdapat keterangan hari dan jam untuk setiap 1 jam.
- Dipakai (Komputer-User | Relasi Many to Many)
 

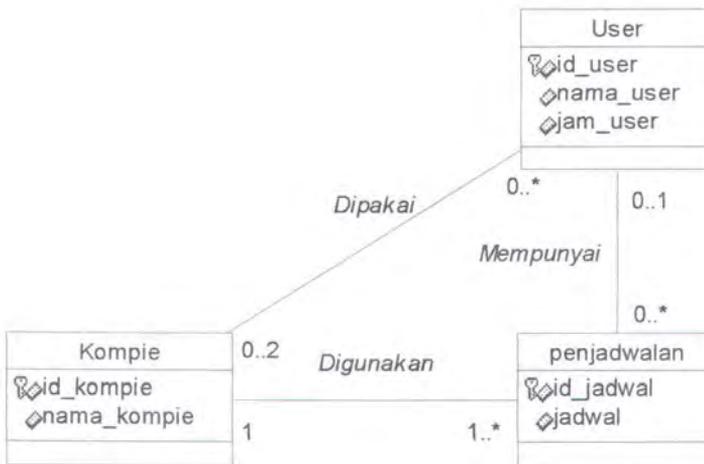
User dapat menggunakan banyak komputer, begitu juga dengan komputer dia dapat digunakan oleh banyak user. Akan tetapi, hanya administratorlah yang berhak menentukan pemakaian komputer oleh seorang user.

- Memiliki ( User – Penjadwalan | Relasi One to Many)

Sebuah user dapat memiliki 1 atau banyak jadwal atau tidak sama sekali.

### 3.2.2.1 Logical Data Model

LDM digunakan untuk menjelaskan lebih dari *Conceptual Design* dari suatu basis data atau keterangan detail dari arsitektur data. Pada gambar 3.6 merupakan *Logical Data Model* dari sistem yang akan dibuat.



Gambar 3.6 Logical Data Model

Dapat dilihat bahwa sebuah komputer digunakan untuk 1 atau lebih dari jadwal yang ada dan seorang user dapat memiliki lebih dari 1 buah jadwal yang ada atau pun tidak sama sekali, dan sebuah jadwal pasti terdapat sebuah komputer untuk digunakan dan jadwal tersebut hanya dapat digunakan oleh seorang user atau tidak sama sekali. Seorang user hanya dapat memakai 2 komputer saja, yang akan ditentukan oleh admin.

### 3.3. PERANCANGAN PROSES USER

Pada aplikasi ini terdapat dua hak akses yaitu user dan administrator. User adalah setiap pengguna laboratorium yang dapat melakukan akses terhadap komputer. Administrator adalah user yang telah diberi wewenang khusus di dalam *Active Directory Server*. Setiap hak akses memiliki perbedaan pilihan menu proses untuk menjalankan aplikasi ini. Setiap user harus melalui proses login agar dapat dibedakan antara user biasa dan administrator.

#### ➤ Hak Akses Admin

- Reset All User

Berfungsi untuk melakukan pengesetan awal atau ulang data user yang akan diambil dari data user dari *Active Directory* dan selanjutnya disimpan dalam basis data *Microsoft Access*.

- Reset All Computer

Berfungsi untuk melakukan pengesetan awal atau ulang data komputer yang terhubung dalam jaringan tersebut, yang diambil dari data komputer dari *Active Directory* dan selanjutnya akan disimpan dalam basis data *Microsoft Access*.

- Reset Schedule

Berfungsi untuk melakukan pengesetan awal atau ulang jadwal, dengan setiap komputer memiliki jadwal seperti dalam *logon hours* yang dimiliki user. Apabila admin melakukan hak ini maka seluruh jadwal akan menjadi kosong atau tidak digunakan.

- **Reset Day**

Berfungsi untuk melakukan pengesetan ulang jadwal menjadi kosong atau tidak digunakan, hanya untuk hari kemarin. Sebagai contoh jika sekarang adalah hari Senin maka apabila admin melakukan hak ini, seluruh jadwal pada hari minggu akan di kosongkan.

- **Reset User Hours**

Berfungsi untuk melakukan pengesetan ulang sisa jam yang dimiliki oleh tiap user, agar menjadi penuh kembali.

- **Memilih Komputer untuk User**

Berfungsi untuk menentukan komputer yang dapat digunakan oleh user maksimal dua buah komputer

- **Hak Akses User**

- **Melihat Jadwal yang dimilikinya**

User dapat melihat jadwal miliknya, berupa komputer yang dapat diaksesnya beserta hari dan jam, serta sisa jam yang dimilikinya

- **Memesan Komputer**

User dapat melakukan pemesanan komputer apabila komputer tersebut masih kosong atau belum ada yang memesan.

- **Melihat siapa pemakai komputer yang sudah ter-*booking***

User dapat melihat siapakah user yang telah *membooking* sebuah komputer pada jam tertentu.

Selain itu pada perangkat lunak terdapat proses peringatan apabila jam yang digunakan oleh user tinggal 5 menit.

### 3.4. PERANCANGAN ANTAR MUKA

Perancangan antar muka digunakan untuk melihat bagaimana sistem berjalan dan mempermudah pengaksesan data pada sistem. Pada aplikasi ini ada beberapa *form – form* yang berperan pada berlangsungnya sistem.

Pada aplikasi ini terdapat beberapa form utama, yaitu :

#### 1. Form Login

Bentuk dasar dari form login pada gambar 3.7

Header	
Login	
Password	
Footer	

**Gambar 3.7** Bentuk Dasar Form Login

Form ini berfungsi sebagai form utama, untuk menentukan hak akses user, apakah sebagai admin atau sebagai user biasa berdasarkan user yang melakukan login.

#### 2. Form Daftar Komputer

Bentuk dasar dari form daftar komputer pada gambar 3.8.

Header		
Nama user		
Menu	No	Nama Komputer
		nama dns
Footer		

**Gambar 3.8** Bentuk Dasar Form Daftar Komputer

Form ini berfungsi untuk menampilkan daftar komputer yang terhubung dalam jaringan *Active Directory Server*. Agar User dapat memilih komputer mana yang ia pakai dan selanjutnya dapat melihat jadwal komputer tersebut.

### 3. Form Jadwal Komputer

Bentuk dasar dari form jadwal komputer pada gambar 3.9

Header							
Nama user	<table border="1"> <tr> <td colspan="2">Nama Komputer</td> </tr> <tr> <td></td> <td>Jam</td> </tr> <tr> <td>Hari</td> <td>Jadwal</td> </tr> </table>	Nama Komputer			Jam	Hari	Jadwal
Nama Komputer							
	Jam						
Hari	Jadwal						
Menu	Daftar Komputer						
Footer							

**Gambar 3.9** Bentuk Dasar Form Jadwal Komputer

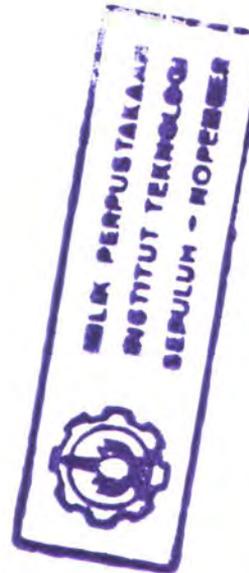
Form ini berfungsi untuk menampilkan jadwal komputer berdasarkan komputer yang dipilih user pada saat melihat form daftar komputer. Form ini menampilkan jadwal setiap komputer setiap hari dan setiap jam.

### 4. Form Pengguna

Bentuk dasar dari form pengguna komputer pada gambar 3.10

Header							
Nama user	<table border="1"> <tr> <td>Pemakai</td> <td></td> </tr> <tr> <td>Komputer</td> <td></td> </tr> <tr> <td>Jam</td> <td></td> </tr> </table>	Pemakai		Komputer		Jam	
Pemakai							
Komputer							
Jam							
Menu							
Footer							

**Gambar 3.10** Bentuk Dasar Form Pengguna



Form ini digunakan oleh untuk mengetahui siapakah yang menggunakan komputer tertentu pada jadwal tertentu, setelah user memilih jadwal yang digunakan pada form jadwal komputer.

5. Form Jadwal User

Bentuk dasar dari form Jadwal User pada gambar 3.11

Header										
Nama user										
Menu	<table border="1"> <tr> <td>Nama</td> <td></td> </tr> <tr> <td>Sisa Jam</td> <td></td> </tr> </table>		Nama		Sisa Jam					
	Nama									
	Sisa Jam									
	<table border="1"> <tr> <td>Komputer</td> <td>Hari</td> <td>Jam</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>		Komputer	Hari	Jam					
Komputer	Hari	Jam								
Footer										

Gambar 3.11 Bentuk Dasar Form Jadwal User

Form ini digunakan oleh user untuk melihat jadwal dirinya setelah melakukan pemesanan ke sejumlah komputer tertentu pada hari dan waktu tertentu.

6. Form Utama-Admin

Bentuk dasar dari form Utama Admin pada gambar 3.12

Header											
Nama user											
Menu Admin	<table border="1"> <tr> <th>No</th> <th>Nama Komputer</th> <th>nama dns</th> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>		No	Nama Komputer	nama dns						
	No	Nama Komputer	nama dns								
Footer											

Gambar 3.12 Bentuk Dasar Form Utama Admin

Form ini menampilkan menu admin dan daftar komputer, akan tetapi pada admin memilih komputer, akan dibawa pada form jadwal komputer

tetapi admin tidak dapat memilih jadwal hanya dapat melihat user yang menggunakan komputer tersebut.

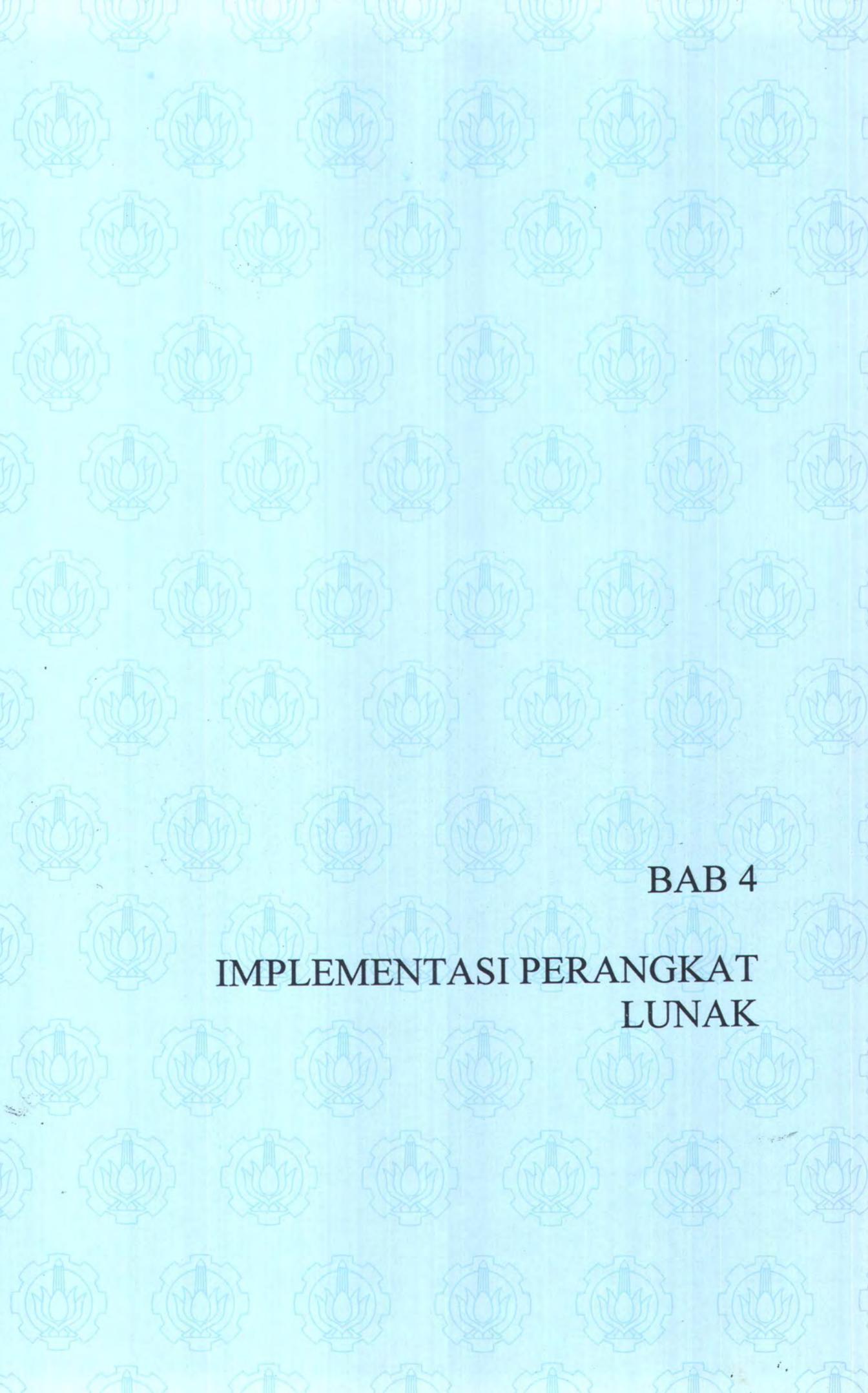
## 7. Form Daftar User-Admin

Bentuk dasar dari form Daftar User-Admin pada gambar 3.13

Header										
Nama user	<table border="1"><thead><tr><th>No</th><th>Nama User</th><th>Sisa Jam</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></tbody></table>	No	Nama User	Sisa Jam						
No		Nama User	Sisa Jam							
Menu Admin										
Footer										

**Gambar 3.13** Bentuk Dasar Form Daftar User-Admin

Form ini berguna bagi administrator untuk melihat daftar user beserta jam yang tersisa dari user. Apabila administrator melakukan transaksi proses dengan cara meng-klik salah satu user, maka sisa jam yang dimiliki oleh user yang bersangkutan akan mengalami perubahan nilai menjadi nilai awal jam yang telah diset sebelumnya.



**BAB 4**

**IMPLEMENTASI PERANGKAT  
LUNAK**

## BAB 4

### IMPLEMENTASI PERANGKAT LUNAK

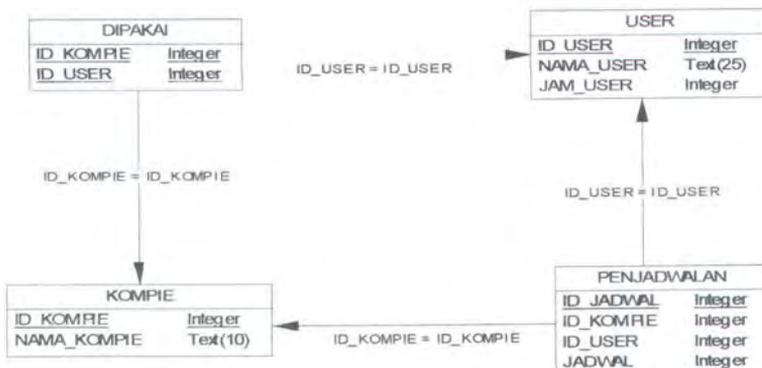
Pada pembahasan perancangan perangkat lunak telah digambarkan secara umum mengenai rancangan sistem aplikasi berbasis web. Pada bab ini akan dijelaskan implementasi dari rancangan yang telah dibuat.

#### 4.1. IMPLEMENTASI DATA

Pada implementasi data berikut ini akan dijelaskan mengenai implementasi basis data yang merupakan wujud dari perancangan data untuk membuat basis data pada Microsoft Access, dan juga akan memberikan penjelasan mengenai cara melakukan koneksi ke basis data dan juga Active Directory.

##### 4.1.1 Implementasi Basis Data

*Physical Data Model (PDM)* diperoleh dari ERD yang telah di-generate, agar sesuai dengan basis data yang akan digunakan, PDM yang dihasilkan dari generate ERD dengan menggunakan tools *Power Designer 6.0* dapat dilihat pada gambar 4.1



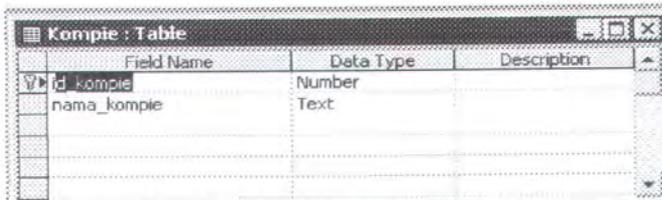
Gambar 4.1 Physical Data Model

Setelah diperoleh PDM, maka dapat dilakukan implementasi dari basis data ke dalam Microsoft Access. Berikut adalah penjelasan field – field dan tipe data yang digunakan dalam setiap tabel :

### 1. Tabel Kompie

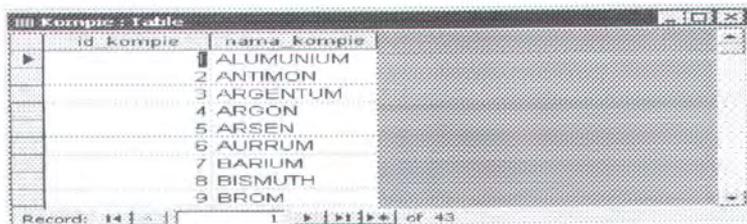
Tabel kompie berfungsi untuk menyimpan data nama\_kompie yang diambil secara langsung dari *Active Directory*.

Tabel Kompie terdiri dari dua buah atribut seperti yang dapat dilihat pada gambar 4.2, yaitu id\_kompie dan nama\_kompie. Untuk id\_kompie akan terisi secara otomatis, setelah program membaca jumlah komputer dalam *Active Directory Server*, dapat dilihat pada gambar 4.3.



Field Name	Data Type	Description
id_kompie	Number	
nama_kompie	Text	

Gambar 4.2 Tipe Data Tabel kompie



id_kompie	nama_kompie
1	ALUMUNIUM
2	ANTIMON
3	ARGENTUM
4	ARGON
5	ARSEN
6	AURRUM
7	BARIUM
8	BISMUTH
9	BROM

Gambar 4.3 Struktur Tabel Kompie

### 2. Tabel User

Tabel user berfungsi untuk menyimpan data nama\_user yang diambil secara langsung dari *Active Directory* dan juga jam\_user yang berisi sisa jam user.

Tabel User terdiri dari tiga buah atribut yaitu id\_user, nama\_user dan jam\_user, dapat dilihat pada gambar 4.4. Untuk id\_user akan terisi secara

otomatis, setelah program membaca jumlah user dalam Active Directory. Sedangkan jam\_user untuk pertama kali akan diisi secara default dengan nilai 10 untuk semua user, seperti yang terlihat pada gambar 4.5.

Field Name	Data Type	Description
id_user	Number	
nama_user	Text	
display	Text	
jam_user	Number	

Gambar 4.4 Tipe data tabel User

id_user	nama_user	display	jam_user
	Kosong		0
2 5201001	Ni Made Anni W		10
3 5201002	Deptha Bagus C		10
4 5201003	Dhany Saputra		10
5 5201004	Arif Wibisono		10
6 5201005	Yenny Kurniawati		10
7 5201006	Dudut Lesmono		10
8 5201007	Irfan Bahalwan		10

Gambar 4.5 Struktur Tabel User

### 3. Tabel Penjadwalan

Seperti yang telah dijelaskan di atas, bahwa tabel penjadwalan berfungsi untuk menyimpan data jadwal komputer dan user yang didapat secara otomatis setelah dilakukan perhitungan jumlah komputer.

Tabel penjadwalan terdiri dari empat buah atribut yaitu id\_jadwal, id\_kompie, id\_user, dan jadwal, dapat dilihat pada gambar 4.7.

Untuk id\_jadwal akan terisi secara otomatis, setelah program membaca jumlah komputer dalam *Active Directory Server*, dengan melakukan perhitungan pada gambar 4.6

Dimana jumlah komputer didapatkan dengan menghitung jumlah komputer yang ada dalam *Active Directory Server*, sedangkan total j hingga 168 adalah total jam dalam 1 minggu atau 7 hari. Sehari terdapat 24 jam sehingga  $7 \times 24 = 168$ .

```

for i=1 to jumlah_komputer
  for j=1 to 168
    k = 168*(i-1)+(j)
    id_jadwal = k
    id_kompie = i
    jadwal = 0
    id_user = 1
  Next
Next

```

**Gambar 4.6** Algoritma Untuk Tabel Penjadwalan

Sehingga id\_jadwal akan terisi dan setiap komputer akan memiliki jadwal sebanyak 168 buah. Dan untuk pertama kalinya jadwal akan terisi 0 yang menandakan komputer belum digunakan dan id\_user akan terisi 1 yang menandakan belum ada user yang memesan komputer pada jadwal tersebut, seperti yang terlihat pada gambar 4.8.

Field Name	Data Type	Description
id_jadwal	Number	
id_kompie	Number	
Jadwal	Number	
id_user	Number	

**Gambar 4.7** Tipe data tabel penjadwalan

id_jadwal	id_kompie	Jadwal	id_user
1	1	0	1
2	1	0	1
3	1	0	1
4	1	0	1
5	1	0	1
6	1	0	1
7	1	0	1
8	1	0	1
9	1	0	1

**Gambar 4.8** Struktur Tabel Penjadwalan

#### 4. Tabel Dipakai

Tabel dipakai digunakan untuk menentukan komputer yang dapat dipakai oleh user, dalam perangkat lunak ini seorang user dibatasi untuk memakai 2 buah komputer saja dan akan diisi secara manual oleh admin. Tabel dipakai hanya terdiri dari dua buah atribut saja yaitu id\_kompie dan

id\_user dapat dilihat pada gambar 4.9. Contoh isi tabel dipakai dapat dilihat pada gambar 4.10

Field Name	Data Type	Description
id_kompie	Number	
id_user	Number	

**Gambar 4.9** Tipe data tabel dipakai

id_kompie	id_user
1	2
2	2
1	1
5	5

**Gambar 4.10** Tipe data tabel dipakai

#### 4.1.2 Koneksi Basis Data

Agar sistem dapat berjalan, maka diperlukan koneksi ke database. Dalam sistem booking komputer ini, diperlukan adanya dua buah koneksi. Yang pertama adalah melakukan koneksi ke *Active Directory Server* dan yang kedua adalah koneksi ke database Microsoft Access.

##### 4.1.2.1 Koneksi Active Directory

Untuk melakukan koneksi ke *Active Directory Server* dapat digunakan Activex Data Object (ADO), dengan menggunakan bahasa pemrograman VBScript maka koneksi ke *Active Directory Server* dapat dilakukan dengan cara pada gambar 4.11

##### 4.1.2.2 Koneksi Microsoft Acces

Dengan menggunakan ADO, dapat pula dilakukan koneksi ke database Microsoft Access yang telah dibuat, seperti pada gambar 4.12

```

<%
Set Conn = Server.CreateObject("ADODB.Connection")
Set RS = Server.CreateObject("ADODB.Recordset")
Conn.Provider = "ADsDSOObject"
strConn = "Active Directory Provider"
Conn.Open strConn
%>

```

**Gambar 4.11** Koneksi Active Directory

```

<%
set conn1=Server.CreateObject("ADODB.Connection")
conn1.Provider="Microsoft.Jet.OLEDB.4.0"
conn1.Open (Server.MapPath("\ldap\database\ldap.mdb"))
%>

```

**Gambar 4.12** Koneksi Microsoft Access

## 4.2. IMPLEMENTASI PROSES USER

Pada sub bab ini akan dijelaskan mengenai implementasi dari perancangan proses user yang telah disebutkan pada bab III.

### 4.2.1 Login

Menampilkan form login untuk memberi kesempatan *user* memasukkan *username* dan *password*. Program kemudian akan mengecek validasi dari *username* dan *password* yang telah dimasukkan oleh user. Jika valid dan user memiliki hak akses sebagai user biasa maka program akan menampilkan form daftar komputer sebagai form utama dan jika user memiliki hak akses sebagai admin maka program akan menampilkan form utama admin. Jika tidak maka program akan kembali menampilkan form login untuk memberi kesempatan ulang pada *user* memasukkan *username* dan *password*.

Pada gambar 4.13 adalah *code* untuk proses login dengan mengambil data user dan password dari Active Directory.

```
<#
strADsPath = labprog.si.its-sby.edu
strPassword = user password
strUserName = user name

If strUserName Not Null Then
    strADsPath = "WinNT://" & strADsPath

    tempstr = strDomain & "\" & strUserName
    Set oADsObject = GetObject(strADsPath)

    strADsNamespace = left(strADsPath, instr(strADsPath, ":"))
    Set oADsNamespace = GetObject(strADsNamespace)
    Set oADsObject = oADsNamespace.OpenDSObject(strADsPath,
tempstr, strPassword, 0)

If Err.number Not 0 Then
    Write "Please provide a correct login name<br>or system
password for the labprog.si.its-sby.edu
    If err.number = -2147022987 Then ' for account logout
        write "Your account has been logged out"
    End If
    Else
        Session("USER_LOGIN") = strUserName
        Session("isLoggedIn") = True
        Session("ValidUser") = True
    End if
End If
#>
```

Gambar 4.13 Algoritma Login Active Directory

## 4.2.2 Proses Untuk Administrator

Seperti yang telah dijelaskan pada Bab 3, bahwa hak akses yang dimiliki oleh seorang administrator berbeda dengan oleh seorang user, berikut ini adalah proses – proses yang dapat dilakukan oleh seorang administrator.

### 4.2.2.1 Reset All User

Berfungsi untuk melakukan pengesetan awal atau ulang data user yang akan diambil dari data user dari Active Directory dan selanjutnya disimpan dalam basis data Microsoft Access.

Dalam Active Directory obyek user terdiri dari user dan komputer, setiap komputer akan diakhiri dengan tanda \$, karena itu dalam mengambil data user dalam Active Directory harus benar. Untuk mengambil user secara tepat dari Active Directory dapat dilihat pada gambar 4.14.

```
SELECT sAMAccountName FROM 'LDAP://domain' WHERE objectClass =
'user' and displayname <> 'LPSI*' and sAMAccountName <> '*$*' and
sAMAccountName <> '*NEWTON*' ORDER by sAMAccountName
```

**Gambar 4.14** Query User pada Active Directory

Setelah data terambil dari Active Directory, maka data tersebut tidak dapat langsung disimpan ke dalam basis data Microsoft Access, karena akan terjadi penumpukan data dengan id yang sama, untuk menghindari hal tersebut data dalam tabel user akan dihapus terlebih dahulu. Gambar 4.15 digunakan untuk menghapus user yang telah ada pada basis data.

```
DELETE * FROM [user] WHERE id_user <> 1
```

**Gambar 4.15** Query untuk menghapus user

Dengan kosongnya tabel user maka kita dapat memasukkan data yang telah didapatkan dari Active Directory dengan algoritma pada gambar 4.16

```
While EOF = False
  id_key=id_key+1
  username = user sAMAccountName
  INSERT INTO [user] (id_user,nama_user,jam_user) VALUES ("&
id_key &"," & username &"," & jam & ")
  Next
Wend
```

**Gambar 4.16** Algoritma dan Query Insert user

#### 4.2.2.2 Reset All Computer

Berfungsi untuk melakukan pengesetan awal atau ulang data komputer yang terhubung dalam jaringan tersebut, yang diambil dari data komputer dari Active Directory dan selanjutnya akan disimpan dalam basis data Microsoft Access.

Dalam Active Directory obyek komputer terdiri dari semua komputer yang ada termasuk juga komputer server, karena itu dalam mengambil data komputer dalam Active Directory harus benar – benar menggunakan pengkondisian yang benar sehingga hanya akan diambil data komputer yang seharusnya dapat digunakan oleh user. Untuk mendapatkan data komputer dari Active Directory secara tepat dapat dilihat pada gambar 4.17

```
SELECT name FROM 'LDAP:// domain' WHERE objectClass ='computer'  
ORDER by name
```

**Gambar 4.17** Query Komputer pada Active Directory

Sebelum data komputer yang terambil dari Active Directory disimpan ke dalam tabel kompie dalam basis data Microsoft Access, haruslah data dalam tabel kompie dihapus terlebih dahulu agar tidak jadi penumpukan data karena id yang sama dengan cara pada gambar 4.18.

```
DELETE * FROM Kompie
```

**Gambar 4.18** Query untuk menghapus komputer

Setelah data dalam tabel kompie kosong maka kita dapat memasukkan data komputer baru ke dalam tabel seperti pada gambar 4.19.

```
While EOF = FALSE  
  i=i+1  
  kompiename = computer name  
  INSERT INTO Kompie (id_kompie,nama_Kompie) VALUES ("& i  
&","& kompiename &")  
  MoveNext  
WEND
```

**Gambar 4.19** Algoritma dan Query Insert Komputer

#### 4.2.2.3 Reset Schedule

Berfungsi untuk melakukan pengesetan awal atau ulang jadwal, dengan setiap komputer memiliki bentuk penjadwalan seperti dalam *logonhours* yang



dimiliki user dalam Active Directory. Apabila admin melakukan hak ini maka seluruh jadwal akan menjadi kosong atau tidak digunakan.

Sama seperti fungsi – fungsi reset sebelumnya untuk memulai jadwal baru atau *update* maka seluruh data dalam tabel harus dihapus terlebih dahulu, dengan cara pada gambar 4.20.

```
DELETE * FROM penjadwalan
```

**Gambar 4.20** Query untuk menghapus pejadwalan

Karena jadwal dalam komputer akan disesuaikan dengan bentuk *logonhours* maka masing – masing jadwal setiap hari pada tiap jamnya akan diberi id. Untuk itu jumlah komputer akan dihitung terlebih dahulu sebelum melakukan penyimpanan jadwal ke dalam tabel. Untuk menghitung komputer dapat dilihat pada gambar 4.21.

```
select count(nama_kompie) as jumlah from kompie  
jumlah_komputer = jumlah
```

**Gambar 4.21** Query Menghitung Jumlah Komputer

Jumlah komputer yang telah didapatkan, digunakan untuk menentukan id dari jadwal, lalu akan dibuat bentuk penjadwalan yang mirip dengan *logonhours* dengan cara seperti pada gambar 4.22.

```
for i=1 to jumlah_komputer  
  for j=1 to 168  
    k = 168*(i-1)+(j)  
    INSERT INTO penjadwalan (id_jadwal,id_kompie,Jadwal, id_user)  
VALUES (" & k & ", " & i & ", 0,1)  
  Next  
Next
```

**Gambar 4.22** Query Insert Penjadwalan

#### 4.2.2.4 Reset Day

Berfungsi untuk melakukan pengesetan ulang jadwal menjadi kosong atau tidak digunakan, hanya untuk hari kemarin. Sebagai contoh jika sekarang adalah

hari Senin maka apabila admin melakukan hak ini, seluruh jadwal pada hari minggu akan di kosongkan.

Untuk reset day, jumlah komputer dihitung terlebih dahulu untuk menentukan id jadwal yang akan di-reset agar sesuai dengan hari yang akan di-reset.

Untuk menentukan hari yang akan di-reset dapat digunakan fungsi *select case (hari)*, untuk melakukan pengesetan ulang agar hari Sabtu seluruhnya menjadi 0, karena hari Sabtu terdapat pada array terakhir jadi akan dihitung mulai array 146 hingga hari Sabtu terakhir yang masuk dalam array ke 0, kode dapat dilihat pada gambar 4.23

```
case 1 :
  for i=1 to jumlah_komputer
    for j=146 to 168
      k = 168*(i-1)+(j)
      UPDATE penjadwalan SET jadwal=" & 0 & " WHERE id_jadwal = "
& k & "
    Next
  Next
```

**Gambar 4.23** Query Update Untuk Hari Sabtu

Untuk melakukan pengesetan ulang agar hari Minggu seluruhnya menjadi 0, karena hari Minggu terdapat pada array pertama jadi akan dihitung mulai array nomer 2 karena array pertama merupakan array 0 yang milik hari sabtu , array hingga hari Sabtu terakhir, kode dapat dilihat pada gambar 4.24

```
case 2 :
  for i=1 to jumlah_komputer
    for j=2 to 25
      k = 168*(i-1)+(j)
      UPDATE penjadwalan SET jadwal=" & 0 & " WHERE id_jadwal =
" & k & "
    Next
  Next
```

**Gambar 4.24** Query Update Untuk Hari Minggu

Untuk melakukan pengesetan ulang agar hari Senin seluruhnya menjadi 0, array 26 hingga array 49 merupakan milik hari Senin, kode dapat dilihat pada gambar 4.25

```
case 3 :
  for i=1 to komputer
    for j=26 to 49
      k = 168*(i-1)+(j)
      UPDATE penjadwalan SET jadwal=" & 0 & " WHERE id_jadwal =
" & k & "
    Next
  Next
```

**Gambar 4.25** Query Update Untuk Hari Senin

Untuk melakukan pengesetan ulang agar hari Selasa seluruhnya menjadi 0, array 50 hingga array 73 merupakan milik hari Selasa, kode dapat dilihat pada gambar 4.26

```
case 4 :
  for i=1 to tes
    for j=50 to 73
      k = 168*(i-1)+(j)
      UPDATE penjadwalan SET jadwal=" & 0 & " WHERE id_jadwal =
" & k & "
    Next
  Next
```

**Gambar 4.26** Query Update Untuk Hari Selasa

Untuk melakukan pengesetan ulang agar hari Rabu seluruhnya menjadi 0, array 74 hingga array 97 merupakan milik hari Rabu, kode dapat dilihat pada gambar 4.27

```
case 5 :
  for i=1 to tes
    for j=74 to 97
      k = 168*(i-1)+(j)
      sql="UPDATE penjadwalan SET jadwal=" & 0 & " WHERE id_jadwal
= " & k & ""
      conn1.Execute sql
    Next
  Next
```

**Gambar 4.27** Query Update Untuk Hari Rabu

Untuk melakukan pengesetan ulang agar hari Kamis seluruhnya menjadi 0, array 98 hingga array 121 merupakan milik hari Kamis, kode dapat dilihat pada gambar 4.28

```
case 6 :
  for i=1 to tes
    for j=98 to 121
      k = 168*(i-1)+(j)
      UPDATE penjadwalan SET jadwal=" & 0 & " WHERE id_jadwal =
" & k & "
    Next
  Next
```

**Gambar 4.28** Query Update Untuk Hari Kamis

Untuk melakukan pengesetan ulang agar hari Jumat seluruhnya menjadi 0, array 122 hingga array 145 merupakan milik hari Jumat, kode dapat dilihat pada gambar 4.29

```
case 7 :
  for i=1 to tes
    for j=122 to 145
      k = 168*(i-1)+(j)
      UPDATE penjadwalan SET jadwal=" & 0 & " WHERE id_jadwal =
" & k & "
    Next
  Next
```

**Gambar 4.29** Query Update Untuk Hari Jumat

#### 4.2.2.5 Reset User Hours

Berfungsi untuk melakukan pengesetan ulang sisa jam yang dimiliki oleh tiap user, agar menjadi penuh kembali sesuai dengan nilai default awal user. Sebelum melakukan pengesetan ulang sisa jam yang dimiliki oleh user, administrator harus memilih user mana yang akan ditambah sisa jamnya, untuk diperlukan halaman yang menampilkan daftar user yang ada beserta jumlah jam tersisa yang dimilikinya.

Setelah seorang admin memilih user yang akan ditambahkan jumlah jamnya maka skrip pada gambar 4.30 akan dijalankan :

```
UPDATE [user] SET jam_user = 10 WHERE nama_user = ''&
Request.QueryString("nama")&''
```

**Gambar 4.30** Query Update Untuk Mengganti jam user

#### 4.2.2.6 Memilih Komputer Untuk User

Setelah user dan komputer dalam tabel terisi, maka proses ini dapat dilakukan, yaitu menentukan dua buah komputer yang dapat dipakai oleh user. Apabila sudah ada komputer maka, sebuah komputer yang dipilih akan dihapus dan diganti, skrip pada gambar 4.31

```
if idwks <>0 then
    sql2 = "DELETE * FROM [dipakai] WHERE id_user ="&id&" and
id_kompie="&idwks&"
    conn1.Execute sql2
end if
    sql1="INSERT INTO [dipakai] (id_user, id_kompie) VALUES
("&id&","&komp1&)"
    conn1.Execute sql1
```

**Gambar 4.31** Untuk menghapus dan mengganti data tabel dipakai

Selanjutnya data dalam tabel dipakai diambil untuk menentukan komputer yang akan disimpan dalam *Active Directory*, pada *userWorkstation* dengan melakukan *binding* terhadap obyek user dan menyimpan dengan *.put*, seperti pada gambar 4.32

```
While RS1.EOF=False
    komp2=komp2&RS1.Fields("nama_kompie")&","
    RS1.MoveNext
Wend
Set User = GetObject
("LDAP://cn="&disp&","cn=users,dc=labprog,dc=si,dc=its-sby,dc=edu")
Response.Write(komp2)
User.Put "userWorkstations",komp2
User.SetInfo
Set user=nothing
```

**Gambar 4.32** Untuk menyimpan data pada userworkstation

### 4.2.3 Proses Untuk User

Setelah mengetahui implementasi dari proses – proses yang dapat dilakukan oleh seorang admin, berikut ini adalah proses – proses yang dapat dilakukan oleh seorang user.

#### 4.2.3.1 Memesan Komputer

Dalam proses memesan komputer terdapat beberapa proses yang berjalan diantaranya proses daftar komputer, jadwal komputer, update jadwal dan update *logonhours* dan *workstations*. Berikut adalah penjelasan dari proses – proses tersebut :

##### 4.2.3.1.1 Proses Daftar Komputer

Daftar komputer yang berada dalam Active Directory akan ditampilkan dalam form daftar komputer, agar seorang user dapat memilih komputer yang dapat dipesan. Dalam program ini, yang akan diambil adalah nama komputer dan nama dns-nya dalam server.

Seperti yang telah dijelaskan dalam BAB II bahwa dalam Active Directory tidak dapat menggunakan perintah, maka harus menggunakan nama – nama yang ada pada properti komputer. Gambar 4.33 dalah cara untuk mengambil data komputer dari Active Directory

```
SELECT name, dnsHostName FROM 'LDAP://domain ' WHERE objectClass =  
'computer' ORDER by name
```

**Gambar 4.33** Query Komputer pada Active Directory

#### 4.2.3.1.2 Jadwal Komputer

Agar user dapat mengetahui komputer – komputer yang terpakai dan yang belum diperlukan jadwal komputer, yang diambil dari basis data Microsoft Access. Untuk menampilkannya digunakan bahasa Query pada gambar 4.34

```
SELECT a.id_jadwal, a.jadwal, a.id_kompie, b.nama_kompie,
c.nama_user FROM [penjadwalan] a, [kompie] b, [user] c where
b.nama_kompie='&Request.QueryString("nama")&' and
b.id_kompie=a.id_kompie and c.id_user=a.id_user
```

Gambar 4.34 Query untuk menampilkan Jadwal Komputer

Setiap jadwal yang ditampilkan akan mengandung nilai nama komputer, nama user dan id jadwal yang nantinya digunakan sebagai *passing variable* untuk menyimpan jadwal dan melihat jadwal pengguna.

#### 4.2.3.1.3 Simpan Jadwal

Setelah user memilih jadwal yang kosong, maka nilai nama komputer, nama user dan id jadwal akan dikirim untuk disimpan dan nilai jadwal akan dirubah menjadi 1 dan jam user akan dikurangi 1. Gambar 4.35 digunakan untuk melakukan penyiman jadwal

```
UPDATE penjadwalan SET jadwal=" & 1 & ", id_user="& iduser &"
WHERE id_jadwal = " & id & "

jamuser= jamuser-1
UPDATE [user] SET jam_user =" & jamuser & " WHERE id_user = " &
iduser & "
```

Gambar 4.35 Query untuk mengupdate jadwal dan user

#### 4.2.3.1.4 Update *logonhours* dan *Workstations*

Untuk melakukan *binding* ke Active Directory agar dapat melakukan *setting* pada atribut tertentu, cara yang digunakan berbeda pada saat akan mengambil suatu atribut, cara yang digunakan pada gambar 4.36.



Bit berikutnya dan setiap 3 bit berikutnya akan ditaruh pada pilihan ke dua, seperti pada gambar 4.40.

```
Case 2,5,8,11,14,17,20:
  if i=21 Then k=0 End if
  For j=0 To 7
    dVal=2^j
    If (strByte And dVal)=dVal Then
      temp(k) = "1"
    Else
      temp(k) = "0"
    End If
```

**Gambar 4.38** Case pertama

```
Case 0,3,6,9,12,15,18:
  For j=0 To 7
    dVal=2^j
    If (strByte And dVal)=dVal Then
      temp(k) = "1"
    Else
      temp(k) = "0"
    End If
```

**Gambar 4.39** Case kedua

```
Case 1,4,7,10,13,16,19:
  For j=0 To 7
    dVal=2^j
    If (strByte And dVal)=dVal Then
      temp(k) = "1"
    Else
      temp(k) = "0"
    End If
```

**Gambar 4.40** Case terakhir

Setelah memperoleh *logonhours* yang telah disimpan dalam variabel *array* temp, maka nilai dalam *array* temp yang sesuai dengan nilai *array*, bila diperoleh 0 maka akan diubah menjadi 1. Setelah itu akan dilakukan penyimpanan ke dalam *array* yang berisi 8 nilai setiap *array*.

Selanjutnya setiap *array* yang telah berisi 8 nilai, dikonversi dalam bilangan *hexa*, dan dilakukan *stream* pada setiap *array* tadi seperti dalam gambar 4.41.

Dengan menggunakan perintah `.put "logonHours"` dengan nilai yang telah dibaca disimpan dalam variabel `LoginHours` maka nilai dapat disimpan dalam Active Directory, seperti dalam gambar 4.42.

```
temp6 = fso.gettemppname()  
ts = fso.createtextfile(temp6)  
  
for i=0 to 20  
    temp2(i) = "&h"&temp2(i)  
    ts.write chr(temp2(i))  
Next  
  
ts.close  
  
stream.open  
stream.loadfromfile temp6  
  
LoginHours = stream.read
```

**Gambar 4.41** Pembuatan file Stream untuk Logon Hours

```
User.Put "logonHours", LoginHours
```

**Gambar 4.42** Put Logon Hours

#### 4.2.3.2 Melihat Jadwal yang Dimilikinya

Seorang user dapat melihat jadwal dirinya setelah melakukan pemesanan ke sejumlah komputer tertentu pada hari dan waktu tertentu. Apabila user belum melakukan pemesanan sama sekali maka hanya akan menampilkan jumlah jam user.

User tersebut dapat melihat sisa jam yang dimilikinya, jadi user bisa lebih mengatur jam yang akan diambilnya. Selain itu user juga dapat melihat jadwal yang telah diambilnya. User tersebut dapat melihat komputer dimana dia dapat login, hari dan waktu.

Untuk menampilkan jadwal, tidak diambil langsung dari Active Directory, tetapi melalui basis data yang telah dibuat dalam Microsoft Access. Untuk mengambil data yang diperlukan digunakan Query pada gambar 4.43

```
SELECT a.nama_user, a.jam_user, b.id_jadwal,b.id_kompie,
c.nama_kompie FROM [user] a, penjadwalan b, kompie c WHERE
a.nama_user=' ' & Session("USER_LOGIN") &' ' and b.jadwal=1 and
a.id_user=b.id_user and b.id_kompie=c.id_kompie
```

**Gambar 4.43** Query untuk mengambil jadwal user

#### 4.2.3.3 Melihat Pengguna Komputer

Apabila pada jadwal komputer user memilih jadwal yang telah digunakan atau bernilai 1, maka akan ditampilkan user yang telah memesan komputer tersebut. Pengkodean yang digunakan mirip dengan jadwal user, hanya saja pada user yang diminta merupakan nilai nama user yang diperoleh pada saat memilih pada jadwal komputer.

#### 4.2.4 Proses Peringatan

Apabila user telah menggunakan komputer sesuai dengan jam yang dipesan, dan jam tersebut tinggal 5 menit maka, akan muncul sebuah pesan pada komputer yang digunakan oleh user.

Proses ini mengambil data pada tabel penjadwalan, pada kolom id\_user, sehingga dapat diketahui user yang sedang menggunakan komputer pada waktu itu. Kemudian waktu dibandingkan dengan jam pada server, sehingga apabila kurang dari 5 menit akan muncul peringatan. Implementasi kode dapat dilihat pada gambar 4.44

```
user = List1.List(temp)
If MyMinute = 55 Then
    Shell ("net send " & user & " waktu 1 jam tinggal 5 menit")
End If
```

**Gambar 4.44** Kode Untuk Proses peringatan

### 4.3. IMPLEMENTASI ANTAR MUKA

Untuk mengimplementasikan antar muka yang telah dirancang pada Bab 3, digunakan *tool Macromedia Dreamweaver MX*, berikut adalah implementasi antar muka yang telah dibuat :

#### 4.3.1 Form Login

Gambar 4.44 merupakan bentuk implementasi dari form login yang telah dirancang pada bab 3.



Gambar 4.45 Form Login

#### 4.3.2 Daftar Komputer

Pada gambar 4.45 merupakan bentuk implementasi dari rancangan pada Bab 3, terlihat dalam Active Directory terdapat 3 buah komputer yang terhubung.



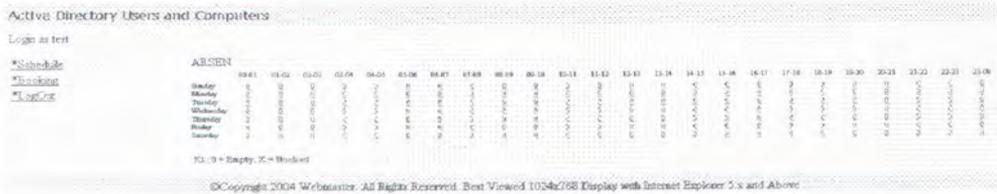
No	Nama	DNS Name
1	ALUMINUM	aluminum.labprog.sits-sby.edu
2	ANTIMON	antimon.labprog.sits-sby.edu
3	ARGENTUM	Argentum.labprog.sits-sby.edu
4	ARGON	argon.labprog.sits-sby.edu
5	ARSEN	arsen.labprog.sits-sby.edu
6	AUREUM	aureum.labprog.sits-sby.edu
7	BARIUM	barium.labprog.sits-sby.edu
8	BISMUTH	Bismuth.labprog.sits-sby.edu
9	BROM	brom.labprog.sits-sby.edu
10	CADMIUM	cadmium.labprog.sits-sby.edu

Gambar 4.46 Form Daftar Komputer

#### 4.3.3 Jadwal Komputer

Pada gambar 4.46 merupakan implementasi dari rancangan pada Bab 3, dalam form tersebut menampilkan jadwal komputer setiap jamnya mulai dari hari

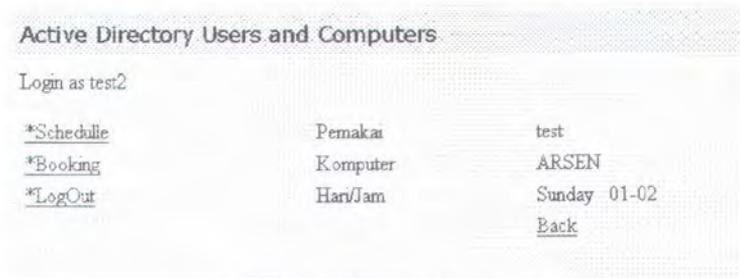
Minggu sampai Sabtu dan dari jam 00.00 hingga 23.00. Tanda X digunakan dan 0 masih kosong



Gambar 4.47 Form Jadwal Komputer

### 4.3.4 Pengguna

Pada gambar 4.47 merupakan bentuk implementasi dari rancangan pada Bab 3, terlihat nama user pemakai komputer, komputer yang dipakai dan jam terpakai.



Gambar 4.48 Form Pengguna

### 4.3.5 Jadwal User

Pada gambar 4.48 merupakan bentuk implementasi dari rancangan pada Bab 3, terlihat nama user, komputer yang dipakai, hari dan jam terpakai.



Gambar 4.49 Form Jadwal User

### 4.3.6 Utama Admin

Pada gambar 4.49 merupakan bentuk implementasi dari rancangan pada Bab 3, terlihat menu administrator yang berbeda dengan menu user biasa. Apabila admin memilih salah satu menu, proses akan dilaksanakan secara *background*. Kecuali pada menu *Reset User Hours*, yang akan membuka form baru yang akan dijelaskan selanjutnya dalam bab ini.



No	Nama	DNS Name
1	ALFREDINDRA	alfredindra.labprog.sits-sby.edu
2	ANTHONY	antonio.labprog.sits-sby.edu
3	ARGENTUM	Argentum.labprog.sits-sby.edu
4	ARGENTUM	argen.labprog.sits-sby.edu
5	ARSEN	arsen.labprog.sits-sby.edu
6	AURUM	aurum.labprog.sits-sby.edu
7	BACCUM	baccum.labprog.sits-sby.edu
8	BISMUTH	Bismuth.labprog.sits-sby.edu
9	BROM	brom.labprog.sits-sby.edu
10	CADMIUM	cadmium.labprog.sits-sby.edu

Gambar 4.50 Form Utama Admin

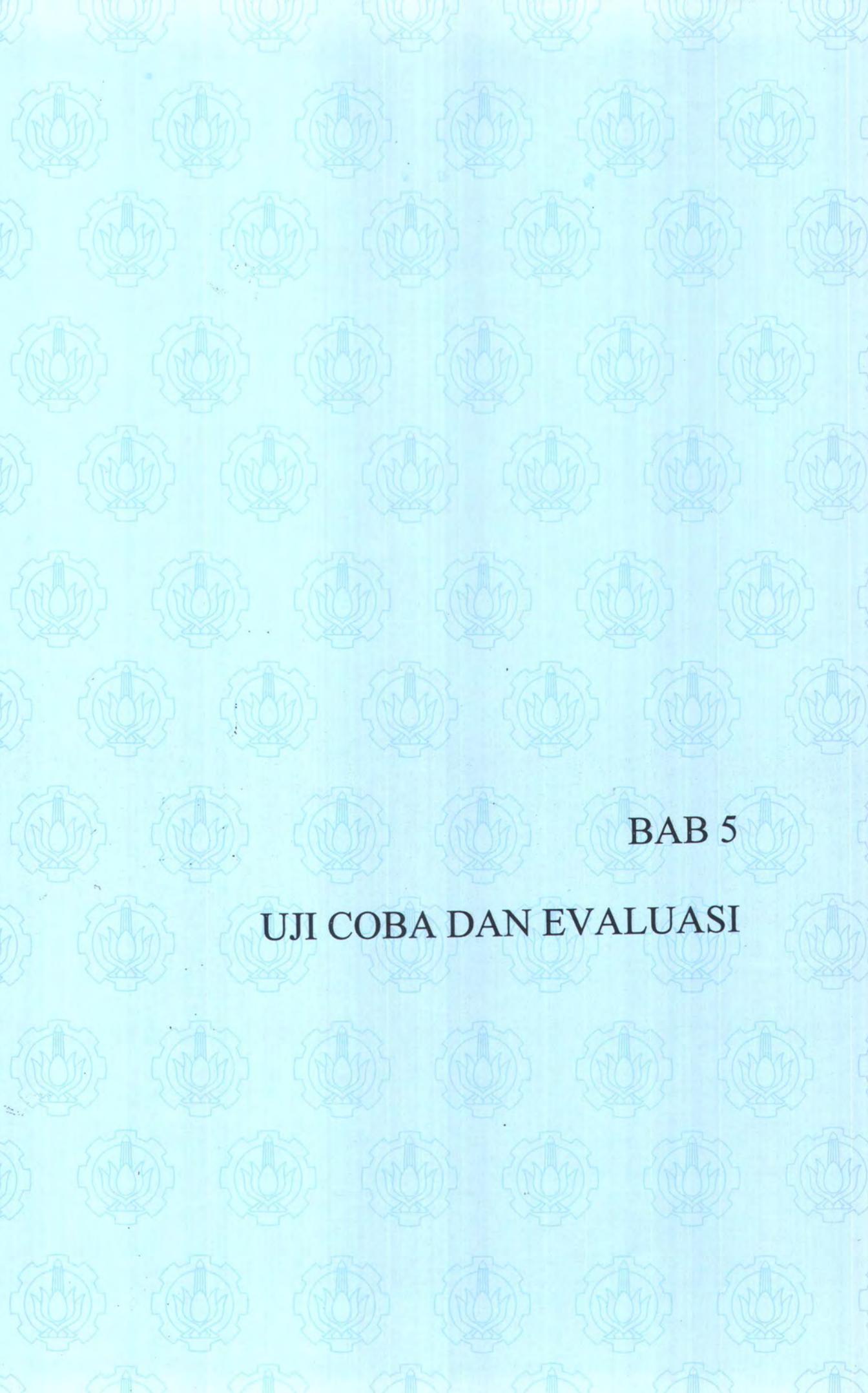
### 4.3.7 Daftar User-Admin

Pada gambar 4.50 merupakan bentuk implementasi dari rancangan pada Bab 3, terlihat nama user dan sisa dari jam.



No	Nama	Sisa Jam
1	5201801	10
2	5201802	10
3	5201803	10
4	5201804	10
5	5201805	10
6	5201806	10
7	5201807	10
8	5201808	10
9	5201809	10
...	...	...

Gambar 4.51 Form Daftar User Admin



**BAB 5**

**UJI COBA DAN EVALUASI**

## BAB 5

### UJI COBA DAN EVALUASI

#### 5.1. LINGKUNGAN PELAKSANAAN UJI COBA

Pada sub bab ini akan dijelaskan mengenai lingkungan uji coba aplikasi yang meliputi lingkungan *server* dan lingkungan *client* yang digunakan. Spesifikasi komputer *server* dapat dilihat pada tabel 5.1 dan komputer *client* dapat dilihat pada tabel 5.2.

Tabel 5.1 Lingkungan Pengujian Aplikasi pada Server

Prosesor	Sistem Operasi	Internet Information Service
Intel Pentium IV 2.40GHz	Windows Server 2003	IIS Versi 6.0

Tabel 5.2 Lingkungan Pengujian Aplikasi pada Client

Prosesor	Sistem Operasi	Aplikasi Browser
Intel Pentium IV 2.40GHz	Windows XP	Opera Browser 7.0

Uji coba dilakukan pada Laboratorium Pemrograman Jurusan Sistem Informasi.

#### 5.2. DATA UJI COBA

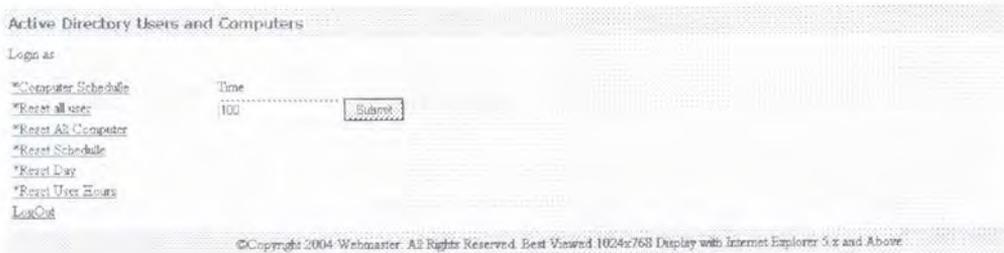
Uji coba akan dilakukan dengan menggunakan 3 orang user, yaitu 2 orang user biasa dan 1 orang administrator.

##### 5.2.1 Uji Coba Administrator Pertama

Uji coba pertama dilakukan dengan menggunakan login administrator yaitu **sinan00**. Pertama kali akan dilakukan operasi penyimpanan user, komputer dan pembuatan jadwal ke basis data.

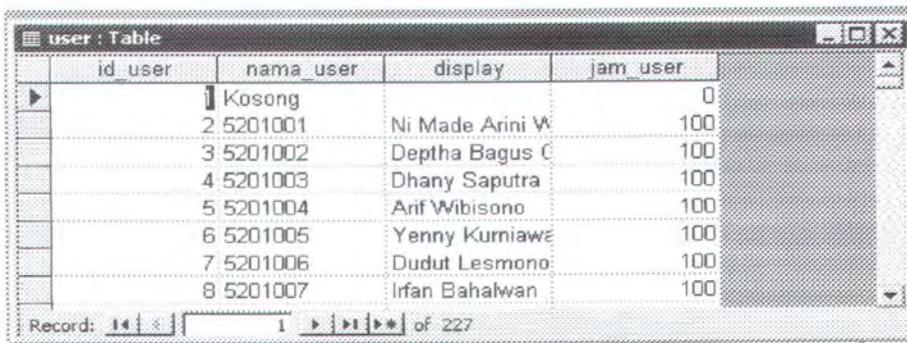
### 5.2.1.1 Reset All User

Jika administrator memilih proses ini maka akan disuruh untuk memasukkan nilai *default* dari jam yang diberikan kepada user, contoh pada gambar 5.1, nilai default 100.



Gambar 5.1 Permintaan nilai default jam

Saat basis data user masih kosong maka proses ini akan langsung mengambil properti user dari Active Directory untuk disimpan ke dalam tabel user. Hasil dapat dilihat pada gambar 5.2.



id_user	nama_user	display	jam_user
	Kosong		0
2 5201001	Ni Made Arini W		100
3 5201002	Deptha Bagus C		100
4 5201003	Dhany Saputra		100
5 5201004	Arif Wibisono		100
6 5201005	Yenny Kurniaw		100
7 5201006	Dudut Lesmono		100
8 5201007	Irfan Bahalwan		100

Record: 1 of 227

Gambar 5.2 Hasil tabel user

### 5.2.1.2 Reset All Computer

Jika administrator memilih proses ini maka proses ini akan langsung mengambil properti user dari Active Directory untuk disimpan ke dalam tabel kompie. Hasil dapat dilihat pada gambar 5.3

id_kompie	nama_kompie
1	ALUMINIUM
2	ANTIMON
3	ARGENTUM
4	ARGON
5	ARSEN
6	AURUM
7	BARIIUM
8	BISMUTH
9	BROM

Gambar 5.3 Hasil tabel kompie

### 5.2.1.3 Reset Schedule

Jika administrator maka proses ini akan melakukan proses perhitungan pada bab 4, dan akan mengisi tabel penjadwalan, hasil pada gambar 5.4. Serta akan mengosongkan seluruh properti *logonhours* pada obyek user di Active Directory, gambar 5.5 menunjukkan *logonhours* yang masih kosong.

id_jadwal	id_kompie	Jadwal	id_user
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0

Gambar 5.4 Hasil tabel penjadwalan

Logon Hours for test

12 • 2 • 4 • 6 • 8 • 10 • 12 • 2 • 4 • 6 • 8 • 10 • 12

All

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday through Saturday from 12:00 AM to 12:00 AM

Logon Permitted

Logon Denied

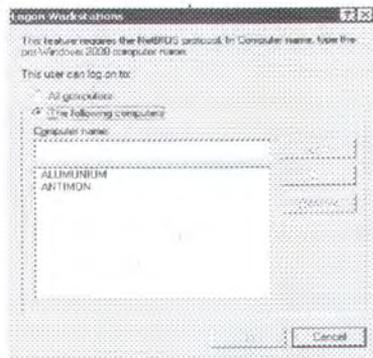
OK

Cancel

Gambar 5.5 Hasil pengosongan logonhours

### 5.2.1.4 Memilih Komputer Untuk User

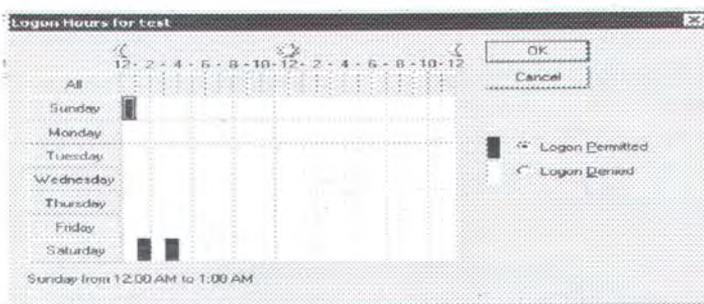
Admin menentukan komputer yang dapat digunakan oleh user, misalnya pada user test ditentukan 2 buah komputer aluminium dan antimon, maka pada Active Directory akan tersimpan 2 buah komputer tersebut, seperti pada gambar 5.6



Gambar 5.6 Komputer pada userworkstation test

### 5.2.2 Uji Coba User Pertama

User yang akan melakukan ujicoba pertama adalah user test. User akan memilih jadwal yang kosong pada form jadwal. Uji coba akan dilakukan pada komputer Aluminium pada hari Minggu pukul 12-01 dan sabtu pukul 01-02 dan pukul 03-04. Hasil pengiriman data waktu ke *logonhours* pada Active Directory dapat dilihat pada gambar 5.7.



Gambar 5.7 Hasil pengiriman waktu pada logonhours test



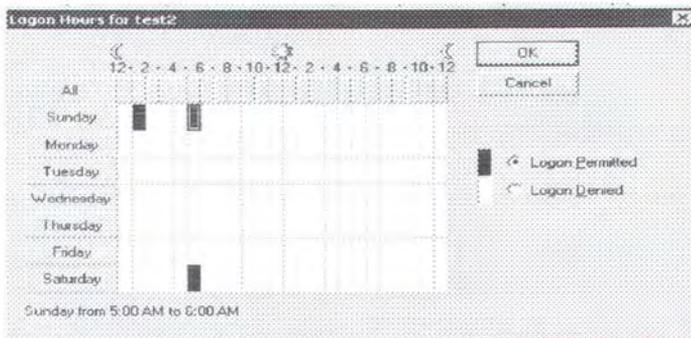
### 5.2.3 Uji Coba User Kedua

User yang akan melakukan ujicoba kedua adalah user **test2**. Pertama user akan mencoba memilih jadwal yang sudah dipilih oleh user pertama yaitu user test. Hasil yang keluar akan menampilkan form pengguna yang menunjukkan user yang telah memesan komputer pada jadwal tersebut seperti pada gambar 5.8.



Gambar 5.8 Hasil apabila user memilih jadwal user lain

Dan apabila user test2 memilih pada jadwal kosong, maka data akan tersimpan dan waktu akan terkirim pada *logonhours*, pada uji coba kali ini user akan memilih komputer pada hari Minggu pukul 01-02, dan 05-06 hari Sabtu pukul 05-06. Maka data yang terkirim pada *logonhours* dapat dilihat pada gambar 5.9



Gambar 5.9 Hasil pengiriman waktu pada logonhours test2

### 5.2.4 UjiCoba Admin Kedua

Uji coba ini dilakukan masih dengan user administrator yang sama yaitu **sinan00**, uji coba ini dilakukan untuk menghapus jadwal user tiap jam dan menambah jam user.

### 5.2.4.1 Menghapus Jadwal User

Administrator berhak membatalkan pemensanan user, hal ini dapat dilakukan dengan memilih jadwal yang telah dipilih oleh user dan apabila proses berhasil dijalankan maka jadwal tadi akan berubah menjadi 0 kembali, gambar 5.10 adalah jadwal sebelum dilakukan penghapusan dan data yang akan dihapus adalah hari Selasa jam 01-02 dan hari Minggu jam 15-16, hasil dapat dilihat pada Gambar 5.11.

	01-01	11-02	22-03	13-04	04-05	05-06	06-07	07-08	08-09	09-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	21-22	22-23	23-01
*Computer Schedule	ARGON																							
*Reset all user	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset All Computer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset Schedule	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset Day	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset User Hours	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LogOut	0: 0 = Empty, X = Booked																							

Gambar 5.10 Jadwal sebelum dihapus

	01-01	01-02	02-03	03-04	04-05	05-06	06-07	07-08	08-09	09-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	21-22	22-23	23-01
*Computer Schedule	ARGON																							
*Reset all user	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset All Computer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset Schedule	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset Day	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*Reset User Hours	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LogOut	0: 0 = Empty, X = Booked																							

Gambar 5.11 Jadwal sesudah dihapus

### 5.2.4.2 Menambah Jam User

Administrator dapat menambah jumlah jam yang dimiliki user atau pun menguranginya. Pada uji coba berikut akan dilakukan penambahan jumlah jam user yang telah berkurang dan mengurangi jam user, gambar 5.12 menunjukkan jumlah jam user test 2 yang telah mencapai 1 dan jam user test yang masih 8 akan dikurangi.

224	<u>test</u>	8
225	<u>test2</u>	1

**Gambar 5.12** Jumlah jam user sebelum dilakukan operasi

Gambar 5.13 menunjukkan jumlah yang akan menjadi jam baru pada user test2.

Active Directory Users and Computers

Login as

- \*Computer Schedule
- \*Reset all user
- \*Reset All Computer
- \*Reset Schedule
- \*Reset Day
- \*Reset User Hours
- LogOut

Nama	Jam
test2	15

Submit

**Gambar 5.13** Jumlah jam baru user test2

Gambar 5.14 menunjukkan jumlah yang akan menjadi jam baru pada user test.

Active Directory Users and Computers

Login as

- \*Computer Schedule
- \*Reset all user
- \*Reset All Computer
- \*Reset Schedule
- \*Reset Day
- \*Reset User Hours
- LogOut

Nama	Jam
test	3

Submit

**Gambar 5.14** Jumlah jam baru user test

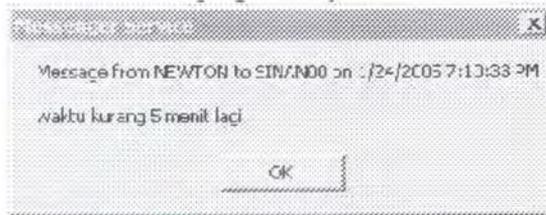
Gambar 5.15 menunjukkan hasil update jam baru untuk user test dan user test 2.

224	<u>test</u>	3
225	<u>test2</u>	15

**Gambar 5.15** Hasil Update jam user

### 5.2.5 Uji Coba Proses Peringatan

Sistem akan memberikan peringatan saat waktu kurang 5 menit, dalam uji coba dilakukan percepatan waktu pada kode. Hasil pengiriman pesan dari server kepada user dapat dilihat pada gambar 5.16



Gambar 5.16 Hasil Pengiriman pesan

### 5.2.6 Uji Force Log Off

Proses ini merupakan proses tambahan yang tidak terhubung dalam sistem secara langsung, melainkan melalui sistem keamanan dalam jaringan.

Untuk dapat melaksanakan proses ini, harus dilakukan *setting* pada *Domain Controller Security Setting* dan *Domain Security Setting*, pada *policy* yang sama, yaitu :

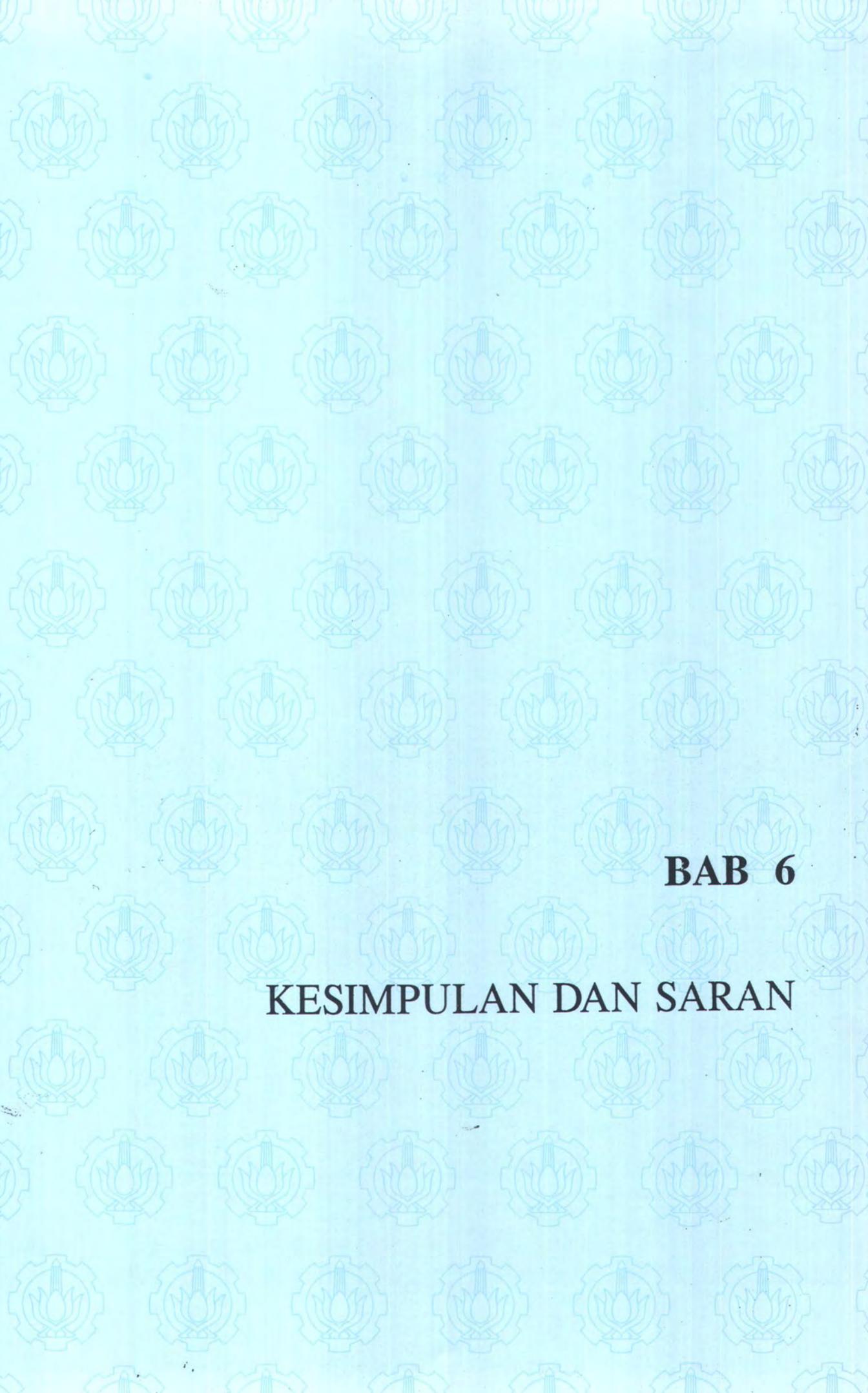
- Microsoft network server: Disconnect clients when logon hours expire
- Network security: Force logoff when logon hours expire

Yang terdapat pada tree Security Settings→Local Policies→Security Options.

Keduanya menentukan untuk memutuskan hubungan dengan user yang terhubung dengan komputer lokal diluar jam pada *logonhours* yang diperbolehkan. *Setting* ini mempengaruhi komponen Server Message Block.

Ketika *policy* pada keadaan *enabled*, menyebabkan sesi klien dengan SMB *Service* akan dipaksa untuk putus hubungan ketika masa *logonhour* yang

diperbolehkan milik klien telah habis. Dan bila *disabled*, sesi pada klien dapat terus berlangsung meskipun masa *logonhours* pada klien telah habis.



**BAB 6**

**KESIMPULAN DAN SARAN**

## BAB 6

### KESIMPULAN DAN SARAN

#### 6.1. KESIMPULAN

Setelah dilakukan serangkaian uji coba dan analisa terhadap perangkat lunak yang dibuat, maka dapat diambil kesimpulan sebagai berikut:

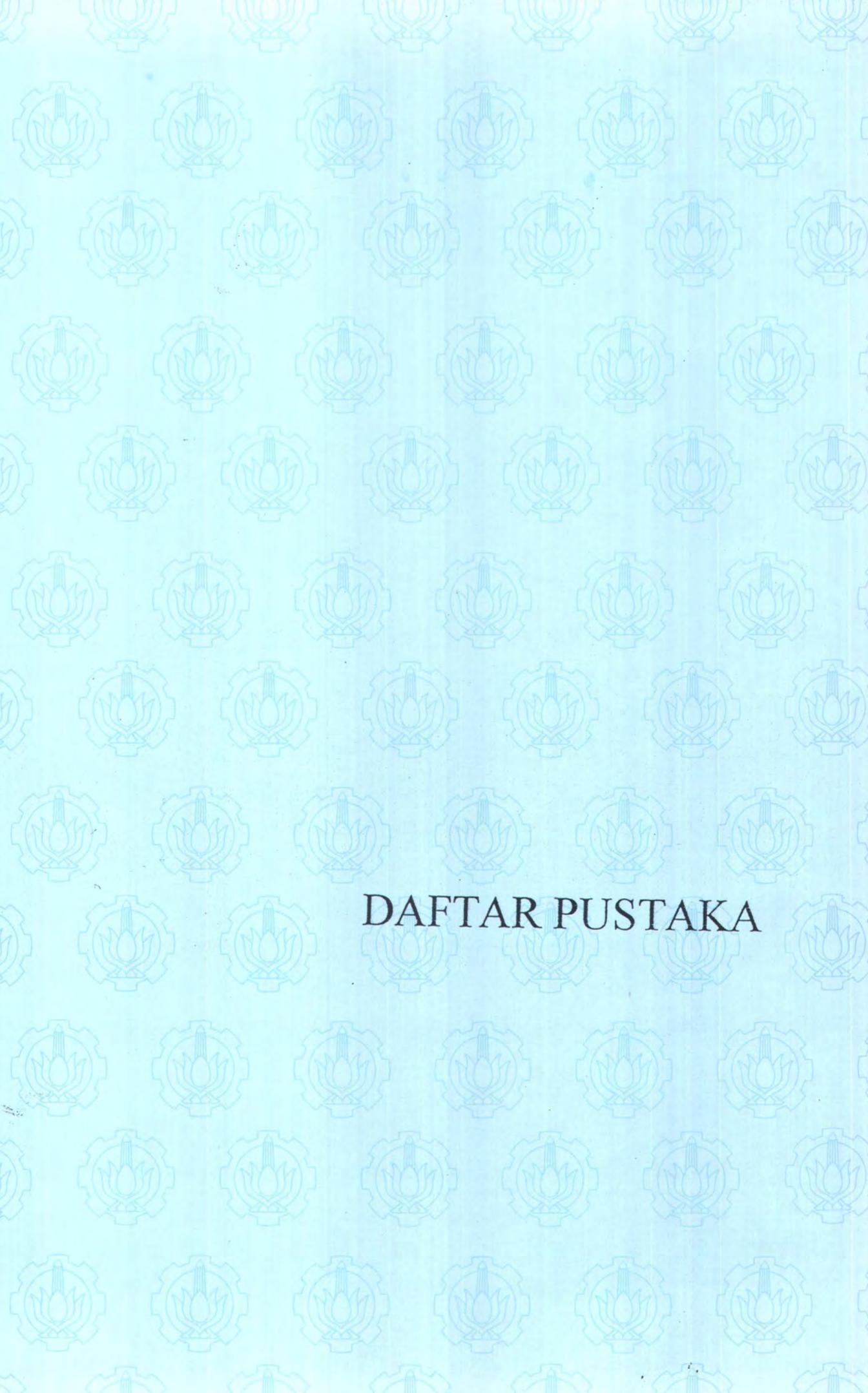
1. Untuk dapat mengakses Active Directory dapat digunakan *ActiveX Data Objects (ADO)*
2. Seluruh atribut atau properti sebuah obyek dalam Active Directory dapat diakses dengan menggunakan *Structure Query Language (SQL)*, atau dengan cara melakukan *binding* terhadap obyek tertentu, sehingga memudahkan untuk mengambil atribut user *loginhours* dan *workstation* untuk mempermudah mengatur jadwal user.
3. Active Directory dapat diakses dengan menggunakan aplikasi web, sehingga akan lebih memudahkan seorang user untuk memesan komputer dari luar kampus, sehingga penjadwalan dapat lebih mudah dilakukan.

#### 6.2. SARAN

Perangkat Lunak pada Tugas Akhir ini hanya melakukan pengaksesan ke Active Directory pada obyek user dan obyek komputer tidak melakukan penambahan atau *update* data yang telah ada, *update* yang dilakukan hanya pada *logonhours* dan *workstation* saja.

Beberapa kemungkinan pengembangan lebih lanjut dari perangkat lunak ini adalah sebagai berikut :

1. Agar perangkat lunak dapat digunakan secara maksimal oleh administrator dapat ditambahkan fungsi – fungsi untuk melakukan penambahan dan *update* data dari obyek – obyek yang ada, sehingga administrator dapat melakukan *management* terhadap Active Directory dari web.
2. Dapat ditambahkan fungsi yang dapat membedakan struktur hirarki dari user berdasarkan perbedaan dari mahasiswa dan mahasiswa TA, sehingga mahasiswa TA dapat memilih jadwal lebih leluasa.
3. Dapat ditambahkan sebuah fungsi untuk saling berkiriman pesan, apabila ada user yang ingin bertukar jadwal dengan user yang lainnya.
4. Agar user dapat melakukan *update* data pribadi yang dimilikinya dapat ditambahkan fungsi – fungsi baik dalam bentuk *SQL* atau tidak, untuk mengakses dan melakukan *update* pada atribut yang dikehendaki



DAFTAR PUSTAKA

## DAFTAR PUSTAKA

- [1] MSDN Library, Oktober 2004
- [2] [ADSIANDDirectoryServices@yahoo.com](mailto:ADSIANDDirectoryServices@yahoo.com)
- [3] Page content maintained by [itsweb@its.caltech.edu](mailto:itsweb@its.caltech.edu), *Active Directory Services*. 11/11/04  
<http://www.its.caltech.edu/win/ad.html>
- [4] *Active Directory Collection*  
<http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/techref/en-us/Default.asp>
- [5] Amri, M. Choirul. *Cepat Mahir Windows 2000 server*, Kuliah Berseri IlmuKomputer.Com Copyright © 2003 IlmuKomputer.Com  
<http://ilmukomputer.com/berseri/choirul-win2000server/index.php>
- [6] Bolte, Remie. *Add to Your ADSI Code Library*  
<http://www.15seconds.com/issue/020130.htm>
- [7] O'Brien, Bill. *What Is Active Directory?*, January 31, 2000  
[http://www.zdnet.com/zdhelp/stories/main/0\\_5594\\_2430231\\_00.html](http://www.zdnet.com/zdhelp/stories/main/0_5594_2430231_00.html)
- [8] Home, Jon Loomes. *Using ADSI and VBScript to Access Objects in NT4 and the Active Director*, March 20, 2000  
<http://www.serverwatch.com/tutorials/article.php/1548191>
- [9] Kouti, Sakari. *Inside Active Director*, 2002, Addison-Wesley  
<http://www.kouti.com/pdf/CH11-05.pdf>
- [10] <http://dev.coadmin.dk-/Resources/ADSI SDK5HTML/samples/ActiveDir/User/vc//iasuser.cpp>
- [11] *Scripts to manage Active Directory Users Using VBScript*  
<http://www.activexperts.com/activmonitor/windowsmanagement/adminscripts/usersgroups/users/>
- [12] Policht, Marcin, *Accessing Active Directory via Active Server Pages (Part 1)*, September 23, 2002  
<http://www.serverwatch.com/tutorials/article.php/1476961>

- [13] Seitsonen, Mika and Kouti, Sakari. *Managing Active Directory OUs, Users, and Groups*, Dec 3, 2004, Addison Wesley Professional  
<http://www.informit.com/articles/article.asp?p=352986>
- [14] Cartwright, Adam S. Using ADSI in your ASP Applications  
[www.topxml.com/conference/wrox/1999\\_dc/Powerpoints/adamadsi.ppt](http://www.topxml.com/conference/wrox/1999_dc/Powerpoints/adamadsi.ppt)
- [15] Harris, Michael. *User.LoginHours*  
[http://groups-beta.google.com/group/microsoft.public.scripting.wsh/browse\\_thread/thread](http://groups-beta.google.com/group/microsoft.public.scripting.wsh/browse_thread/thread)
- [16] Julian, *Querying an LDAP Server using Active Directory and ADO*  
<http://www.4guysfromrolla.com/webtech/041800-1.shtml>