

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pemantauan dan pengendalian komputer secara terpusat di Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta dapat dilakukan dengan memanfaatkan *Remote Instalation Services* (RIS) dan membuat suatu perangkat lunak baru (*Ancilla 1.0*) yang menggunakan protokol UDP. UDP dipilih karena sangat cocok digunakan dalam transfer data yang bersifat *one-to-many*. Pengembangan perangkat lunak terdiri dari 2 bagian yaitu : di sisi *server* yang meminta layanan/memberi perintah dan di sisi *client* yang menjalankan perintah.

Berbagai teknologi dapat dimanfaatkan dalam mendukung sistem pemantauan dan pengendalian komputer secara terpusat. Pemantauan sumber daya komputer dapat dilakukan dengan memanfaatkan WMI. Untuk menghidupkan komputer dapat memanfaatkan teknologi *Wake On LAN* (WOL). Untuk melakukan instalasi sistem operasi dapat memanfaatkan RIS. Sistem yang dikembangkan telah diuji dan dapat dikatakan berhasil dikembangkan dan dapat diterapkan di laboratorium tersebut.

5.2 Saran

Sistem pemantauan dan pengendalian komputer secara terpusat masih perlu dikembangkan lebih lanjut. Pengembangan dapat berupa penambahan fungsionalitas seperti mengendalikan komputer dari komputer *server* sebagaimana secara fisik berada di depan *client*.

DAFTAR PUSTAKA

Adnan, Risman, 2007, *Windows Networking dengan UDP dan .NET*, www.microsoft.com/indonesia/msdn.

Agung, RJB. Wahyu, Henri S.V. Simanjuntak, Kusworo Anindito, 2002, *Diktat Jaringan Komputer*, Universitas Atma Jaya Yogyakarta, Yogyakarta.

Ardiansyah, Dian, 2004, *Teknologi Jaringan Komputer*, <http://www.ilmukomputer.com>.

Hasan, Hardi, 2005, *Sistem Pengendalian dan Pemantauan Komputer Jarak Jauh dengan Menggunakan Winsock Control (Skripsi)*, Universitas Atma Jaya Yogyakarta, Yogyakarta.

Kurniawan, Agus, 2005, *Pemrograman Dasar Socket dengan C#*, Project Otak-<http://otak.csharpindonesia.net>.

Lieberman Software Corporation, 2006, *White Paper: Wake on LAN Technology*, www.liebsoft.com.

Microsoft, 2005, *MSDN Library - Visual Studio 2005*, Microsoft Corporation.

Nugroho, Antonius Sunarmanto, 2006, *Pengembangan Aplikasi Monitoring Koneksi Jaringan Berbasis TCP/IP (Skripsi)*, Universitas Atma Jaya Yogyakarta, Yogyakarta.

Prihanto, Harry, 2003. *Membangun Jaringan Komputer: Mengenal Hardware dan Topologi Jaringan*, IlmuKomputer.Com.

Purbo, Onno W., dkk, 1998, *Buku Pintar Internet TCP/IP*, PT Elex Media Komputindo, Jakarta.

Srivastava, Amitabh, Edward Jezierski, 2001, *Monitoring in .NET Distributed Application Design*, Microsoft Corporation.

Stalling, William, 2001, *Komunikasi Data dan Komputer : Dasar-Dasar Komunikasi Data -Edisi Bahasa Indonesia, Judul Asli: Data & Computer Communications, 6th Edition-*, Salemba Teknika, Jakarta.

Tanenbaum, Andrew S., 2000, *Jaringan Komputer - Ed.3 (Revisi), Bahasa Indonesia, Judul asli: Computer Networks 3e-*, Prenhalindo, Jakarta.

Utdirartatmo, Firrar, 2003, *"Remote Access dengan VNC dan Windows Terminal Services di Linux dan Windows"*, Penerbit ANDI, Yogyakarta.

_____. <http://id.wikipedia.org>, diakses pada bulan Januari 2007.

_____. <http://www.microsoft.com>, diakses pada bulan Februari 2007.



Apendiks :

Remote Installation Services (RIS)

Pengantar

Remote Installation Services (RIS) didesain untuk mendistribusikan sistem operasi yang termasuk dalam *Microsoft Windows Platform*. RIS merupakan *windows component* pada *Windows Server 2003*.

Spesifikasi kebutuhan RIS

Untuk menggunakan RIS terdapat spesifikasi kebutuhan sebagai berikut :

1. *User*

User dalam menjalankan RIS harus memiliki hak akses sebagai administrator serta memiliki kemampuan teknis dalam pengelolaan jaringan.

2. *Server*

Server adalah komputer yang berperan sebagai penyedia layanan RIS. *Server* memiliki spesifikasi kebutuhan antara lain :

- a. *Server* memiliki kemampuan menjalankan *Windows Server 2003 Standard* atau *Enterprise Edition*.
- b. *Component RIS* sudah diaktifkan dalam sebagai *Windows Services*.
- c. *Server* memiliki partisi drive NTFS versi 5 atau terbaru dan memiliki kapasitas drive kosong 2 GB (rekomendasi) sebagai tempat dimana RIS akan di-install.
- d. *Server* atau RIS server terkonfigurasi AD (*Active Directory*), DNS (*Domain Name Server*) serta DHCP (*Dynamic Host Configuration Protocol*) dengan baik.
- e. Terdapat *images* *Windows XP* yang akan didistribusikan ke *client*.

3. *Client*

Client yang akan di-install sistem operasi *Windows XP* harus memiliki kemampuan sebagai berikut :

- a. *Client* yang akan di-*install* melalui RIS harus memiliki spesifikasi kebutuhan minimum untuk menjalankan Windows XP.
- b. *Client* harus memiliki *Network Interface Card*(NIC) yang mendukung protokol *Preboot eXecution Environment* (PXE) versi .99C atau terbaru atau NIC yang akan bekerja dengan *boot floppy* yang akan dibuat khusus dengan RIS (akan dijelaskan di selanjutnya).
- c. *Client* terkoneksi dengan *server* menggunakan protokol TCP/IP.

Proses Instalasi

Langkah-langkah dalam melakukan instalasi RIS adalah sebagai berikut :

1. *Server*

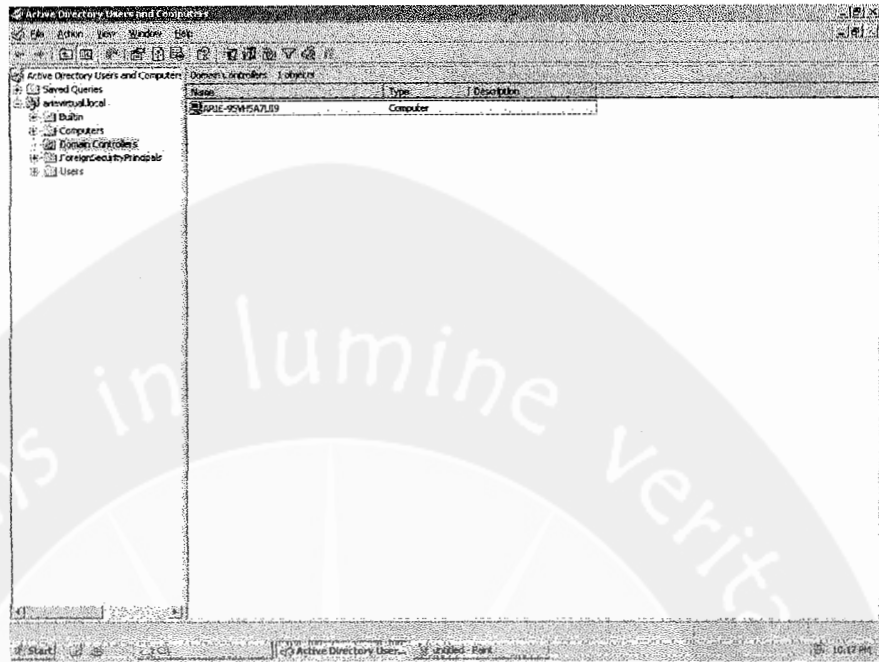
- a. Instalasi Windows Server 2003

Windows Server 2003 merupakan penyedia layanan RIS. Windows Server harus di-*install* pada *machine server*. Instalasi Windows Server 2003 tidak dijelaskan lebih lanjut.

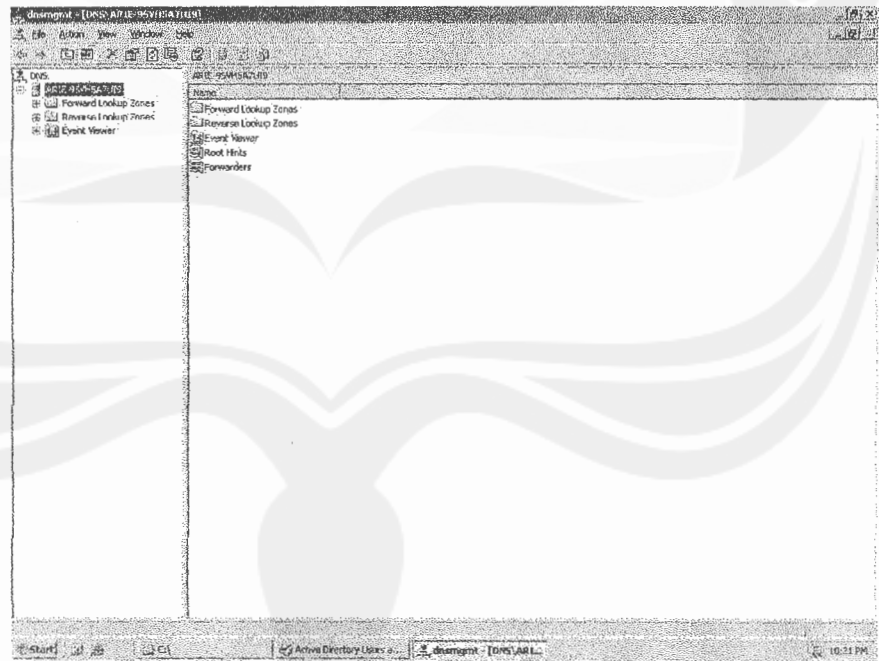
- b. Instalasi AD, DNS serta DHCP

Active Directory harus ada. RIS server harus menjadi *member server* dalam AD *forest*. Demikian pula DNS dan DHCP, kedua komponen ini harus berfungsi dan terkonfigurasi untuk mendukung operasi PXE. PXE digunakan untuk *boot network* oleh *client*. Instalasi AD, DNS dan DHCP tidak dijelaskan lebih lanjut.

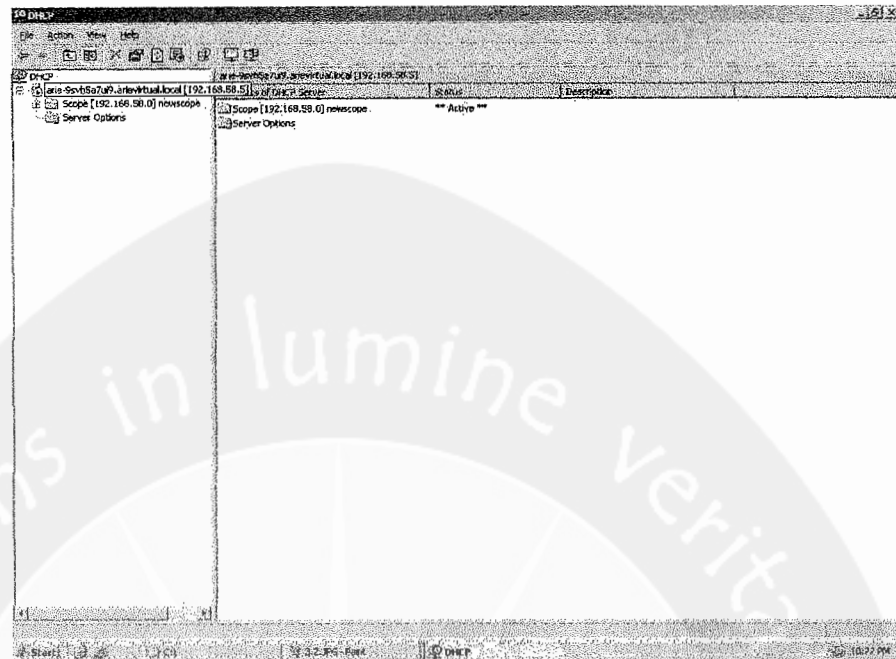
Contoh *form* AD, DNS dan DHCP yang sudah terkonfigurasi dapat dilihat seperti pada gambar 1, 2 dan 3.



Gambar 1 Active Directory



Gambar 2 DNS

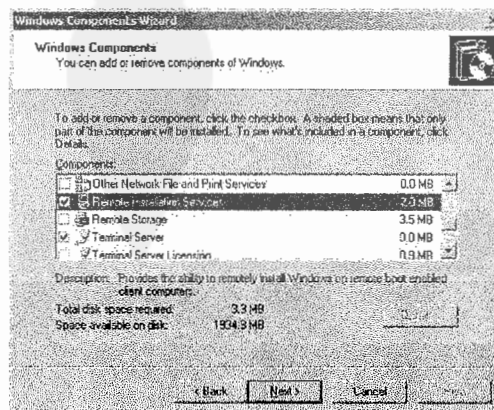


Gambar 3 DHCP

c. Instalasi RIS

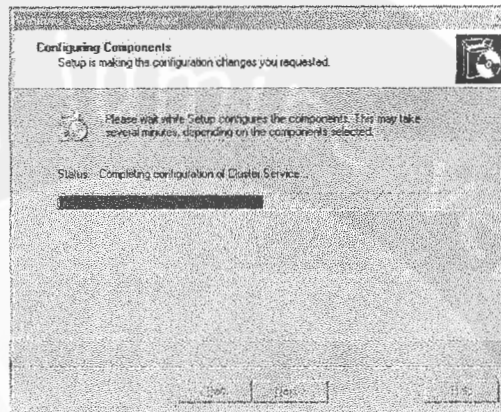
RIS tidak di-set pada konfigurasi Windows Server 2003 *Standard* maupun *Enterprise Edition* secara normal. Maka dari itu, RIS memerlukan instalasi tersendiri. RIS termasuk dalam *Windows Components Wizard* yang di-install dari menu *Add Or Remove Program*. Langkah-langkahnya sebagai berikut :

1. Buka Start Menu, pilih *Control Panel*, klik *Add Or Remove Programs*, klik *Add/Remove Windows Component*. Halaman *Windows Components Wizard* akan terbuka seperti pada gambar 4.



Gambar 4 Windows Component Wizard

2. Pilih *Remote Installation Services*, klik *Next*. Selanjutnya sistem meminta input CD installer Windows Server 2003. Kemudian, sistem akan melakukan *configuring component* layanan *Remote Installation Services* seperti pada gambar 5. Klik *Finish* ketika selesai.



Gambar 5 *Configuring Component*

3. Tutup halaman *Add Or Remove Program* dan restart komputer.
- d. *Service Setup* dan membuat *images* Windows XP.

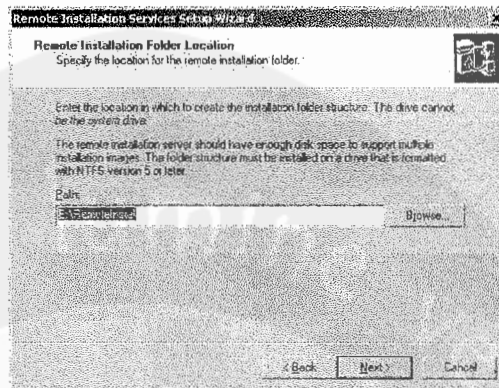
Untuk mengaktifkan server RIS, perlu dilakukan proses *setup* sebagai berikut :

1. Buka *Start Menu* >> *All Programs* >> *Administratif tools*>> *Remote Installation Service Setup*.
2. *Setup Wizard* akan menampilkan informasi *requirement* dalam RIS antara lain partisi yang dibutuhkan (*NTFS version 5*). Akan muncul form seperti pada gambar 6. Klik *Next*.



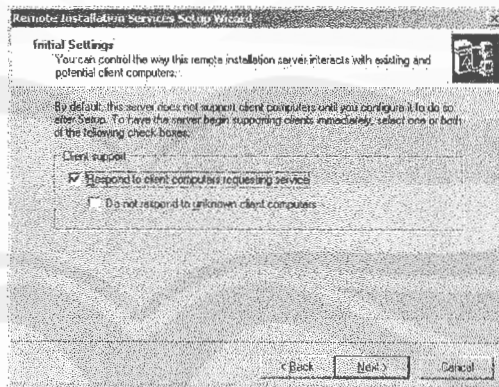
Gambar 6 *Setup Wizard*

3. Berikutnya, RIS meminta lokasi *file* sistem operasi yang akan dibuat *image file*-nya. *Browse* dan klik *next*.



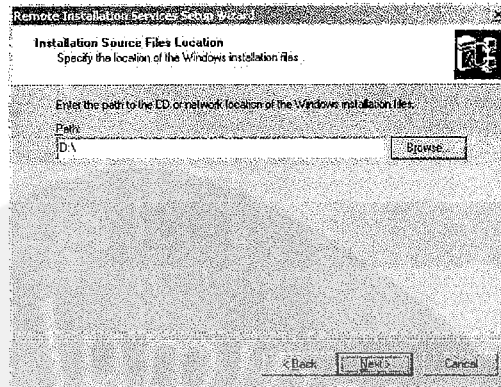
Gambar 7 Folder Location

4. Sistem akan menanyakan *control* dalam melayani *client* seperti pada gambar 7. Sistem memungkinkan adanya pemilihan *client* yang boleh dilayani. Respon untuk *client* yang dikenali diperoleh dari data GUID/UUID yang dikirim *client*.



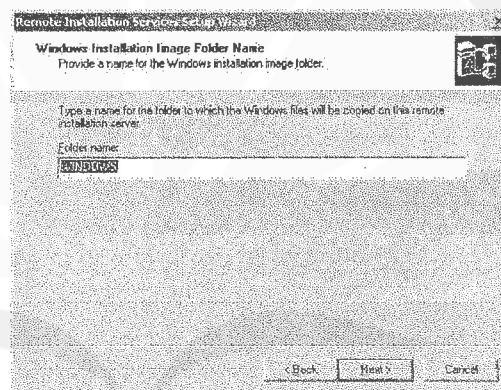
Gambar 7 Control Service

5. Berikutnya, RIS meminta lokasi *source file location* untuk *images* yang akan dibuat seperti pada gambar 8. *Browse* dan Klik *Next*.



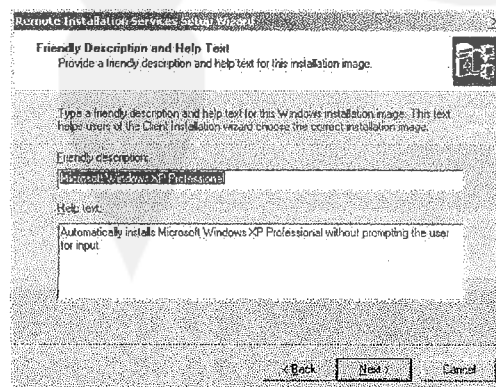
Gambar 8 Source File Location

6. Berikutnya, RIS meminta masukkan nama *folder* dimana *windows installation file* akan dibuat ("Windows" sebagai *default*) sebagaimana gambar 9. Klik *Next*.



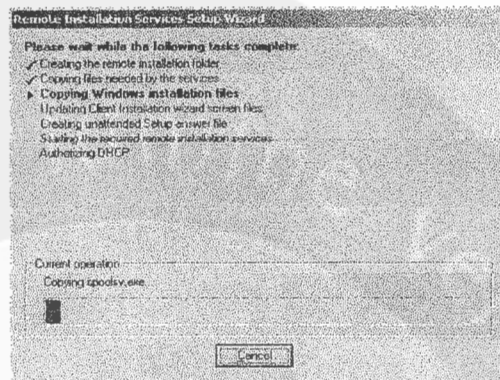
Gambar 9 Image Folder name

7. Selanjutnya tertampil *form friendly description* dan *help text* sebagaimana gambar 10. Ubah sesuai kebutuhan dan klik *Next*.



Gambar 10 Friendly Description

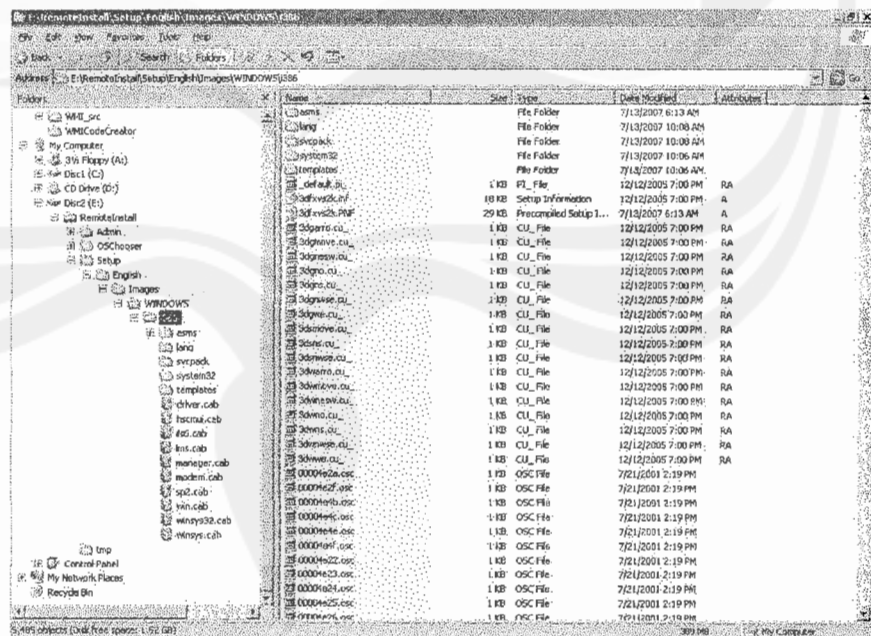
8. Proses berikutnya adalah yang paling penting. Akan muncul *form* seperti pada gambar 11. RIS akan membuat *folder*, menggandakan *file* (menjadi suatu *image*) dan membuat atau memulai konfigurasi komponen yang dibutuhkan.



Gambar 11 Apply Setup

9. Ketika RIS Setup Wizard selesai. Klik Done.

Images telah dibuat, bisa dilakukan pengecekan pada *folder* dimana lokasi *images* disimpan seperti pada gambar 12.

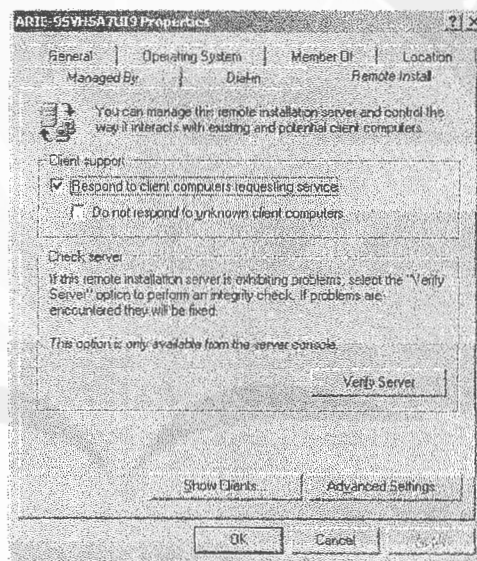


Gambar 12 Images

e. Konfigurasi RIS

Untuk memastikan bahwa RIS server dapat berfungsi dengan baik perlu dilakukan konfigurasi sebagaimana berikut :

1. Buka *Active Directory form*. Pada *Start Menu >> Administratif Tools >> Active Directory User And Computer*. Maka akan tertampil form seperti pada gambar 1 diatas.
2. Kemudian buka *Domain Controller folder* dan locate pada server. Klik kanan dan pilih *properties*.
3. Pada *properties dialog box* klik tab *Remote Install* maka akan tertampil *properties dialog box* seperti gambar 13.



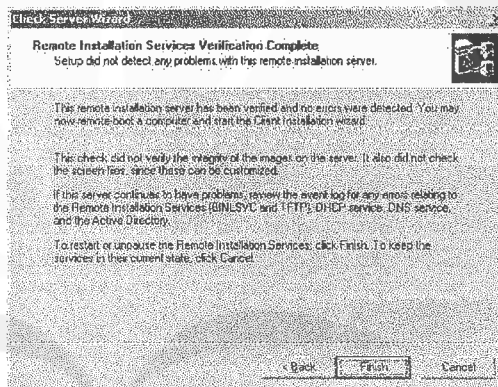
Gambar 13 *Properties Remote Install*

4. Pilih *respond to client computer requesting service* (apabila sudah di set, biarkan saja). Apabila tidak ingin merespon *client* yang tidak dikenali pilih *do not respon to unknown client computer*.
5. Untuk menu *Verify Server*.
 - a. Muncul halaman keterangan sebagaimana gambar 14.



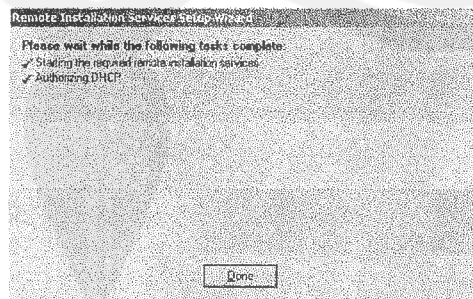
Gambar 14 Check server

- b. Muncul keterangan *report status* pengecekan sebagaimana gambar 15. Untuk melakukan restart RIS klik tombol *finish*.



Gambar 15 Report Status

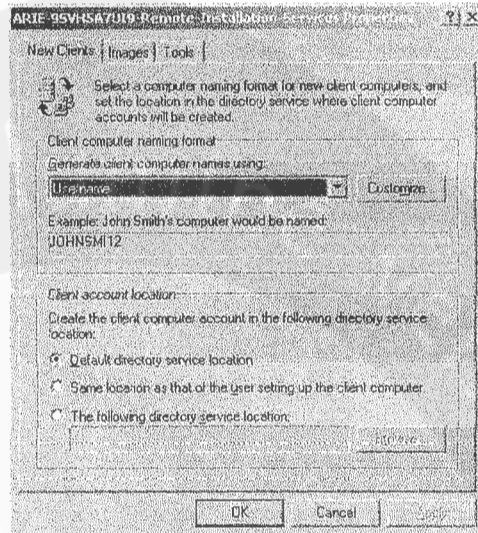
- c. RIS akan di restart sebagaimana gambar 16. Klik *done* jika selesai.



Gambar 16 Restart RIS

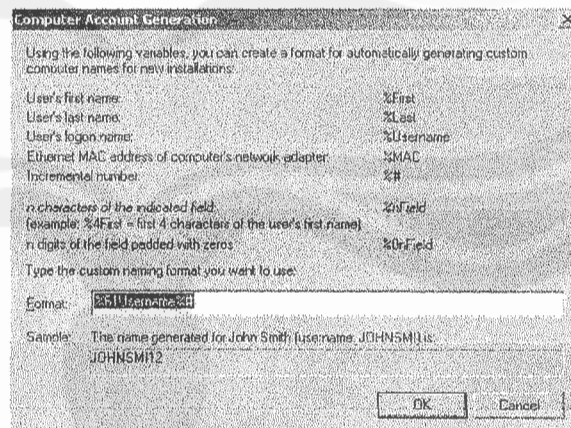
6. Untuk menu *advance setting*.

- a. Muncul *form* seperti gambar 17. *Form* ini dipakai untuk melakukan setting terhadap *generate name client* dan *setting account client location*.



Gambar 17 Advance Setting

- b. Untuk *customize name client*, muncul *form* seperti gambar 18. *Form* ini dipakai untuk melakukan setting terhadap *generate name client*.



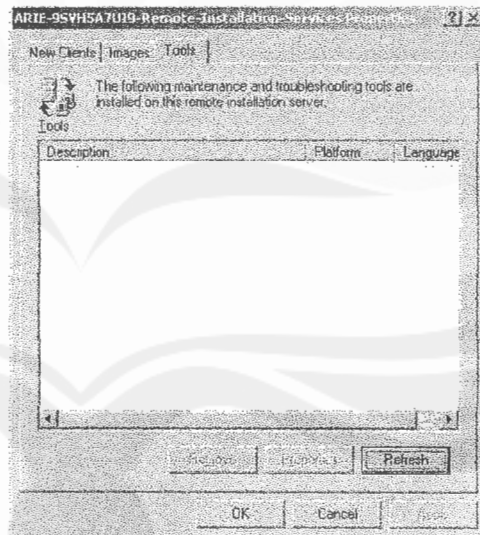
Gambar 18 Account setting

- c. Untuk konfigurasi *images*, dapat mengaktifkan *images tab* seperti pada gambar 19. *User* dapat melakukan penambahan maupun menghapus *images* yang sudah ada.



Gambar 19 Images setting

- d. Untuk konfigurasi *tools*, dapat mengaktifkan *tools* tab seperti gambar 20. User dapat melakukan perubahan konfigurasi.



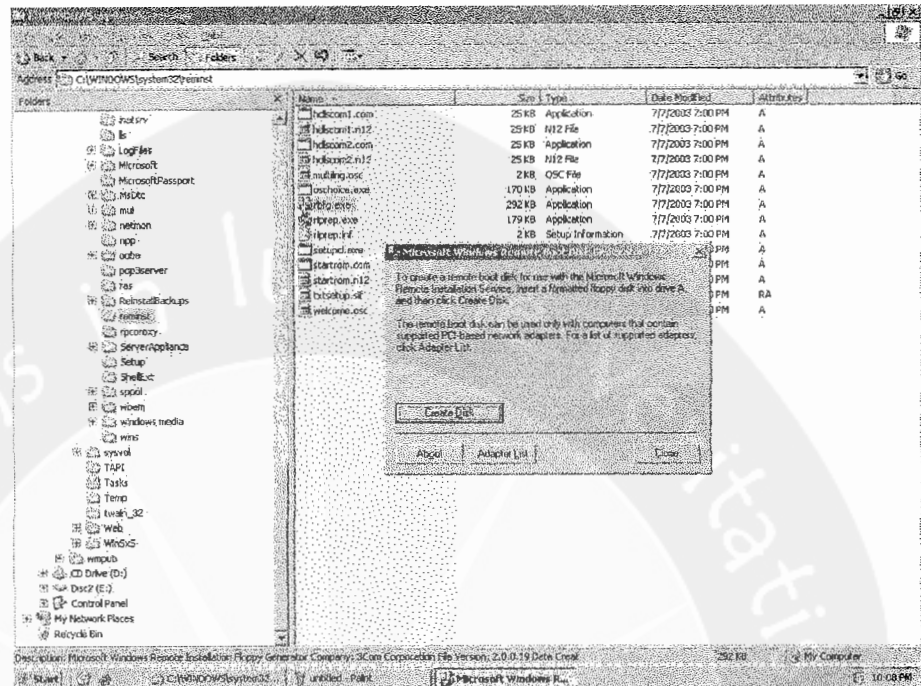
Gambar 20 Tools setting

f. Membuat *Disk Boot*

Disk boot adalah disket yang secara khusus digunakan oleh *client* untuk melakukan *request* layanan RIS ke *server*. *Disk* ini dibutuhkan jika suatu *client* tidak dapat melakukan *network boot* sebagai *first boot*. Langkah pembuatannya :

1. Buka *Windows Explorer*. Buka folder *C:\Windows\System32\RemInst* dan klik ganda pada file

Rbfg.exe. Remote Boot Disk Generator akan muncul sebagaimana gambar 21.



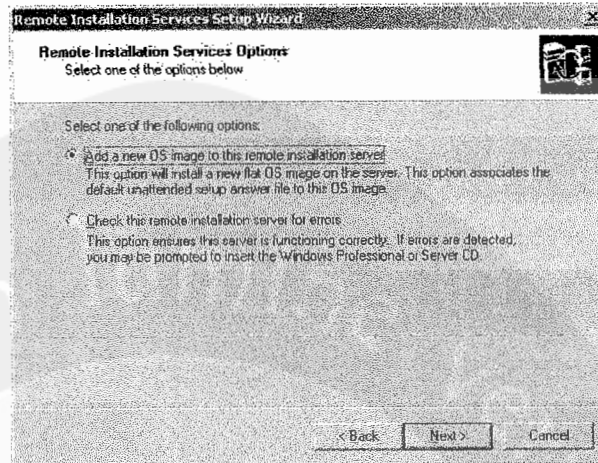
Gambar 21 Disk boot generator

2. Klik *Adapter List* dan *adapter list* yang mendukung RIS akan ditampilkan. Fasilitas ini untuk mengetahui contoh *adapter* yang *support* RIS.
3. Masukkan *floppy disk* dan klik *create disk*. *Floppy disk* akan diverifikasi dan akan dilakukan peng-copy-an data. Ketika proses selesai, sistem akan menunggu perintah akan membuat *floppy disk* baru atau tidak.
4. Ketika proses selesai tutup *form* dan keluarkan *floppy disk*.

g. Melakukan pembuatan *images* baru

1. Buka halaman *Remote Installation Service Setup* dari *start menu* >> *Administratif tools* >> *Remote Installation Services Setup*. Akan muncul form seperti gambar 22. Pilih *radiobutton* pada *Add new OS* untuk membuat *images* baru, *radiobutton* yang lain untuk melakukan pengecekan RIS. *Check server* digunakan untuk

melakukan pengecekan RIS, sudah dijelaskan diatas.
Klik next.



Gambar 22 Service Option

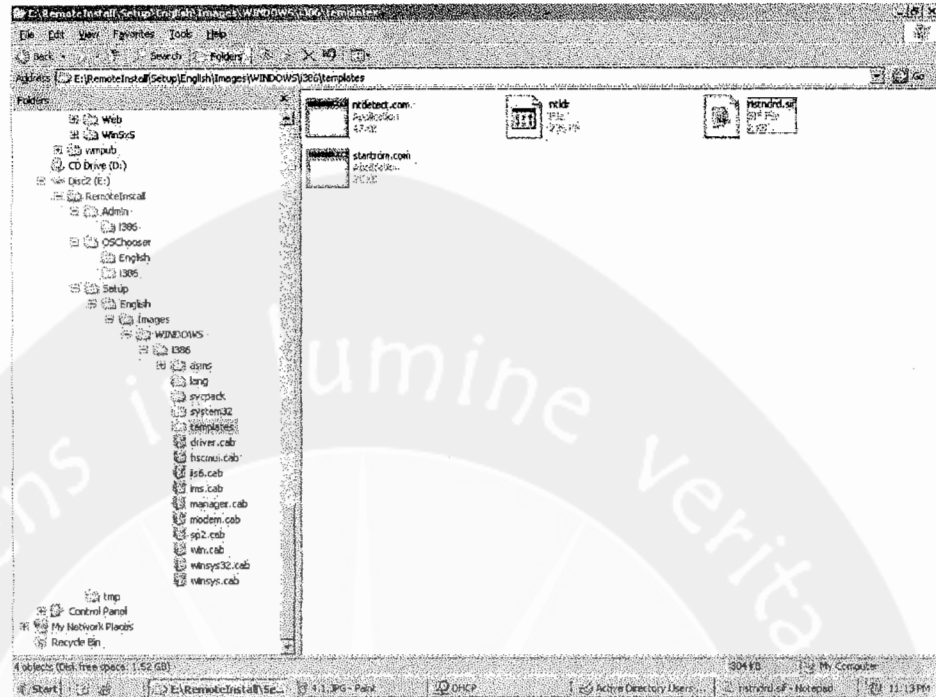
2. Langkah berikutnya adalah sama dengan pembuatan *images* seperti yang sudah dijelaskan di awal.

h. Melakukan *customize script*

Pada RIS server bisa dilakukan *customize script*. *Customize* yang dapat dilakukan meliputi :

1. Konfigurasi *OEM folder*. Pilih *Yes* untuk menyertakan data dan *driver* dalam instalasi.
2. Menambahkan *product key*, sehingga pada waktu instalasi tidak perlu memasukkan secara *manual*.
3. Menambahkan *user data* meliputi : *FullName*, *OrgName* dan *Computer Name* sehingga pada waktu instalasi tidak perlu memasukkan secara *manual*.
4. Menambahkan *password* untuk memproteksi sistem.
5. Menambahkan resolusi layar.
6. Menambahkan *domain administrator username* dan *password*.

Untuk melakukan *customize*, buka *Riprep.sif* file dengan *notepad* dan lakukan perubahan *script*. File ini terdapat pada *images folder* yang telah dibuat yaitu `\RemoteInstall\Setup\English\Images\Windows\i386\Templates` seperti pada gambar 23.



Gambar 23 Lokasi Script

Contoh perubahan script dapat dilihat sebagai berikut :

```
[data]
floppyless = "1"
msdosinitiated = "1"
OriSrc = "\\%SERVERNAME%\RemInst\%INSTALLPATH%\%MACHINETYPE%"
OriTyp = "4"
LocalSourceOnCD = 1
DisableAdminAccountOnDomainJoin = 1

[SetupData]
OsLoadOptions = "/noguiboot /fastdetect"
SetupSourceDevice =
"\Device\LanmanRedirector\%SERVERNAME%\RemInst\%INSTALLPATH%"

[Unattended]
OemPreinstall = Yes
FileSystem = LeaveAlone
ExtendOEMPartition = 0
TargetPath = \WINDOWS
OemSkipEula = yes
InstallFilePath = "\\%SERVERNAME%\RemInst\%INSTALLPATH%\%MACHINETYPE%"
LegacyNIC = 1

[UserData]
ProductID = btfrc-ghc7v-2wfbw-y42xw-9wdky
FullName = "Arie Nugroho"
OrgName = "Arie Enterprise"
ComputerName = "Arie01"

[CuiUnattended]
OemSkipWelcome = 1
OemSkipRegional = 1
TimeZone = %TIMEZONE%
```

AdminPassword = "arie"

[Display]

BitsPerPel = 16
XResolution = 1024
YResolution = 768
VRefresh = 72

[Networking]

[NetServices]
MS_Server=params.MS_P Sched

[Identification]

JoinDomain=arievirtual
DomainAdmin=Administrator
DomainAdminPassword=arie
DoOldStyleDomainJoin = Yes

[RemoteInstall]

Repartition = Yes
UseWholeDisk = Yes

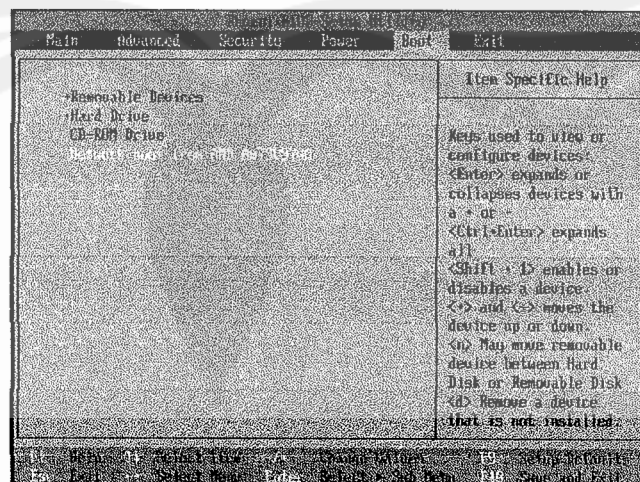
[OSChooser]

Description ="Microsoft Windows XP Professional"
Help ="Automatically installs Microsoft Windows XP Professional without prompting the user for input."
LaunchFile = "%INSTALLPATH%\%MACHINETYPE%\templates\startrom.com"
ImageType =Flat
Version="5.1 (0)"

2. Client

a. Konfigurasi BIOS

Lakukan pengesetan agar *network boot* sebagai *first boot priority*, apabila tidak ada pilihan tersebut bisa menjadikan *floppy* sebagai *first boot*.



Gambar 24 BIOS

b. Siapkan *disk boot*.

Langkah ini dilakukan untuk sistem BIOS yang tidak menyediakan pilihan *network boot* sebagai *first boot*. Jadi *first boot* adalah ke *floppy*. Siapkan *disk boot* yang telah dibuat sebelumnya.

c. Koneksikan dengan server.

Protokol yang digunakan dalam koneksi adalah TCP/IP. *Point* paling penting disini adalah *network interface card* yang dipakai *support* untuk *network boot*.

Penggunaan RIS

Langkah-langkah dalam menggunakan RIS adalah sebagai berikut :

1. Siapkan suatu *machine server* dalam keadaan hidup dan memiliki *RIS server*. *Machine server* ini harus memenuhi semua *requirement* dan telah dilakukan instalasi sebagaimana yang telah disebutkan di atas (lihat proses instalasi *server*).
2. Siapkan suatu *machine client* yang terkoneksi dengan *machine server*. *Machine client* ini harus memenuhi semua *requirement* dan telah dilakukan instalasi sebagaimana yang telah disebutkan di atas (lihat proses instalasi *client*).
3. Ketika *client* hidup dia akan memanfaatkan *Preboot eXecution Environment (PXE)* untuk melakukan *network boot*. *PXE* adalah suatu *environment* yang memungkinkan *boot* komputer dari *network interface* secara mandiri (bukan dari data *storage*). *PXE* akan meminta layanan *BOOTP* lewat jaringan. *BOOTP* dan *DHCP* sangat berhubungan erat. *Client* akan meminta *IP address* dengan parameter *MAC address* yang dimiliki oleh *network interface card* sebagai alamat unik. *Client* juga akan memberitahukan *GUID (Globally Unique Identifier)* atau *UUID (Universally Unique Identifier)*. *GUID/UUID* digunakan untuk mengidentifikasi *client*, sehingga dimungkinkan hanya *client* yang dikenali yang dapat di proses (lihat instalasi *server*).
Terlihat pada gambar, *client* mengirimkan *MAC : 00 0C 29 0B 6D DC* dan *GUID : 564D6746-6785-ECC8-854C-F1RE950B6DDC*.
Kemudian *client* tersebut mendapat alamat *IP 192.168.58.1*, setelah mendapat alamat *IP* dapat dikatakan *client* siap melakukan *network boot* seperti pada gambar 25.

```

Network boot from AMB AM79C970A
Copyright (C) 2003-2004 VMware, Inc.
Copyright (C) 1997-2000 Intel Corporation

CLIENT MAC ADDR: 00 0C 29 0B 6D 0C GUID: 564B6746-6785-ECC8-854C-F1BE9586D0C
CLIENT IP: 192.168.58.1 MASK: 255.255.255.0 DHCP IP: 192.168.58.5
GATEWAY IP: 192.168.58.2

```

```

Press F12 for network service boot
-

```

Gambar 25 Boot

Proses ini mungkin tidak berjalan sebagai mana mestinya, kesalahan yang mungkin terjadi antara lain:

a. *Network Interface Card* tidak mendukung.

Biasanya akan akan muncul pesan "*Error:Could not find a supported network card*". Solusinya adalah dengan mengganti NIC yang baru.

b. Tidak bisa *booting* lewat *network* atau *floppy drive*.

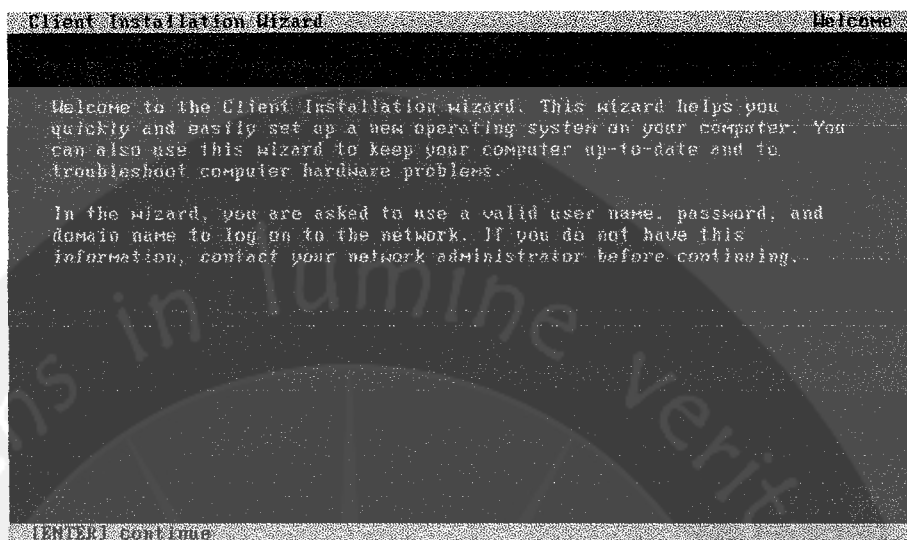
RIS mungkin berjalan apabila *client* melakukan *request boot* ke machine server pada saat dia hidup, sehingga *network boot* menjadi *first boot priority*. Untuk *motherboard* tertentu sudah menyediakan fasilitas *setting first boot* dari *network interface card*. Bila tidak ada fasilitas ini, lakukan pengesetan ke BIOS agar *first boot* ke *floppy device*. Masukkan *disk boot* yang telah dibuat dan ulangi proses *boot*. Disk boot ini akan memicu client melakukan *network boot*. Apabila proses gagal coba ganti master disk *floppy* dan ulangi proses.

c. Tidak bisa melakukan koneksi dengan *machine server*.

Biasanya akan akan muncul pesan "*No reply from a server, press a key to reboot system*". Solusi yang bisa dilakukan diantaranya :

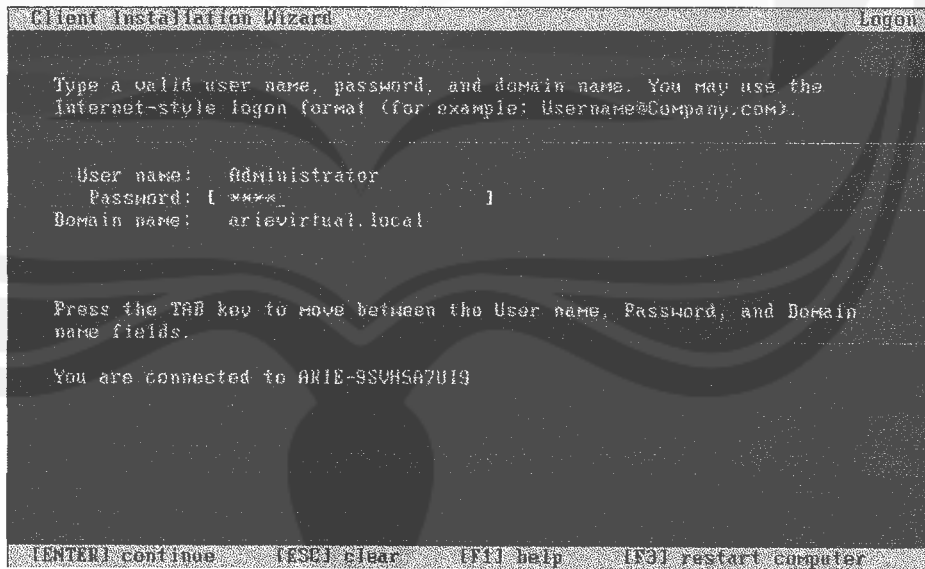
1. Set up jaringan. Pastikan Konfigurasi DNS dan DHCP terkonfigurasi dengan baik.
2. Pastikan RIS server merupakan bagian dari Active Directory.

4. Apabila tidak muncul kesalahan maka akan tampil seperti pada gambar 26. Tekan *enter*.



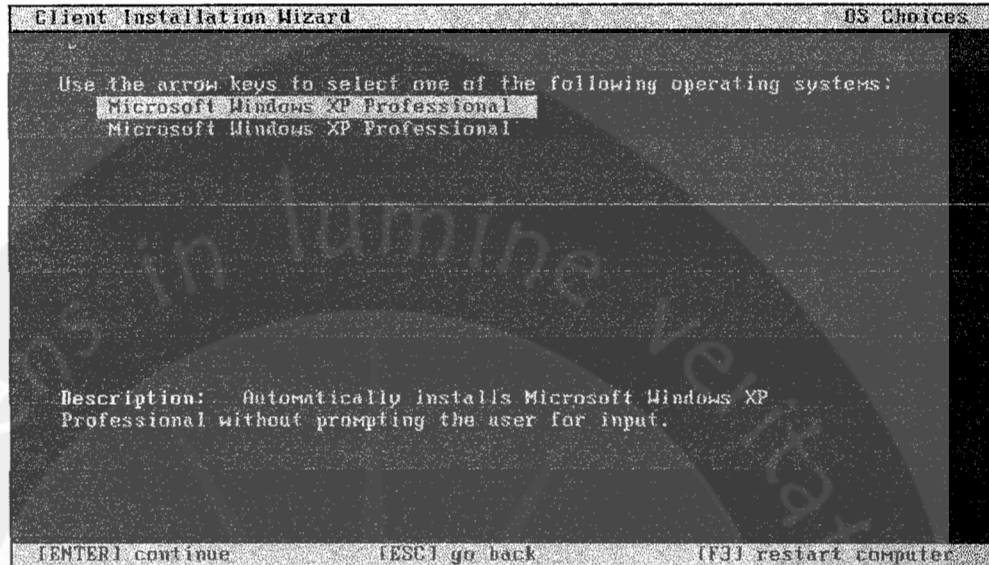
Gambar 26 Welcome messages

5. Selanjutnya sistem meminta masukkan *username* dan *password* untuk *log* ke layanan *domain server* seperti pada gambar 27. Masukkan data kemudian tekan *enter*.



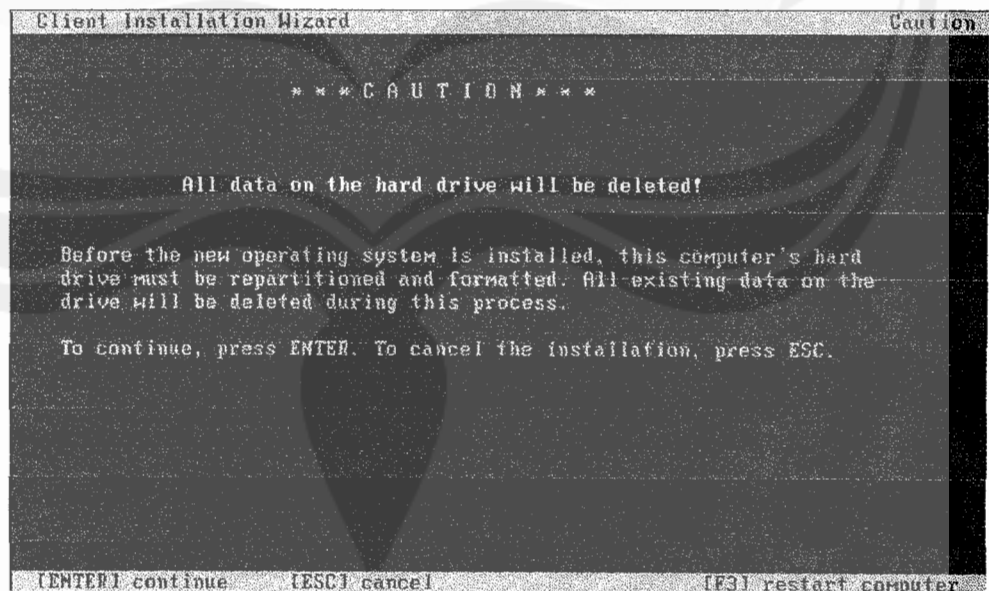
Gambar 27 Insert Password

6. Sistem memberikan pilihan *image* sistem operasi mana yang akan di-*install* seperti pada gambar 28. Pilih kemudian tekan *enter*.



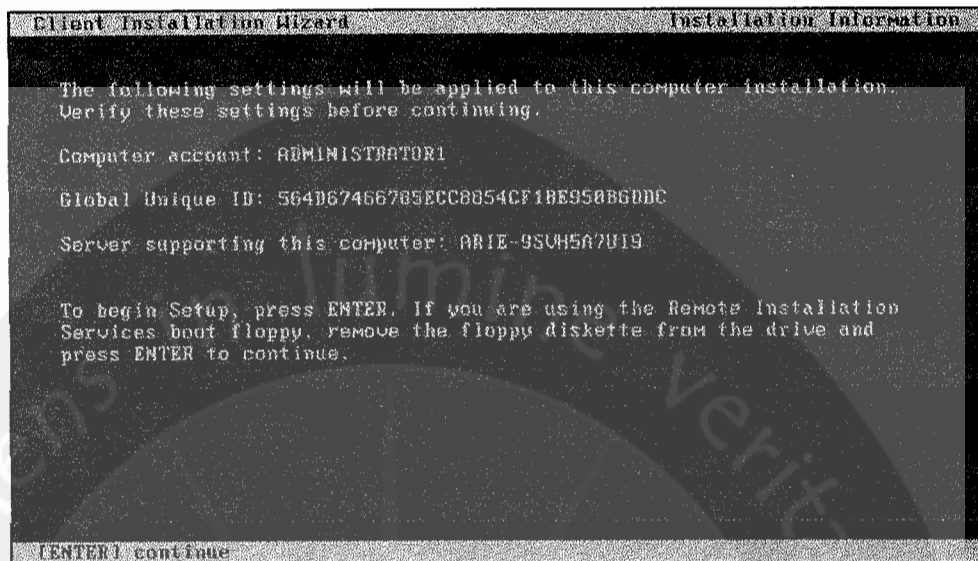
Gambar 28 Choose Images

7. Sistem memberikan *alert* bahwa semua data di *hard drive client* akan terhapus selama instalasi seperti pada gambar 29. Tekan *enter*.



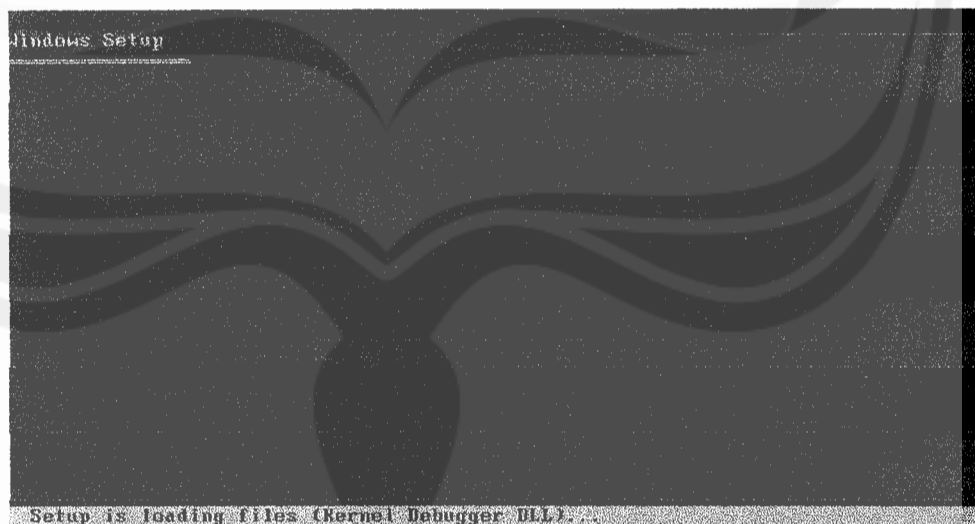
Gambar 29 Delete messages

8. Sistem akan menampilkan informasi *setting* instalasi seperti pada gambar 30. Tekan *enter*.



Gambar 30 *Information setting*

9. Apabila semua proses berjalan lancar maka akan muncul tampilan sebagaimana gambar 31. Setelah proses ini maka instalasi sistem operasi bisa dilakukan. Proses selanjutnya sebagaimana melakukan instalasi biasa.



Gambar 31 Mulai menginstalasi sistem operasi

Referensi :

1. Ivens, Kathy, et al., 2003, *Windows Server 2003 : The Complete Reference*, McGraw-Hill, California.
2. Mathews, Martin S., 2003, *Windows Server 2003 : A Beginner's Guide*, McGraw-Hill, California.

SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

Ancilla 1.0

Untuk :

Tugas akhir

Disusun oleh:


Aryanto Adi Nugroho

02 07 03605

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta	Nomor Dokumen		Halaman
		<i>SKPL-Ancilla</i>		
		Revisi		

DAFTAR PERUBAHAN

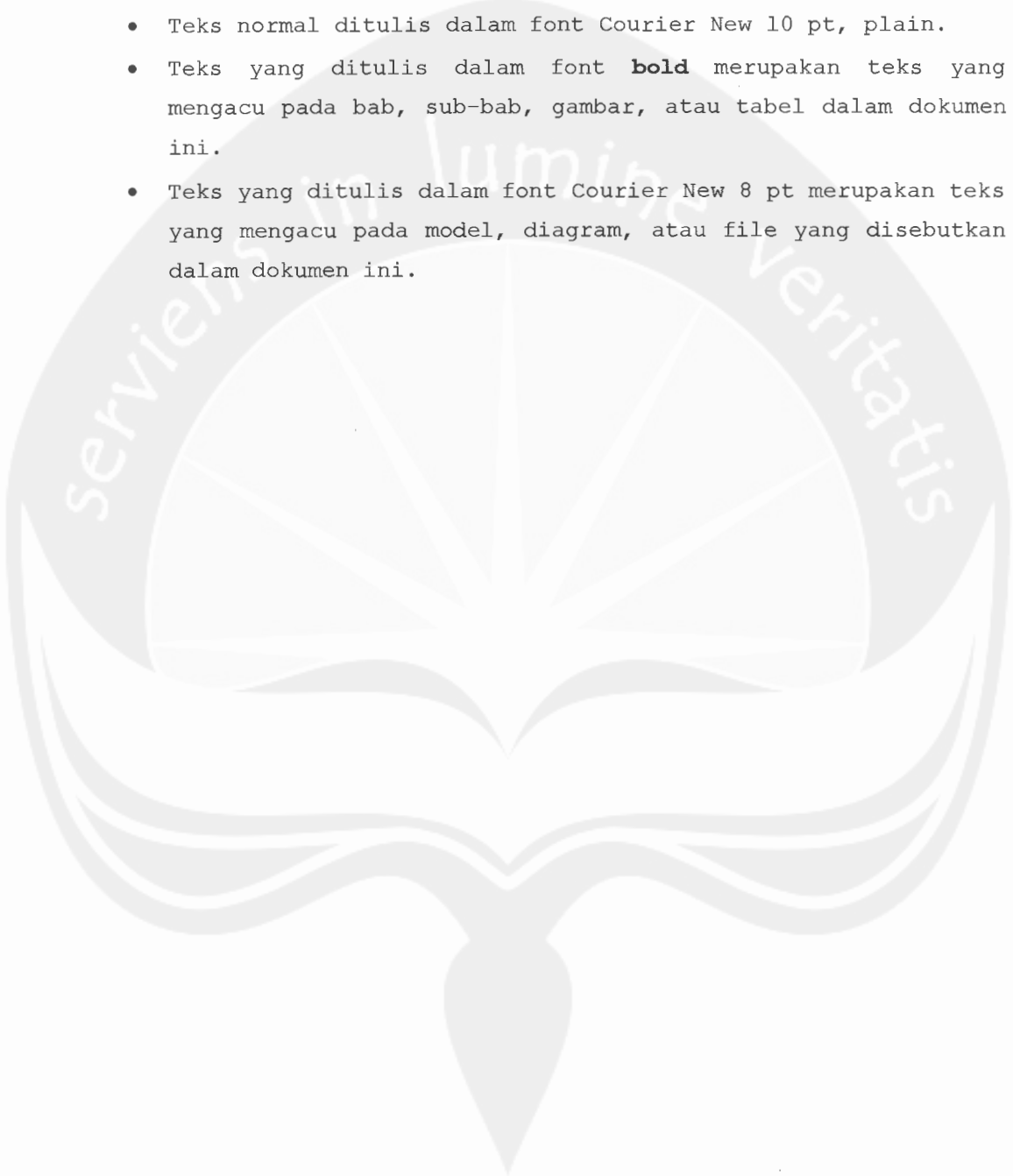
Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E
Ditulis oleh	AR					
Diperiksa oleh	KSN KA	<i>Package Diagram</i>				
Disetujui oleh	KSN KA					

NOTASI DOKUMEN

Notasi yang digunakan dalam dokumen ini adalah sebagai berikut :

- Teks normal ditulis dalam font Courier New 10 pt, plain.
- Teks yang ditulis dalam font **bold** merupakan teks yang mengacu pada bab, sub-bab, gambar, atau tabel dalam dokumen ini.
- Teks yang ditulis dalam font Courier New 8 pt merupakan teks yang mengacu pada model, diagram, atau file yang disebutkan dalam dokumen ini.



DAFTAR ISI

Daftar Perubahan	2
Notasi Dokumen	3
Daftar Isi	4
Daftar Gambar	5
Daftar Tabel	5
1 Pendahuluan	6
1.1 Tujuan	6
1.2 Lingkup Masalah	6
1.3 Definisi, Istilah dan Singkatan	8
1.4 Referensi	8
1.5 Deskripsi Umum (Overview)	8
2 Deskripsi Keseluruhan	9
2.1 Perspektif Produk	9
2.2 Kebutuhan fungsionalitas perangkat Lunak	10
2.3 Karakteristik Aktor...	12
3 Spesifikasi Kebutuhan	12
3.1 Spesifikasi Kebutuhan Fungsionalitas	12
3.1.1 Spesifikasi Use Case : Login	12
3.1.2 Spesifikasi Use Case : Monitoring Client	13
3.1.3 Spesifikasi Use Case : Controlling Client	13
3.1.4 Spesifikasi Use Case : Account Management	14
3.1.5 Spesifikasi Use Case : Data Management	14
3.1.6 Spesifikasi Use Case : Scan Active Client.....	15
3.2 Spesifikasi Kebutuhan Non-Fungsionalitas	16
3.2.1 Kebutuhan Antarmuka Eksternal	16
3.2.2 Antarmuka Pemakai	16
3.2.3 Antarmuka Perangkat Keras	16
3.2.4 Antarmuka perangkat Lunak	16
3.2.5 Antarmuka Komunikasi	17
3.2.6 Persistent Data	17
3.2.6.1 Entity Relationship Diagram	17
4 Realisasi Use Case	17
4.1 Static Structure Diagram.....	17
4.2 Interaction Diagram	20
Appendix A	27

DAFTAR GAMBAR

Gambar 1.	<i>Activity Diagram</i>	7
Gambar 2.	Implementasi Perangkat Lunak pada LAN.....	10
Gambar 3.	<i>Use case</i> Perangkat Lunak Ancilla 1.0.....	10
Gambar 4.	<i>Entity Relational Diagram(ERD)</i>	17
Gambar 5.	<i>Package Dependency</i>	17
Gambar 6.	<i>Analysis Class Diagram : Use Case</i> AncillaRein.....	18
Gambar 7.	<i>Analysis Class Diagram : Use Case</i> AncillaRein.....	18
Gambar 8.	<i>Analysis Class Diagram : Use Case</i> AncillaRemote.....	18
Gambar 9.	<i>Analysis Class Diagram : Use Case</i> Login.....	19
Gambar 10.	<i>Analysis Class Diagram : Use Case</i> Monitoring Client.....	19
Gambar 11.	<i>Analysis Class Diagram : Use Case</i> Controlling Client.....	19
Gambar 12.	<i>Analysis Class Diagram : Use Case</i> Account Management.....	19
Gambar 13.	<i>Analysis Class Diagram : Use Case</i> Data Management.....	20
Gambar 14.	<i>Analysis Class Diagram : Use Case</i> Scan Active Client.....	20
Gambar 15.	<i>Analysis Collaboration Diagram : Use Case</i> Login.....	20
Gambar 16.	<i>Analysis Collaboration Diagram : Use Case</i> Scan Active Client.....	21
Gambar 17.	<i>Analysis Collaboration Diagram : Use Case</i> Clean Data.....	21
Gambar 18.	<i>Analysis Collaboration Diagram : Use Case</i> AccountSet.....	21
Gambar 19.	<i>Analysis Collaboration Diagram : Use Case</i> Lock.....	22
Gambar 20.	<i>Analysis Collaboration Diagram : Use Case</i> LogOff.....	22
Gambar 21.	<i>Analysis Collaboration Diagram : Use Case</i> Restart.....	22
Gambar 22.	<i>Analysis Collaboration Diagram : Use Case</i> Shutdown.....	23
Gambar 23.	<i>Analysis Collaboration Diagram : Use Case</i> Wake Up.....	23
Gambar 24.	<i>Analysis Collaboration Diagram : Use Case</i> Block Input.....	23
Gambar 25.	<i>Analysis Collaboration Diagram : Use Case</i> Unblock Input.....	24
Gambar 26.	<i>Analysis Collaboration Diagram : Use Case</i> Run Process.....	24
Gambar 27.	<i>Analysis Collaboration Diagram : Use Case</i> Kill Process.....	24
Gambar 28.	<i>Analysis Collaboration Diagram : Use Case</i> Ping Process.....	25
Gambar 29.	<i>Analysis Collaboration Diagram : Use Case</i> View User.....	25
Gambar 30.	<i>Analysis Collaboration Diagram : Use Case</i> View Process.....	25
Gambar 31.	<i>Analysis Collaboration Diagram : Use Case</i> View Data.....	26

DAFTAR TABEL

Tabel 1.	Spesifikasi <i>Use Case</i> : Login	12
Tabel 2.	Spesifikasi <i>Use Case</i> : Monitoring Client.....	13
Tabel 3.	Spesifikasi <i>Use Case</i> : Controlling Client.....	13
Tabel 4.	Spesifikasi <i>Use Case</i> : Account Management.....	14
Tabel 5.	Spesifikasi <i>Use Case</i> : Account Management.....	14
Tabel 6.	Spesifikasi <i>Use Case</i> : Scan Active Client	15

1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini dibuat untuk menyediakan deskripsi lengkap mengenai kebutuhan fungsionalitas dan non-fungsionalitas dari perangkat lunak Ancilla, versi 1.0, yang digunakan untuk memenuhi beberapa fungsionalitas dalam sistem pemantauan dan pengembangan komputer secara terpusat. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis untuk pengembangan perangkat lunak dan ditujukan untuk orang-orang lain yang tertarik untuk mengembangkan perangkat lunak ini lebih lanjut. SKPL ini akan menjelaskan kebutuhan-kebutuhan yang harus tersedia agar perangkat lunak yang diharapkan dapat terwujud.

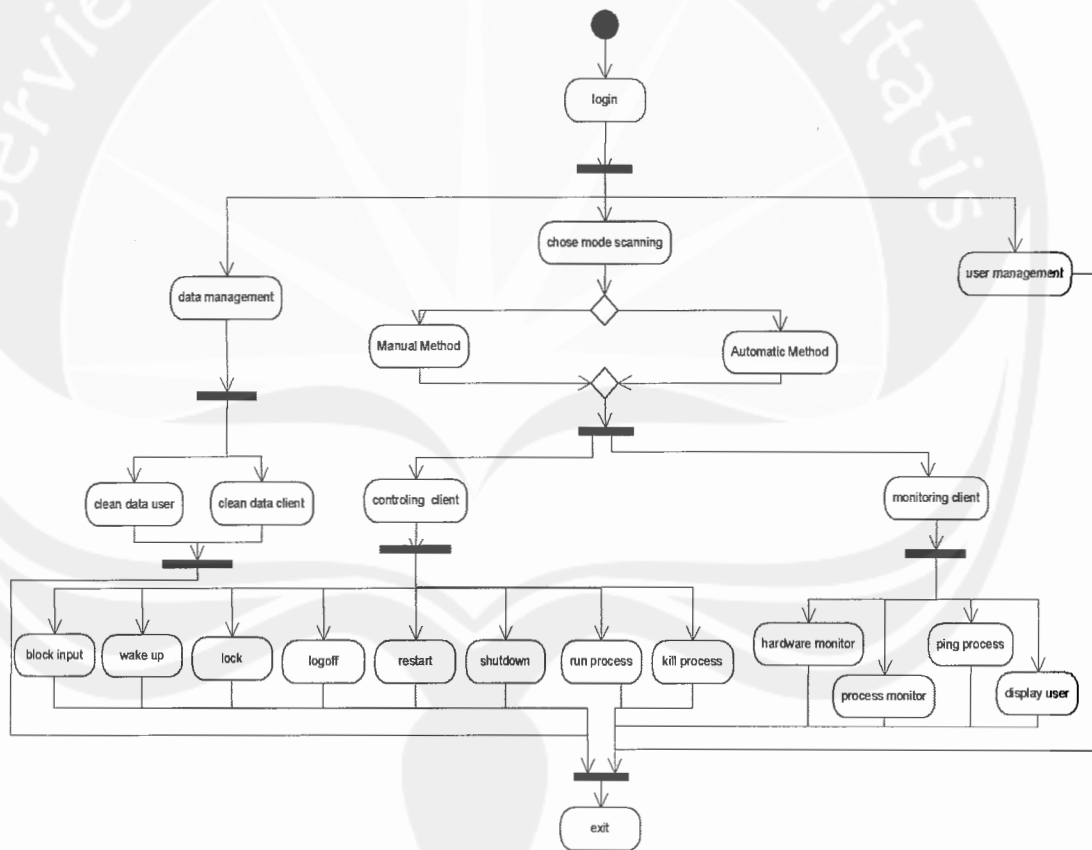
1.2 Lingkup Masalah

Sebagaimana LAN pada umumnya, Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta juga memerlukan suatu penanganan manajemen jaringan. Laboratorium ini terdiri dari puluhan komputer bersistem operasi Windows XP yang saling terkoneksi di bawah kendali server yang menggunakan Windows Server 2003. Administrator membutuhkan suatu aplikasi yang memungkinkan penanganan manajemen jaringan di laboratorium ini menjadi lebih mudah.

Penanganan yang dimaksudkan meliputi pemantauan dan pengendalian komputer secara terpusat. Pemantauan yang diperlukan antara lain untuk mengetahui *user* yang sedang atau yang telah *login* di suatu komputer(*client network*), *process* yang *running*, informasi sumber daya perangkat keras/*hardware* beserta *alert* apabila ada perubahan konfigurasi. Pengendalian meliputi fungsi *wakeup*, *lock*, *restart*, *logoff*, *shutdown* serta *input blocking* pada sebuah atau keseluruhan komputer. Fungsi pengendalian lain yang diperlukan adalah melakukan *running program/process* maupun *killing process* pada sebuah atau keseluruhan komputer secara terpusat. Seluruh fungsi, baik pemantauan maupun pengendalian dilakukan secara terpusat dari satu komputer(*server network*).

Perangkat lunak yang akan dikembangkan (Ancilla) diharapkan mampu menangani kebutuhan fungsionalitas yang telah disebutkan diatas. Perangkat lunak ini dikembangkan secara khusus untuk diterapkan di Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta.

Perangkat Lunak Ancilla ini akan mencoba memecahkan masalah tersebut dengan cara menyediakan fungsi-fungsi pemantauan dan pengendalian. Fungsi dan alur yang dimiliki dapat dilihat dalam *Activity Diagram*. *Activity Diagram* ini untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja. *Activity diagram* dalam perangkat lunak ini dapat dilihat dalam gambar berikut :



Gambar 1 Activity Diagram

1.3 Definisi, Istilah dan Singkatan

Untuk definisi dan singkatan yang digunakan didalam dokumen ini dapat mengacu pada **Apendiks A : Daftar Istilah dan Singkatan**.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Nugroho, Antonius Sunarmanto, *Spesifikasi Kebutuhan Perangkat Lunak Pengembangan Aplikasi Monitoring Koneksi Jaringan Berbasis TCP/IP (TCPM)*, Universitas Atma Jaya Yogyakarta, 2006.
2. Hasan, Hardi, *Spesifikasi Kebutuhan Perangkat Lunak Sistem Pemantauan dan Pengendalian Komputer Jarak Jauh (SPPKJJ)*, Universitas Atma Jaya Yogyakarta, 2005.
3. Fowler, Martin, *UML Distilled Ed. 3*, Pearson Education Inc., Penerbit Andi, Yogyakarta, 2005.
4. Munawar, *Pemodelan visual dengan UML*, Graha Ilmu, Yogyakarta, 2005.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 4 bagian utama. Bagian pertama adalah **Pendahuluan** berisi penjelasan mengenai dokumen SKPL yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua adalah **Deskripsi Keseluruhan** berisi penjelasan umum tentang perangkat lunak yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak tersebut.

Bagian ketiga adalah **Spesifikasi Kebutuhan** berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak yang akan dikembangkan.

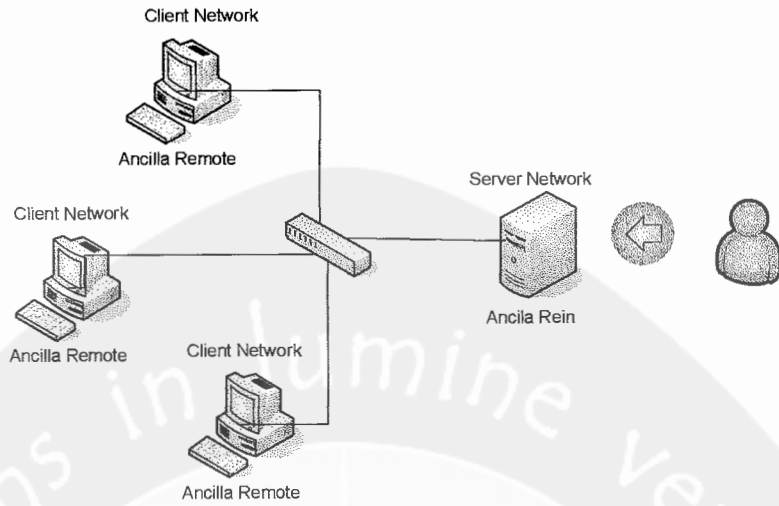
2 Deskripsi Keseluruhan

2.1 Perspektif Produk

Ancilla 1.0 adalah suatu perangkat lunak yang dikembangkan untuk melakukan pemantauan dan pengendalian komputer terpusat di Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta. Perangkat lunak ini tidak memiliki semua fungsionalitas yang diperlukan dalam manajemen jaringan. Fungsionalitas yang dimiliki adalah sesuai kebutuhan/permintaan yang telah disebutkan di lingkup masalah diatas.

Perangkat lunak Ancilla 1.0 dibangun menggunakan *.Net Framework* dengan bahasa pemrograman *C#*. Perangkat lunak ini menggunakan *database Ms. Access*. Pengguna berinteraksi dengan sistem dalam bentuk antarmuka *GUI*, dengan piranti masukan yang digunakan adalah *mouse* dan *keyboard*. Seluruh pertukaran data menggunakan protokol *UDP*.

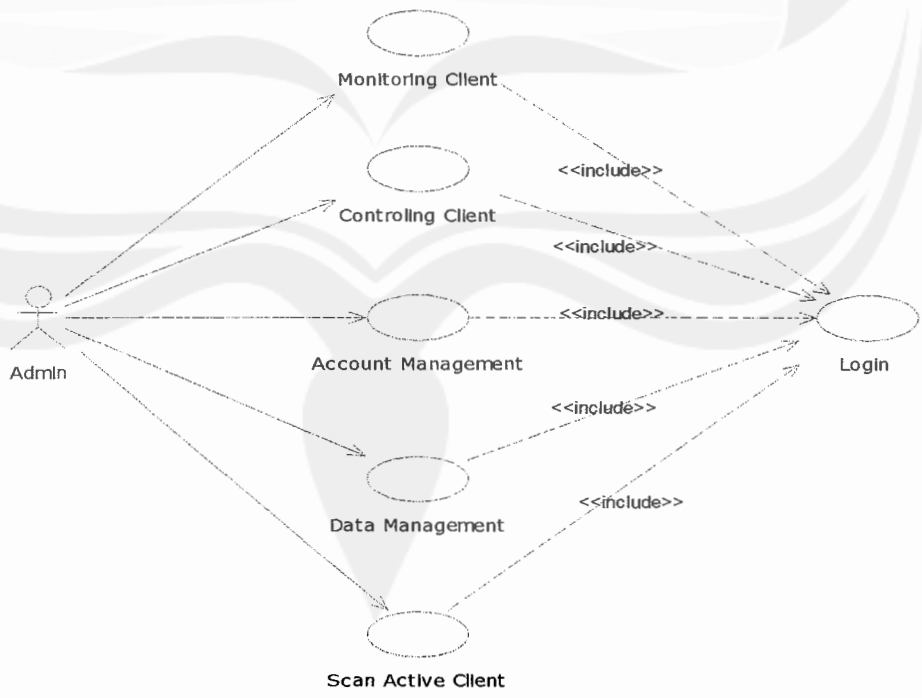
Perangkat lunak terdiri dari 2 bagian yaitu : Ancilla 1.0 *REIN* dan Ancilla 1.0 *REMOTE*. Ancilla 1.0 *REMOTE* adalah perangkat lunak yang tertanam pada seluruh komputer dalam jaringan (*client network*) yang akan dipantau dan dikendalikan. Perangkat lunak ini menunggu perintah kemudian melakukan eksekusi perintah tersebut. Ancilla 1.0 *REIN* adalah perangkat lunak yang tertanam di komputer pengendali (*server network*). Perangkat lunak ini digunakan untuk meminta layanan ke Ancilla 1.0 *REMOTE* (*client network*). Bagian *REIN* inilah yang digunakan untuk memantau dan mengendalikan komputer di Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta. Konfigurasi sistem dapat dilihat pada gambar 2.



Gambar 2. Implementasi Perangkat Lunak pada LAN

2.2 Kebutuhan Fungsionalitas Perangkat Lunak.

Kebutuhan fungsionalitas perangkat lunak ini akan direpresentasikan kedalam suatu *use case*. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberikan penjelasan bagaimana sistem digunakan. *Use case* sistem ini dapat dilihat sebagai berikut :



Gambar 3. Use case Perangkat Lunak Ancilla 1.0

2.2.1 Use Case : Login

Deskripsi : Use case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Sistem tidak menyediakan multiple user (*default user* adalah administrator). *Login* didasarkan pada *password* yang berupa rangkaian karakter.

Lihat : Spesifikasi Use Case : Login

2.2.2 Use Case : Monitoring Client

Deskripsi : Use case ini digunakan oleh aktor untuk memantau data *client*. Pemantauan tersebut meliputi user yang sedang atau yang telah *login* di suatu komputer(*client network*), *process* yang *running*, informasi sumber daya perangkat keras/*hardware* beserta *alert* apabila ada perubahan konfigurasi.

Lihat : Spesifikasi Use Case : Monitoring Client

2.2.3 Use Case : Controlling client

Deskripsi : Use case ini untuk mengendalikan komputer *client* dari komputer administrator (*server*). Proses *control* yang disediakan antara lain: *wakeup*, *lock*, *restart*, *logoff*, *shutdown* serta *input blocking* pada sebuah atau keseluruhan komputer. Fungsi pengendalian lain yang diperlukan adalah melakukan *running program/process* maupun *killing process* pada sebuah atau keseluruhan komputer secara terpusat.

Lihat : Spesifikasi Use Case : Controlling client

2.2.4 Use Case : Account Management

Deskripsi : Use case ini digunakan untuk melakukan perubahan data *account user system*.

Lihat : Spesifikasi Use Case : User Management

2.2.5 Use Case : Data Management

Deskripsi : Use case ini digunakan oleh aktor untuk melakukan perubahan data *client*.

Lihat : Spesifikasi Use Case : Data Management

2.2.6 Use Case : Scan Active Client

Deskripsi : Use case ini digunakan oleh aktor untuk melakukan *scanning* terhadap *client* yang aktif dalam jaringan, *me-record* informasi secara otomatis atas informasi yang diberikan *client* dan memberikan *alert* pada aktor apabila ada perubahan konfigurasi *hardware*.

Lihat : Spesifikasi Use Case : Scan Active Client

2.3 Karakteristik Aktor

Pengguna perangkat lunak ini adalah administrator jaringan. Administrator jaringan memiliki karakteristik sebagai berikut :

1. Memahami pengoperasian komputer secara aktif.
2. Memahami sistem komputer tempat perangkat lunak dijalankan.
3. Memahami konsep dan operasi jaringan LAN beserta *maintenance* yang harus dilakukan.
4. Memiliki hak akses utama dalam jaringan Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta.

3 Spesifikasi Kebutuhan

3.1 Spesifikasi Kebutuhan Fungsionalitas

3.1.1 Spesifikasi Use Case : Login

Tabel 1 Spesifikasi Use Case : Login

<i>Use Case ID</i>	UC-Ancilla-01
<i>Use Case Name</i>	<i>Login</i>
<i>Use Case Type</i>	<i>Essential</i>
<i>Priority</i>	<i>High</i>
<i>Actors</i>	Administrator
<i>Description</i>	Use case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada <i>password</i> yang berupa rangkaian karakter.
<i>Preconditions</i>	-
<i>Basic Path</i>	<ol style="list-style-type: none">1. Sistem menampilkan antarmuka untuk <i>login</i>.2. Admin memasukkan <i>password</i>.3. Sistem memeriksa <i>password</i> admin/user.4. Sistem memberikan akses ke admin/user.
<i>Alternative Paths</i>	-
<i>Postconditions</i>	Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.
<i>Exception Paths</i>	<ol style="list-style-type: none">1. <i>Username</i> atau <i>password</i> yang dimasukkan tidak sesuai

	(setelah Tabel 3.1 <i>Basic Path 2</i>) a. Sistem menampilkan peringatan bahwa password tidak sesuai. b. Kembali ke Tabel 3.1 <i>Basic Path 2</i> .
<i>Extends</i>	-
<i>Includes</i>	-
<i>Business Rules</i>	-

3.1.2 Spesifikasi Use Case : Monitoring Client

Tabel 2. Spesifikasi Use Case : Monitoring Client

<i>Use Case ID</i>	UC-Ancilla-02
<i>Use Case Name</i>	<i>Monitoring Client</i>
<i>Use Case Type</i>	<i>Essential</i>
<i>Priority</i>	<i>High</i>
<i>Actors</i>	Administrator
<i>Description</i>	Use case ini digunakan oleh aktor untuk memonitor client. <i>Monitoring</i> yang dilakukan mencakup hal-hal yang sudah dijelaskan sebelumnya.
<i>Preconditions</i>	Use Case : Scan Active Client sudah dilaksanakan dan aktor sudah berhasil memasuki sistem.
<i>Basic Path</i>	<ol style="list-style-type: none"> 1. Sistem memberikan membutuhkan parameter yaitu <i>destination</i> komputer tujuan. 2. Sistem mengirimkan <i>packet request</i> informasi. 3. Sistem melakukan <i>record</i> informasi yang dikirim dan menampilkannya. 4. Sistem memberikan <i>alert</i> apabila ada perubahan konfigurasi <i>hardware</i>.
<i>Alternative Paths</i>	-
<i>Postconditions</i>	<ol style="list-style-type: none"> a. Data client ditampilkan. b. <i>Alert</i> ditampilkan (apabila ada)
<i>Exception Paths</i>	<ol style="list-style-type: none"> a. Jika koneksi gagal, baik berupa <i>power off</i> maupun gagal koneksi maka statusnya akan ditampilkan. b. Kembali ke Tabel 3.7 <i>Basic Path 1</i>.
<i>Extends</i>	-
<i>Includes</i>	Use Case : Monitoring Client
<i>Business Rules</i>	-

3.1.3 Spesifikasi Use Case : Controlling client

Tabel 3. Spesifikasi Use Case : Controlling client

<i>Use Case ID</i>	UC-Ancilla-03
<i>Use Case Name</i>	<i>Controlling client</i>
<i>Use Case Type</i>	<i>Essential</i>
<i>Priority</i>	<i>High</i>
<i>Actors</i>	Administrator
<i>Description</i>	Use case ini digunakan oleh aktor untuk melakukan

	<i>control secara remote</i> dari komputer server.
Preconditions	Use Case : Login sudah dilaksanakan dan aktor sudah berhasil memasuki sistem.
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk pilihan antara lain: <i>wakeup, lock, restart, logoff, shutdown, input blocking</i> serta <i>run/kill process</i>. 2. Aktor memilih fungsi yang akan dilakukan. 3. Sistem menjalankan fungsi yang dipilih dengan mengirimkan paket perintah ke <i>client</i>.
Alternative Paths	-
Postconditions	<i>Client</i> menjalankan perintah yang dipilih.
Exception Paths	Pengiriman paket belum berhasil dilakukan. Kembali ke tabel 3.2 Basic Path 1.
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.3 Spesifikasi Use Case : Account Management

Tabel 4. Spesifikasi Use Case : Account Management

Use Case ID	UC-Ancilla-04
Use Case Name	Account Management
Use Case Type	Essential
Priority	Low
Actors	Administrator
Description	Use case ini digunakan oleh aktor untuk melakukan perubahan terhadap <i>data account</i> .
Preconditions	Use Case : Login sudah dilaksanakan dan aktor sudah berhasil memasuki sistem.
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>data account</i> yang dimiliki user. 2. Administrator memasukkan <i>data account</i> baru. 3. Sistem akan menyimpan <i>data account</i> di <i>database</i>.
Alternative Paths	-
Postconditions	<i>Data account</i> berubah.
Exception Paths	Sistem gagal dalam melakukan penyimpanan muncul <i>record error</i> .
Extends	-
Includes	Use Case : Login
Business Rules	-

3.1.3.1 Spesifikasi Use Case : Data Management

Tabel 5. Spesifikasi Use Case : Data management

Use Case ID	UC-Ancilla-05
Use Case Name	Data Management
Use Case Type	Essential
Priority	Low
Actors	Administrator, user

<i>Description</i>	<i>Use case</i> ini digunakan oleh aktor untuk melakukan penghapusan terhadap <i>database</i> . Proses dilakukan untuk <i>management storage</i> .
<i>Preconditions</i>	Use Case : Login sudah dilaksanakan dan aktor sudah berhasil memasuki sistem..
<i>Basic Path</i>	Sistem memberikan pilihan untuk mengosongkan data <i>log user</i> maupun data sistem <i>client</i> . Aktor memberikan pilihan dan sistem melakukan proses ke <i>database</i> .
<i>Alternative Paths</i>	-
<i>Postconditions</i>	Data berubah.
<i>Exception Paths</i>	Sistem gagal dalam melakukan penyimpanan muncul <i>record error</i> .
<i>Extends</i>	-
<i>Includes</i>	Use Case : Login
<i>Business Rules</i>	-

3.1.4 Spesifikasi Use Case : Scan Active Client

Tabel 6. Spesifikasi Use Case : Scan Active Client

<i>Use Case ID</i>	UC-Ancilla-06
<i>Use Case Name</i>	Scan Active Client
<i>Use Case Type</i>	Essential
<i>Priority</i>	High
<i>Actors</i>	Administrator
<i>Description</i>	<i>Use case</i> ini digunakan untuk menampilkan data <i>client</i> yang aktif dalam jaringan, sekaligus melakukan <i>record data</i> . Data disimpan dan disesuaikan dalam <i>database</i> . Apabila ada perubahan konfigurasi terdapat <i>alert</i> . Metode <i>scanning</i> ada 2, yaitu : <i>manual</i> dan <i>automatic</i> . <i>Manual</i> berarti aktor diminta melakukan <i>scanning</i> dengan melakukan aksi tertentu (menekan tombol <i>scan</i>). <i>Automatic</i> berarti sistem melakukan <i>scanning</i> secara otomatis per satuan waktu.
<i>Preconditions</i>	Use Case : Login sudah dilaksanakan dan aktor sudah berhasil memasuki sistem.
<i>Basic Path</i>	Sistem menampilkan antarmuka untuk menampilkan <i>client</i> yang aktif.
<i>Alternative Paths</i>	-
<i>Postconditions</i>	Aktor dapat mengetahui informasi <i>network status</i> .
<i>Exception Paths</i>	Sistem gagal dalam melakukan <i>scanning</i> muncul <i>info error</i> .
<i>Extends</i>	-
<i>Includes</i>	Use Case : Login
<i>Business Rules</i>	-

3.2 Spesifikasi Kebutuhan Non-Fungsionalitas

3.2.1 Kebutuhan Antarmuka Eksternal

Kebutuhan antar muka eksternal pada perangkat lunak Ancilla 1.0 meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak dan antarmuka komunikasi.

3.2.2 Antarmuka Pemakai

Pengguna berinteraksi langsung dengan sistem Ancilla 1.0 dengan antarmuka berbasis GUI. Piranti masukan yang digunakan untuk memasukkan data masukan adalah *keyboard* dan *mouse*. Sedangkan keluaran dari sistem berupa hasil eksekusi perintah dan ditampilkan langsung ke layar monitor.

3.2.3 Antarmuka Perangkat Keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak Ancilla 1.0 adalah:

1. *PC Compatible*.
2. *Keyboard* dan *Mouse*.
3. *Motherboard* dan *Network Adapter Card* yang mendukung teknologi *Wake-On LAN*. Secara khusus dilakukan konfigurasi *enable wake up* pada BIOS.

3.2.4 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak Ancilla 1.0 adalah sebagai berikut :

1. Nama : Windows XP
Sumber : Microsoft.
Sebagai sistem operasi komputer.
2. Nama : .Net Framework 2.0
Sumber : Microsoft
Sebagai *framework*.
3. Nama : C#.NET
Sumber : Microsoft
Sebagai bahasa pemrograman perangkat lunak.

3. Nama : MS. Access

Sumber : Microsoft

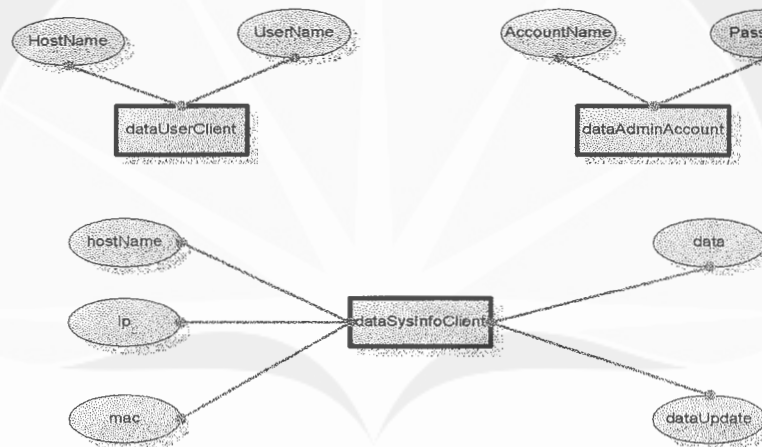
Sebagai *database* yang dibutuhkan dalam mengoperasikan perangkat lunak.

3.2.5 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak Ancilla 1.0 menggunakan protokol TCP/IP karena perangkat lunak ini digunakan untuk mendukung model sistem *client-server*.

3.2.6 Persistent Data

3.2.6.1 Entity Relational Diagram (ERD)



Gambar 4. Entity Relational Diagram (ERD)

4 Realisasi Use Case

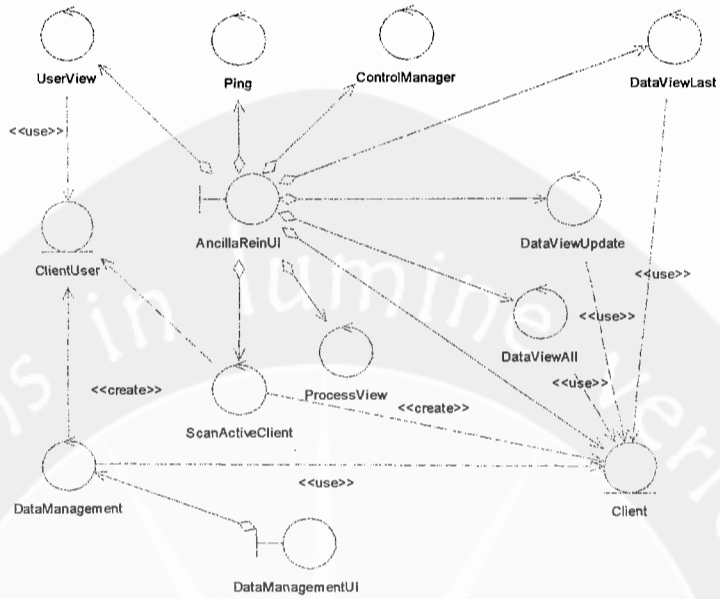
4.1 Static Structure Diagram

4.1.1 Analysis Class Diagram : Package Dependencies

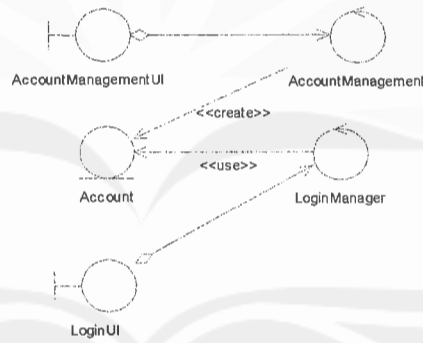


Gambar 5. Package Dependencies

4.1.2 Analysis Class Diagram : Package AncillaRein



Gambar 6 Analysis Class Diagram : Use Case AncillaRein



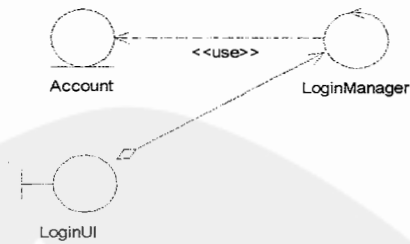
Gambar 7 Analysis Class Diagram : Use Case AncillaRein

4.1.3 Analysis Class Diagram : Package AncillaRemote



Gambar 8. Analysis Class Diagram : Use Case AncillaRemote

4.1.4 Analysis Class Diagram : Use case Login



Gambar 9. Analysis Class Diagram : Use Case Login

4.1.5 Analysis Class Diagram : Use Case Monitoring Client



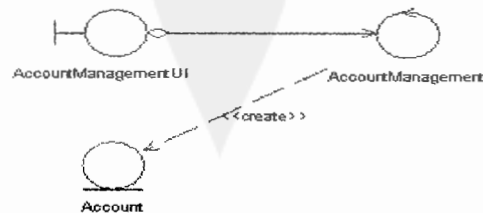
Gambar 10. Analisis Class Diagram : Use Case Monitoring Client

4.1.6 Analysis Class Diagram : Use Case Controlling Client



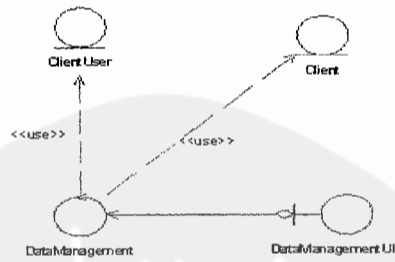
Gambar 11. Analisis Class Diagram : Use Case Controlling Client

4.1.7 Analysis Class Diagram : Use Case Account Management



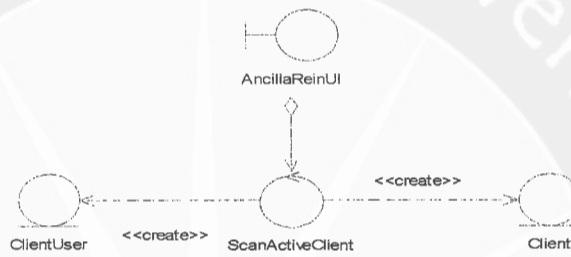
Gambar 12. Analisis Class Diagram : Use Case Account Management

4.1.8 Analysis Class Diagram : Use Case Data Management



Gambar 13. Analysis Class Diagram : Use Case Data Management

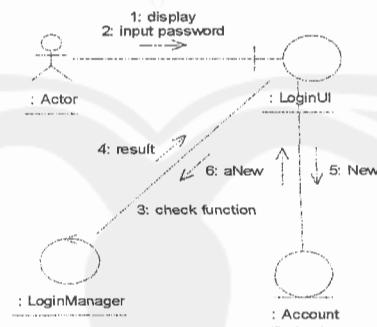
4.1.9 Analysis Class Diagram : Use Case Scanning Active Client



Gambar 14. Analysis Class Diagram : Use Case Scanning Active Client

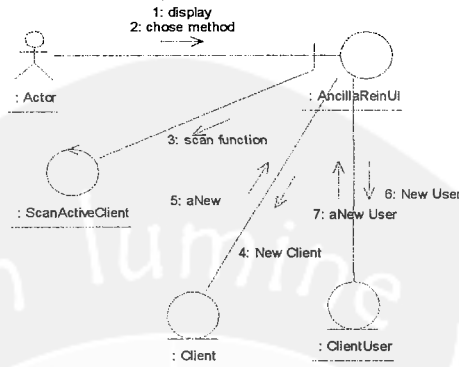
4.2 Interaction Diagram

4.2.1 Analysis Collaboration Diagram : Use Case Login



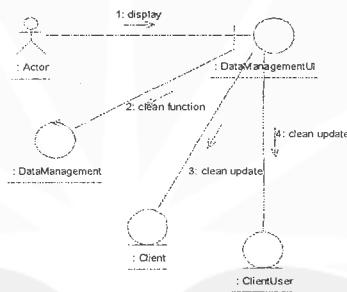
Gambar 15. Analysis Collaboration Diagram : Use Case Login

4.2.2 Analysis Collaboration Diagram : Use Case Scanning Active Client



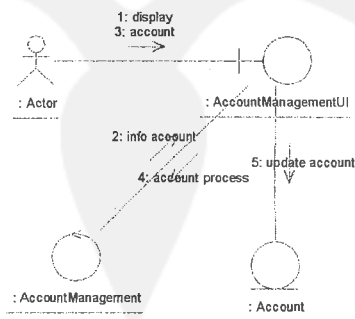
Gambar 16. Analysis Collaboration Diagram : Use Case Scanning Active Client

4.2.3 Analysis Collaboration Diagram : Use Case Clean Data



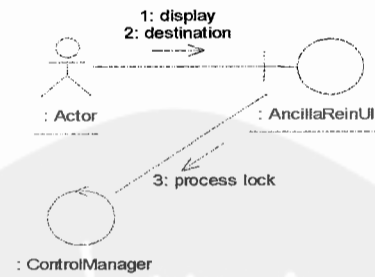
Gambar 17. Analysis Collaboration Diagram : Use Case Clean Data

4.2.4 Analysis Collaboration Diagram : Use Case Account Set



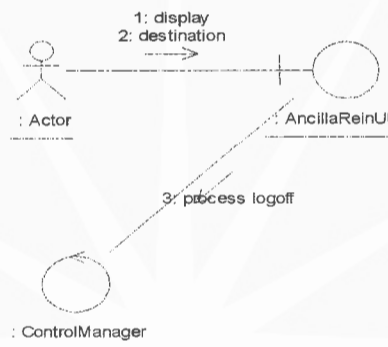
Gambar 18. Analysis Collaboration Diagram : Use Case Account Set

4.2.5 Analysis Collaboration Diagram : Use Case Lock



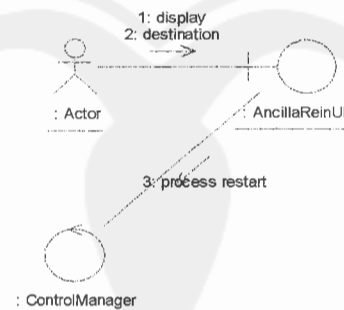
Gambar 19 Analysis Collaboration Diagram : Use Case Lock

4.2.6 Analysis Collaboration Diagram : Use Case LogOff



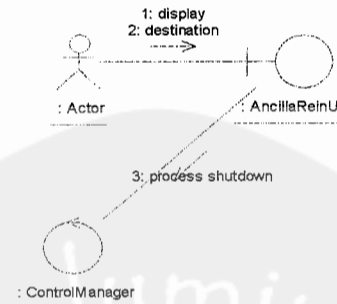
Gambar 20. Analysis Collaboration Diagram : Use Case LogOff

4.2.7 Analysis Collaboration Diagram : Use Case Restart



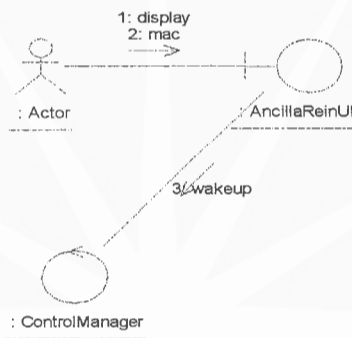
Gambar 21. Analysis Collaboration Diagram : Use Case Restart

4.2.8 Analysis Collaboration Diagram : Use Case Shutdown



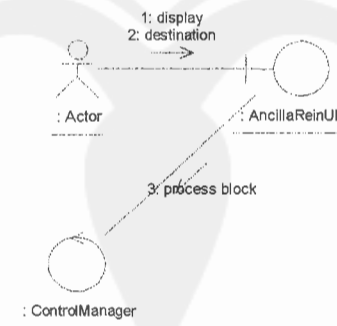
Gambar 22. Analysis Collaboration Diagram : Use Case Shutdown

4.2.9 Analysis Collaboration Diagram : Use Case Wake Up



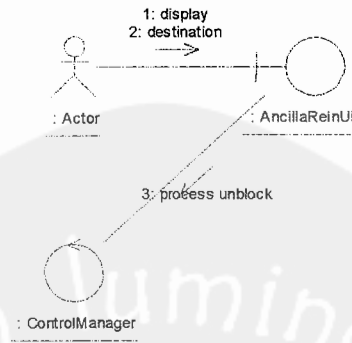
Gambar 23. Analysis Collaboration Diagram : Use Case Wake Up

4.2.10 Analysis Collaboration Diagram : Use Case Block Input



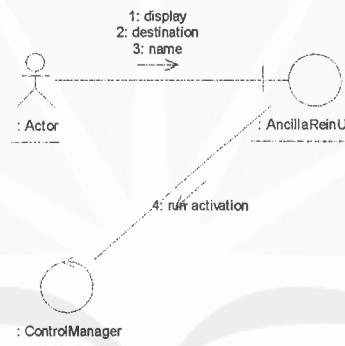
Gambar 24. Analysis Collaboration Diagram : Use Case Block Input

4.2.11 Analysis Collaboration Diagram : Use Case Unblock Input



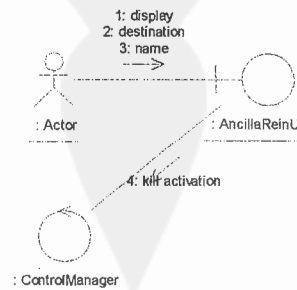
Gambar 25. Analysis Collaboration Diagram : Use Case Unblock Input

4.2.12 Analysis Collaboration Diagram : Use Case Run Process



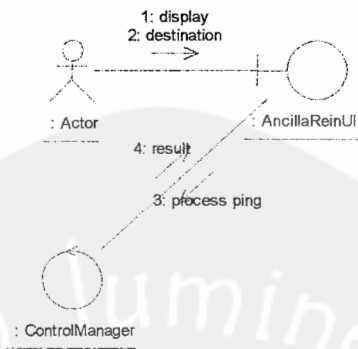
Gambar 26. Analysis Collaboration Diagram : Use Case Run Process

4.2.13 Analysis Collaboration Diagram : Use Case Kill Process



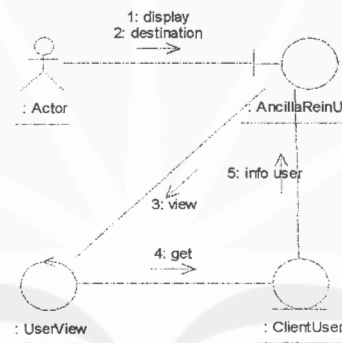
Gambar 27. Analysis Collaboration Diagram : Use Case Kill Process

4.2.14 Analysis Collaboration Diagram : Use Case Ping Process



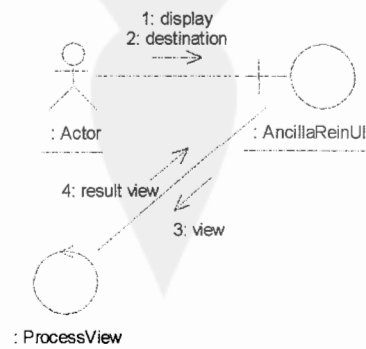
Gambar 28. Analysis Collaboration Diagram : Use Case Ping Process

4.2.15 Analysis Collaboration Diagram : Use Case View User



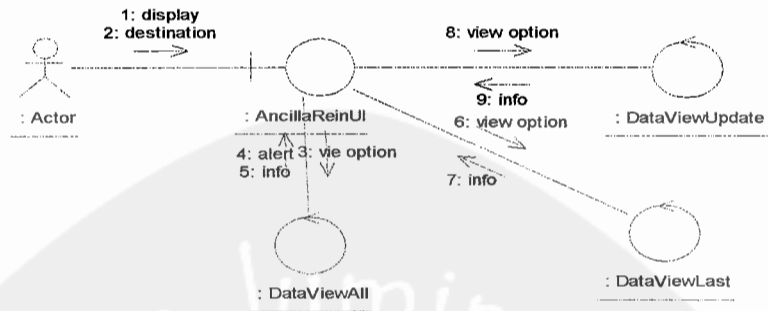
Gambar 29. Analysis Collaboration Diagram : Use Case View User

4.2.16 Analysis Collaboration Diagram : Use Case View Process



Gambar 30. Analysis Collaboration Diagram : Use Case View Process

4.2.17 Analysis Collaboration Diagram : Use Case View Data



Gambar 31. Analysis Collaboration Diagram : Use Case View Data

Apendiks A : Daftar Istilah dan Singkatan

Keyword/ Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
Ancilla 1.0	Perangkat lunak yang berfungsi untuk melakukan pengendalian dan pemantauan komputer secara terpusat
User	Pengguna yang dapat berinteraksi dengan perangkat lunak.
ERD	<i>Entity Relationship Diagram</i> merupakan teknis grafis/diagram yang menggambarkan objek dan hubungan antar objek.
Administrator	Pengguna yang memiliki full Access terhadap pengelolaan data-data yang ada pada sistem.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan
Client	Komputer yang berfungsi meminta layanan server yang terhubung melalui jaringan.
.NET Framework	adalah lingkungan untuk membangun, menyebarkan dan menjalankan perangkat lunak dan layanan berbasis web.

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Ancilla 1.0

Untuk :


Tugas akhir

Disusun oleh:

Aryanto Adi Nugroho

02 07 03605

Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
	Universitas Atma Jaya Yogyakarta	DPPL - Ancilla		1/43
		Revisi		26/06/2007

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E
Ditulis oleh	AR					
Diperiksa oleh	KSN KA					
Disetujui oleh	KSN KA					

NOTASI DOKUMEN

Notasi yang digunakan dalam dokumen ini adalah sebagai berikut :

- Teks normal ditulis dalam font Courier New 10 pt, plain.
- Teks yang ditulis dalam font **bold** merupakan teks yang mengacu pada bab, sub-bab, gambar, atau tabel dalam dokumen ini.
- Teks yang ditulis dalam font 8 pt Courier New merupakan teks yang mengacu pada model, diagram, atau file yang disebutkan dalam dokumen ini.



DAFTAR ISI

1	Pendahuluan	8
1.1	Tujuan.....	8
1.2	Lingkup Dokumen8	
1.3	Definisi, Akronim, dan Singkatan.....	8
1.4	Referensi.....	8
1.5	Deskripsi Umum (Overview).....	8
2	Deskripsi Perancangan Arsitektural	9
2.1	Deployment Diagram.....	9
2.1.1	Node : Client	9
2.1.2	Node : Server	9
2.2	Design Class.....	9
2.2.1	Pengantar	9
2.2.2	<i>Package Dependencies</i>	10
2.2.3	<i>Package AncillaRein (Ancilla.AncillaRein)</i>	10
2.2.4	<i>Package AncillaRemote (Ancilla.AncillaRemote)</i>	
2.3	Realisasi <i>Use Case</i>	18
3	Deskripsi Perancangan Persistent Data	29
3.1	Basis Data.....	29
4	Deskripsi Perancangan Antarmuka	30
	Apendiks A : Daftar Istilah dan Singkatan	43

1 Pendahuluan

1.1 Tujuan

Dokumen ini menjelaskan tentang Deskripsi Perancangan Perangkat Lunak. Untuk penamaan dokumen ini selanjutnya akan digunakan istilah DPPL. Dokumen ini merupakan dokumen deskripsi perancangan perangkat lunak Ancilla, versi 1.0, yang akan dibangun. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis untuk pengembangan perangkat lunak Ancilla.

1.2 Lingkup Dokumen

Dokumen DPPL ini menyediakan deskripsi lengkap perancangan perangkat lunak Ancilla versi 1.0. Perancangan ini merupakan arsitektur sistem yang dijelaskan melalui perancangan *class/modul*, *detail* operasi apa yang akan dilakukan oleh masing-masing *class/modul*, dan *layout* basis data.

1.3 Definisi, Akronim, dan Singkatan

Untuk definisi istilah dan singkatan yang digunakan dalam dokumen ini dapat mengacu pada **Apendiks A :Daftar Istilah dan Singkatan**.

1.4 Referensi

Referensi yang digunakan dalam pembuatan dokumen ini adalah :

- Nugroho, Aryanto Adi. *Spesifikasi Kebutuhan Perangkat Lunak Ancilla 1.0*. Universitas Atma Jaya Yogyakarta. 2007.
- Martin Fowler, Kendall Scott. *UML Distilled - Second Edition*. Adisson Wesley. 1999.

1.5 Deskripsi Umum (Overview)

Dokumen ini terdiri dari empat bab. Bab pertama adalah **Pendahuluan**, yang berisi deskripsi dokumen. Bab kedua adalah **Deskripsi Perancangan Arsitektural**, yang berisi deskripsi arsitektur sistem. Bab ketiga adalah **Deskripsi Perancangan Persistent Data**, yang berisi deskripsi data-data yang akan disimpan pada *persistent storage*. Bab keempat adalah **Deskripsi**

Perancangan Antarmuka, yang berisi deskripsi rancangan GUI yang digunakan sistem untuk berinteraksi dengan user.

2 Deskripsi Perancangan Arsitektural

2.1 Deployment Diagram

Deployment diagram ini dibuat untuk menunjukkan semua node (lihat **Apendiks A : Daftar Istilah dan Singkatan**) pada sistem, hubungan diantara mereka, dan proses yang akan dijalankan di masing-masing node.



Gambar 2.1 *Deployment Diagram Ancilla*

2.1.1 Node : Client

Client merupakan komputer yang akan dipantau dan dikendalikan. Proses yang ada didalamnya adalah :

1. Membuat *socket*, kemudian melakukan *listening* perintah.
2. Apabila ada perintah datang maka *AncillaRemote* akan mengeksekusi perintah yang diberikan.

2.1.2 Node : Server

Server merupakan komputer yang digunakan untuk melakukan pemantauan dan pengendalian. Proses yang ada didalamnya adalah :

1. *AncillaRein*, merupakan program utama yang akan melakukan pemantauan dan pengendalian. Program ini membuat suatu *socket* untuk mengirimkan perintah sekaligus mendengarkan dan me-*record* respon yang diberikan.
2. *Access Database*, merupakan database yang akan digunakan oleh *Ancilla*.

2.2 Design Class

2.2.1 Pengantar

Nama *class* yang digunakan dalam *design class* adalah nama *class* yang valid, termasuk nama *package* yang dimiliki. Untuk *class-class* yang berasal dari *framework.Net* juga digunakan nama *class* dengan *package* lengkap, misalnya *System.IO.File*. Untuk

penjelasan tipe data yang utuh dapat dilihat pada bagian deskripsi *class*, sedangkan gambar *design class* tidak akan menggunakan nama *package* yang lengkap.

Stereotype yang digunakan dalam *design class* adalah :

- `<< boundary >>`
Boundary class merupakan *class* yang berfungsi untuk menghubungkan sistem dengan *user* di luar sistem.
- `<< control >>`
Control class adalah suatu *class* yang objek-nya melakukan interaksi antar sekelompok objek lain. *Control class* biasanya memiliki karakteristik yang spesifik untuk satu *use case*, dan objek *class* ini biasanya hanya aktif pada *realisasi use case*.
- `<< entity >>`
Entity class adalah *class* yang bersifat pasif, dalam arti *class* tersebut tidak memulai interaksi dengan *class* lain. *Entity class* ini biasanya merepresentasikan suatu objek yang disimpan dalam *persistent storage*.

Untuk hirarki *class design* berdasarkan *package*, dapat dilihat pada **Apendiks B : Hirarki Design Class berdasarkan Package**.

2.2.2 Package Dependencies

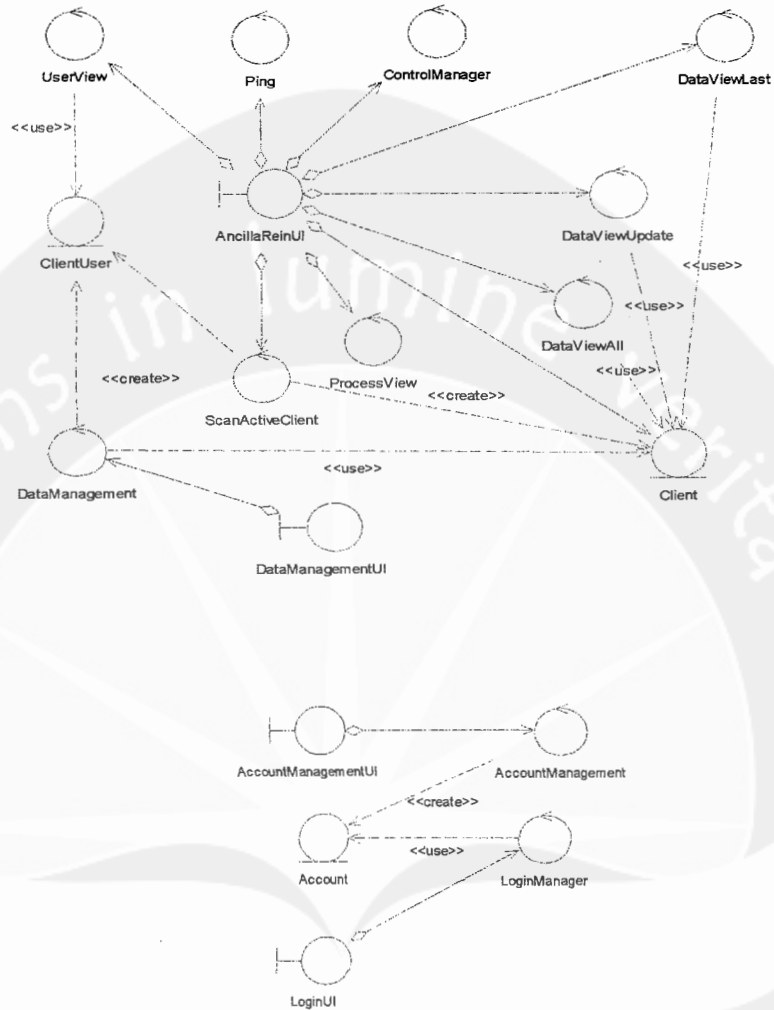


Gambar 2.2 Package Dependencies Ancilla

2.2.3 Package AncillaRein (Ancilla.AncillaRein)

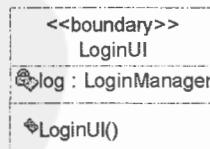
Package ini menyediakan *class-class* yang digunakan oleh program untuk mengelola hal-hal yang berhubungan dengan pemantauan dan pengendalian, antara lain : menampilkan *monitoring client* (UC-Ancilla-02), *controlling client* (UC-Ancilla-03), *account management* (UC-Ancilla-04), *data management* (UC-Ancilla-05) dan *scanning active client* (UC-Ancilla-06).

2.2.3.1 Class Diagram Package Ancilla.AncillaRein



Gambar 2.3 Class Diagram Package Ancilla.AncillaRein

2.2.3.2 Class Ancilla.AncillaRein.LoginUI

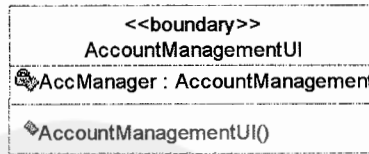


Gambar 2.4 Class Ancilla.AncillaUser.LoginUI

Deskripsi

Class ini merupakan GUI untuk use case Login (UC-Ancilla-01).

2.2.3.3 Class Ancilla.AncillaRein.AccountManagementUI

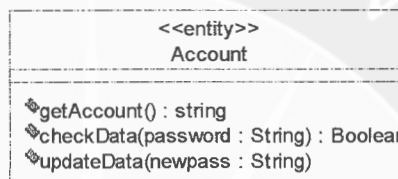


Gambar 2.5 Class Ancilla.AncillaUser.AccountManagement

Deskripsi

Class ini merupakan GUI untuk use case Data Management (UC-Ancilla-04).

2.2.3.4 Class Ancilla.AncillaRein.Account

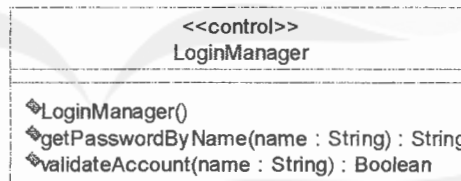


Gambar 2.6 Class Ancilla.AncillaUser.Account

Deskripsi

Class ini merepresentasikan data account pada sistem Ancilla.

2.2.3.5 Class Ancilla.AncillaRein.LoginManager

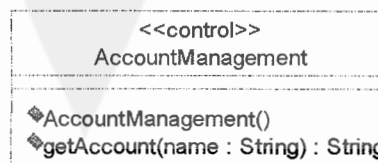


Gambar 2.7 Class Ancilla.AncillaUser.LoginManager

Deskripsi

Class yang berperan sebagai control class untuk aksi yang berhubungan dengan proses login.

2.2.3.6 Class Ancilla.AncillaRein.AccountManager

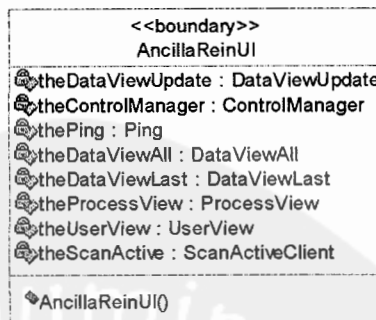


Gambar 2.8 Class Ancilla.AncillaUser.AccountManager

Deskripsi

Class yang berperan sebagai control class untuk aksi yang berhubungan dengan management account.

2.2.3.7 Class Ancilla.AncillaRein.AncillaReinUI

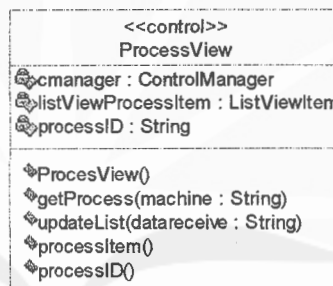


Gambar 2.10 Class Ancilla.AncillaRein.AncillaReinUI

Deskripsi

Class ini merupakan GUI untuk use case *Monitoring Client* (UC-Ancilla-02), *Controlling Client* (UC-Ancilla-03), *Scanning Active Client* (UC-Ancilla).

2.2.3.8 Class Ancilla.AncillaRein.ProcessView

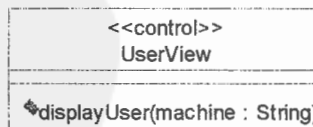


Gambar 2.11 Class Ancilla.AncillaRein.ProcessView

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan menampilkan proses yang ada di *client*.

2.2.3.9 Class Ancilla.AncillaRein.UserView

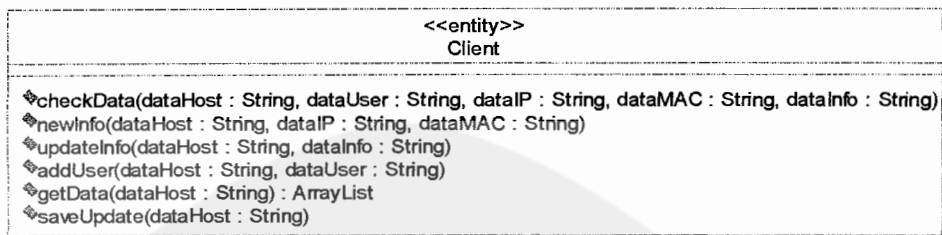


Gambar 2.12 Class Ancilla.AncillaRein.UserView

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan menampilkan *user* yang sedang/pernah *log* pada suatu *client*.

2.2.3.10 Class *Ancilla.AncillaRein.Client*



Gambar 2.13 Class *Ancilla.AncillaRein.Client*

Deskripsi

Class ini merepresentasikan data *client* pada sistem *monitoring*.

2.2.3.11 Class *Ancilla.AncillaRein.ClientUser*

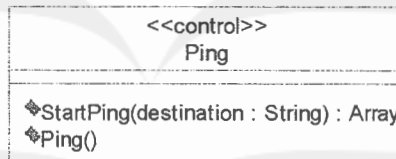


Gambar 2.14 Class *Ancilla.AncillaRein.ClientUser*

Deskripsi

Class ini merepresentasikan data *client user* pada sistem *monitoring*.

2.2.3.12 Class *Ancilla.AncillaRein.Ping*

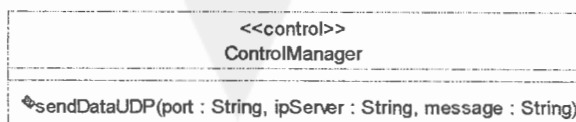


Gambar 2.15 Class *Ancilla.AncillaRein.Ping*

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan proses pengecekan konektifitas jaringan.

2.2.3.13 Class *Ancilla.AncillaRein.ControlManager*

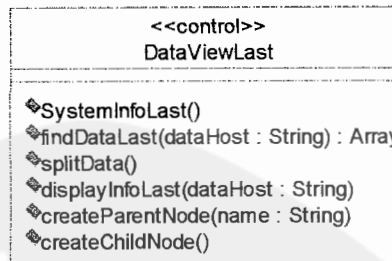


Gambar 2.16 Class *Ancilla.AncillaRein.ControlManager*

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan proses *controlling*, mengirimkan paket data UDP.

2.2.3.14 Class *Ancilla.AncillaRein.DataViewLast*

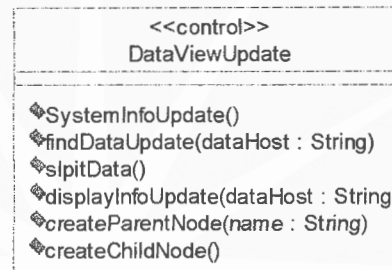


Gambar 2.17 Class *Ancilla.AncillaRein.DataViewLast*

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan menampilkan data konfigurasi terakhir suatu *client*.

2.2.3.15 Class *Ancilla.AncillaRein.DataViewUpdate*

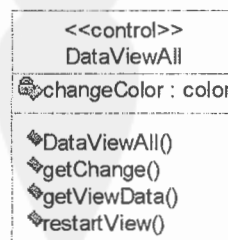


Gambar 2.18 Class *Ancilla.AncillaRein.DataViewUpdate*

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan menampilkan data konfigurasi terbaru dari *client*.

2.2.3.16 Class *Ancilla.AncillaRein.DataViewAll*

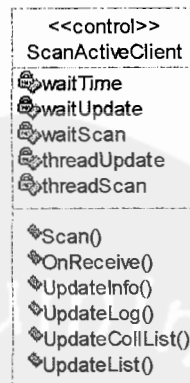


Gambar 2.19 Class *Ancilla.AncillaRein.DataViewAll*

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan menampilkan data semua *client*.

2.2.3.17 Class *Ancilla.AncillaRein.ScanActiveClient*



Gambar 2.20 Class *Ancilla.AncillaRein.ScanActiveClient*

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan menampilkan *client* yang aktif di jaringan.

2.2.3.18 Class *Ancilla.AncillaRein.DataManagement*

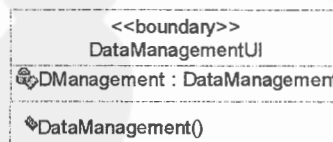


Gambar 2.121 Class *Ancilla.AncillaRein.DataManagement*

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan manajemen data.

2.2.3.19 Class *Ancilla.AncillaRein.DataManagementUI*



Gambar 2.22 Class *Ancilla.AncillaRein.DataManagementUI*

Deskripsi

Class ini merupakan GUI untuk use case *Data Management* (UC-Ancilla-05).

2.2.4 Package AncillaREMOTE (Ancilla.AncillaREMOTE)

Package ini menyediakan class-class yang diperlukan untuk melakukan eksekusi dan mengirimkan informasi *client*.

2.2.4.3 Class Diagram Package Ancilla.AncillaREMOTE



Gambar 2.23 Class Diagram Package Ancilla.AncillaRemote

2.2.4.4 Class ANCILLA.AncillaREMOTE.ClientManager

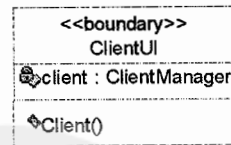


Gambar 2.24 Class Ancilla.AncillaRemote.ClientManager

Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan semua proses respon permintaan eksekusi.

2.2.4.5 Class *ANCILLA.AncillaREMOTE.ClientUI*



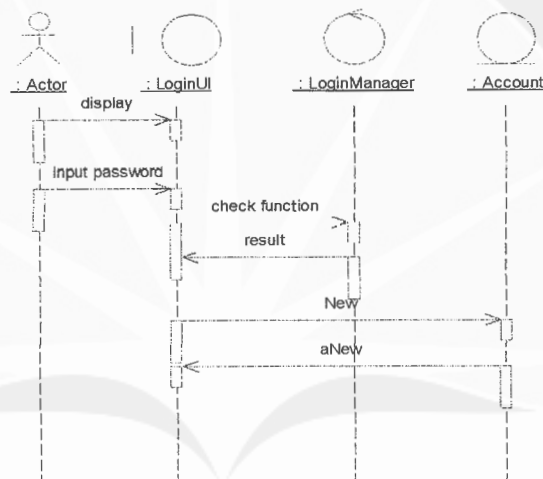
Gambar 2.25 Class *Ancilla.AncillaRemote.ClientUI*

Deskripsi

Class ini merupakan *interface* yang berjalan di setiap *client*.

2.3 Realisasi Use Case

2.3.3 Use Case : Login Process



Gambar 2.26

Design Sequence Diagram : Use Case Login

Flow of events :

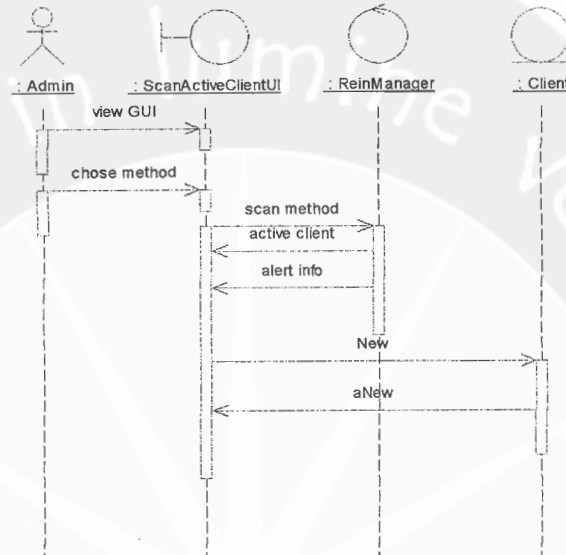
1. Aktor menampilkan antarmuka untuk login, yaitu *boundary class LoginUI*.
2. Aktor memasukkan *password*. *Password* ini merupakan rangkaian karakter dengan panjang 5-20 karakter, dan bersifat *case-sensitive*.

Kemudian administrator memicu sistem untuk melakukan validasi melalui *LoginUI* (*boundary class*). *LoginUI* membentuk *aNew*, yang merupakan instance baru *class User*. *LoginUI* melakukan validasi terhadap *password* yang diinputkan yang merupakan atribut dari *aNew*. *LoginManager* memvalidasi *aNew* ini dengan memanggil *check function method* milik *LoginManager*, yang akan memeriksa di basis data. Sistem akan

membandingkan *username password* tersebut dari basis data dengan *password input user* dari *aNewUser*. User akan dapat masuk ke sistem bila *password-nya* sesuai.

3. Objek *aNew* digunakan sebagai *account* aktif pada *AncillaRein*.

2.3.4 Use Case : Scanning Active Client



Gambar 2.27

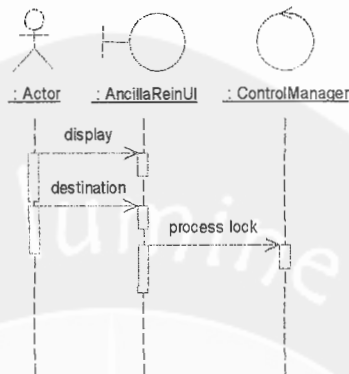
Design Sequence Diagram : Use Case Scanning Active Client

Flow of events :

1. Aktor menampilkan antarmuka untuk *Scanning Active Client*, yaitu *boundary class LoginUI*.
2. Aktor memilih metode *scanning*, antara manual atau otomatis. Untuk metode manual *scanning* dilakukan dengan menunggu aktor menekan tombol *scan*. Sedangkan *scanning* otomatis dilakukan oleh sistem dalam jangka waktu tertentu. Proses ini akan memicu *ScanActiveClientUI* memanggil *REINmanager control* untuk melakukan pengiriman paket pengecekan *client*. *REINmanager* akan melakukan filtering data *client* melakukan pengecekan ke *entity client*. dan membentuk *active Client*, yang merupakan instance baru *client* yang aktif dalam jaringan beserta *alert info* apabila ada perubahan konfigurasi.
3. Objek *aNew* digunakan sebagai *client* aktif pada *AncillaRein*.

2.3.5 Use Case : Controlling Client

2.3.5.3 Lock



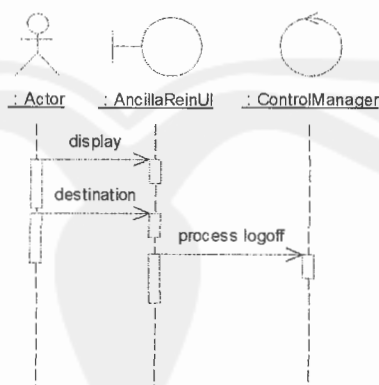
Gambar 2.28

Design Sequence Diagram : Use Case Controlling Client-Lock

Flow of events :

1. Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.
2. Aktor memasukkan *destination*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *lock* ke *client*.

2.3.5.4 LogOff



Gambar 2.29

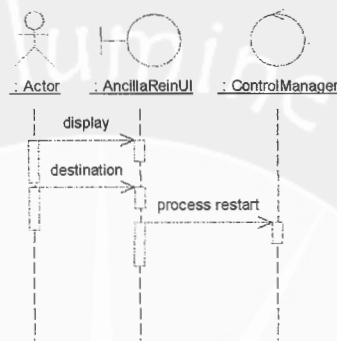
Design Sequence Diagram : Use Case Controlling Client-LogOff

Flow of events :

1. Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.

- Aktor memasukkan *destination*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *LogOff* ke *client*.

2.3.5.5 Restart



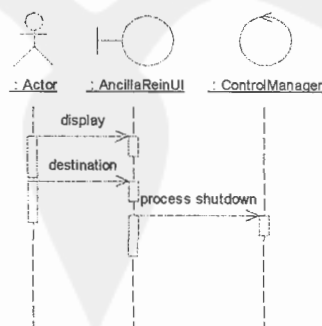
Gambar 2.30

Design Sequence Diagram : Use Case Controlling Client-Restart

Flow of events :

- Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.
- Aktor memasukkan *destination*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *restart* ke *client*.

2.3.5.6 Shutdown



Gambar 2.31

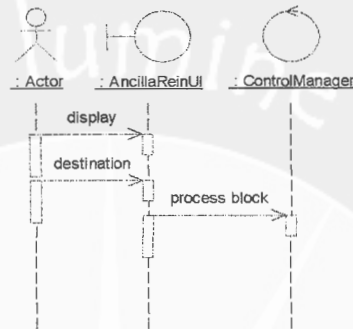
Design Sequence Diagram : Use Case Controlling Client-Shutdown

Flow of events :

- Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.

- Aktor memasukkan *destination*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *shutdown* ke *client*.

2.3.5.7 Block Input



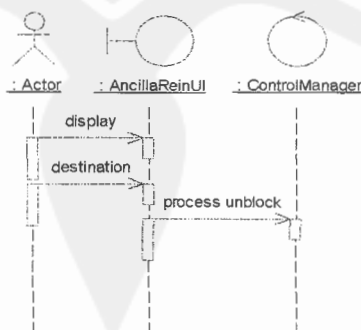
Gambar 2.32

Design Sequence Diagram : Use Case Controlling Client-Block Input

Flow of events :

- Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.
- Aktor memasukkan *destination*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *Block Input* ke *client*.

2.3.5.8 Unblock Input



Gambar 2.33

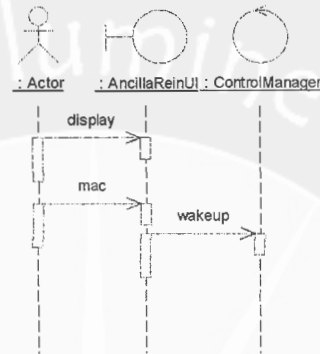
Design Sequence Diagram : Use Case Controlling Client-Unblock Input

Flow of events :

- Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.

- Aktor memasukkan *destination*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *Unblock Input* ke *client*.

2.3.5.9 Wake up



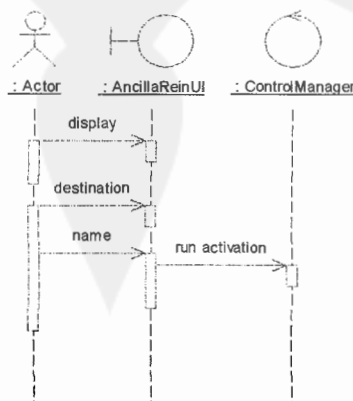
Gambar 2.34

Design Sequence Diagram : Use Case Controlling Client-WakeUp

Flow of events :

- Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.
- Aktor memasukkan *destination*(berupa MAC Address). Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *WakeUP* ke *client*.

2.3.5.10 Run Process



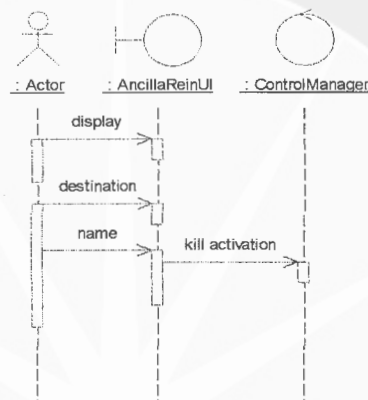
Gambar 2.35

Design Sequence Diagram : Use Case Controlling Client-Run Process

Flow of events :

1. Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.
2. Aktor memasukkan *destination* dan *file name*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI (boundary class)*. *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *run* ke *client*.

2.3.5.11 Kill Process



Gambar 2.36

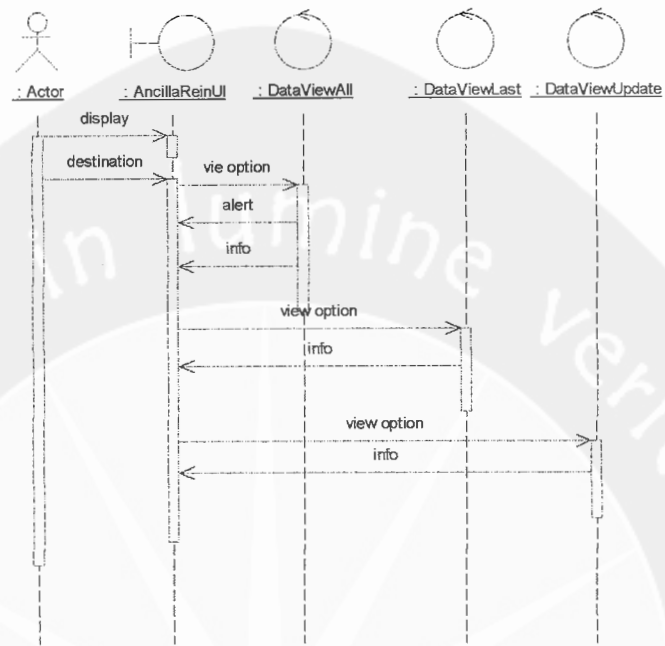
Design Sequence Diagram : Use Case Controlling Client-Kill Process

Flow of events :

1. Aktor menampilkan antarmuka untuk *Controlling Client*, yaitu *boundary class AncillaReinUI*.
2. Aktor memasukkan *destination* dan *file name*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AncillaReinUI (boundary class)*. *AncillaReinUI* menjalankan *ControlManager*. *ControlManager* melakukan pengiriman paket untuk melakukan perintah *kill process* ke *client*.

2.3.6 Use Case : Monitoring Client

2.3.6.3 View Data



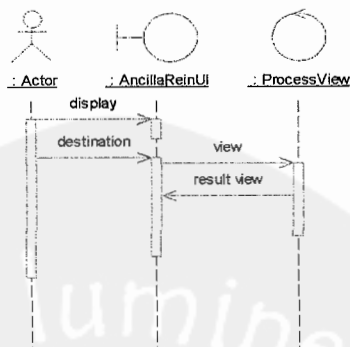
Gambar 2.37

Design Sequence Diagram : Use Case Monitoring Client-View Data

Flow of events :

1. Aktor menampilkan antarmuka untuk *monitoring client*, yaitu *boundary class AncillaReinUI*.
2. Aktor memasukkan *destination* parameter. Aktor memicu sistem untuk melakukan proses melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* akan melakukan proses control sesuai mekanisme yang di inginkan. *DataViewAll* memvalidasi untuk menampilkan seluruh data dari tiap *client*. *DataViewLast* memvalidasi untuk menampilkan seluruh data pada suatu *client*. *DataViewUpdate* memvalidasi untuk menampilkan seluruh data (apabila ada perubahan konfigurasi) pada suatu *client*.
3. Objek *info* digunakan sebagai data *client* yang akan ditampilkan di *AncillaUI*.

2.3.6.4 View Process



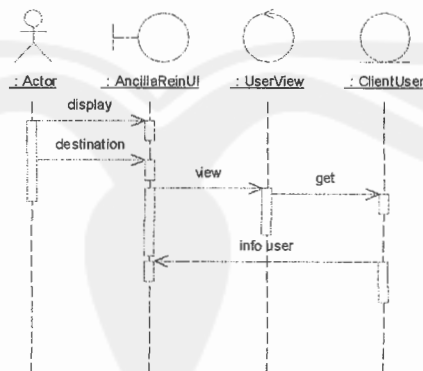
Gambar 2.38

Design Sequence Diagram : Use Case Monitoring Client-View Process

Flow of events :

1. Aktor menampilkan antarmuka untuk *monitoring client*, yaitu *boundary class AncillaReinUI*.
2. Aktor memasukkan *destination* parameter. Aktor memicu sistem untuk melakukan *process* melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* akan melakukan proses *control* sesuai mekanisme yang di inginkan. *ProcessView* memvalidasi untuk menampilkan seluruh proses pada suatu *client*.
3. Objek *resultview* digunakan sebagai data proses *client* yang akan ditampilkan di *AncillaUI*.

2.3.6.5 View User



Gambar 2.39

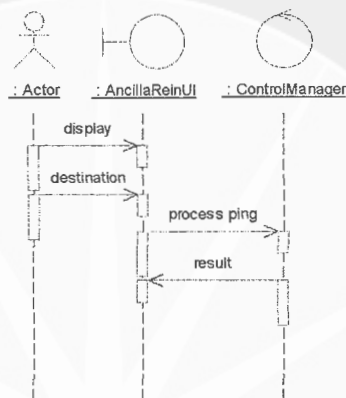
Design Sequence Diagram : Use Case Monitoring Client-View User

Flow of events :

1. Aktor menampilkan antarmuka untuk *monitoring client*, yaitu *boundary class AncillaReinUI*.

2. Aktor memasukkan *destination* parameter. Aktor memicu sistem untuk melakukan *process* melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* akan melakukan proses *control* sesuai mekanisme yang di inginkan. *UIView* memvalidasi untuk menampilkan seluruh user (yang pernah/sedang log) pada suatu *client*.
3. Objek *info user* digunakan sebagai data user pada *client* yang akan ditampilkan di *AncillaUI*.

2.3.6.6 Ping



Gambar 2.40

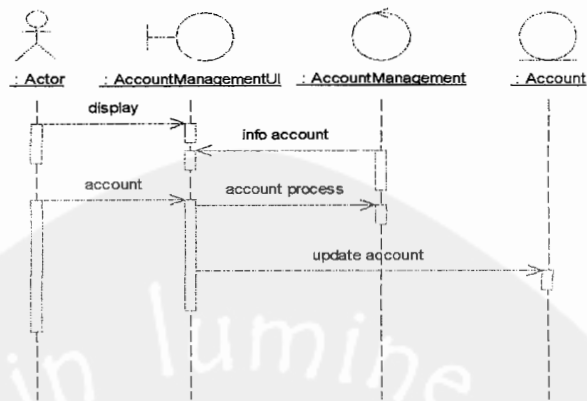
Design Sequence Diagram : Use Case Monitoring Client-Ping

Flow of events :

1. Aktor menampilkan antarmuka untuk *monitoring client*, yaitu *boundary class AncillaReinUI*.
2. Aktor memasukkan *destination* parameter. Aktor memicu sistem untuk melakukan *process* melalui *AncillaReinUI* (*boundary class*). *AncillaReinUI* akan melakukan proses *control* sesuai mekanisme yang di inginkan. *ControlManager* memvalidasi untuk melakukan proses *ping* pada suatu *client*. Proses ping dilakukan dengan mengirimkan paket data tertentu.
3. Objek *result ping* digunakan sebagai data balasan dari *client* yang akan ditampilkan di *AncillaUI*.

2.3.7 Use Case : Account Management

2.3.7.3 Change Account



Gambar 2.41

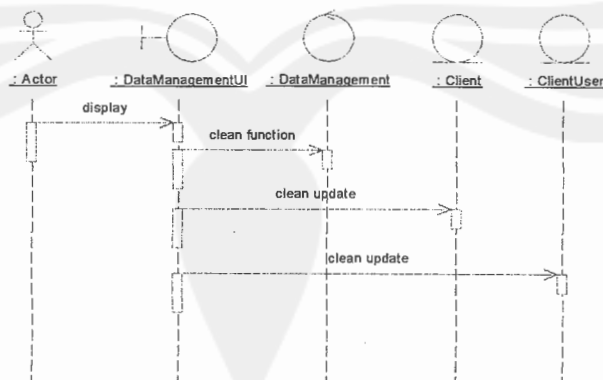
Design Sequence Diagram : Use Case Account Management-ChangeAccount

Flow of events :

1. Aktor menampilkan antarmuka untuk *Account Management*, yaitu *boundary class AccountManagementUI*. *AccountManagement* akan menampilkan *info account*.
2. Aktor memasukkan data *account* yang baru. Kemudian, administrator memicu sistem untuk melakukan proses melalui *AccountManagementUI* (*boundary class*). *AccountManagementUI* menjalankan *AccountManagement*.
3. Objek *updateAccount* digunakan sebagai *account* baru pada *Account*.

2.3.8 Use Case : Data Management

2.3.5.1 Clean Data



Gambar 2.42

Design Sequence Diagram : Use Case Data Management-Clean Data

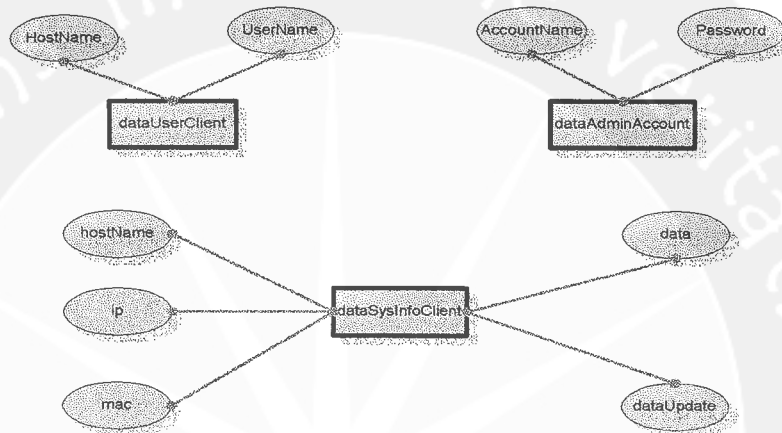
Flow of events :

1. Aktor menampilkan antarmuka untuk *Data Management*, yaitu *boundary class DataManagementUI*.

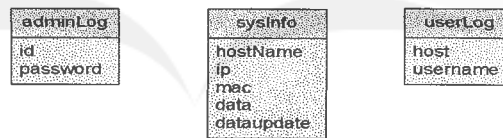
2. Aktor memasukkan pilihan aksi (*choose function*) antara *clean data user* atau *clean data client*. Kemudian, administrator memicu sistem untuk melakukan proses melalui *DataManagementUI* (*boundary class*). *DataManagementUI* menjalankan *DataManagement*.
3. Objek *clean update* digunakan sebagai *entity* baru pada *Client* dan *ClientUser*.

3. Deskripsi Perancangan Persistent Data

3.2 Basis Data



Gambar 3.1 Entity Relationship Diagram Ancilla



Gambar 3.2 Conceptual Data Model

adminLog		sysInfo	
id	Varchar(50)	hostName	Varchar(50)
password	Varchar(10)	ip	Varchar(50)
		mac	Varchar(50)
		data	Varchar(50)
		dataupdate	Varchar(50)
userLog			
host	Varchar(20)		
UserName	Varchar(20)		

Gambar 3.3 Physical Data Model

Tabel- tabel pada basis data disesuaikan dengan **Gambar 3.1**, **Gambar 3.2** dan **Gambar 3.3**. DBMS yang akan digunakan adalah Ms Access 2003.

3.2.3 Tabel Data *System Info Client*

Tabel ini merepresentasikan entitas User.

Tabel 3.1 Tabel *sysInfo*

Field	Tipe Data	Deskripsi
hostName	Varchar(50)	Host Name dari client yang akan dipantau dan dikendalikan. Primary Key.
ip	Varchar(50)	ip host
mac	Varchar(50)	mac host
data	Varchar(50)	Data system info tiap host
dataupdate	Varchar(50)	Data system info tiap host apabila ada perubahan konfigurasi

3.2.4 Tabel Data *Admin Account*

Tabel ini merepresentasikan entitas Admin Account.

Tabel 3.2 Tabel *userLog*

Field	Tipe Data	Deskripsi
id	Varchar(50)	id dari <i>account</i> , Ancilla hanya menyediakan 1 default <i>account</i> yaitu <i>administrator</i> .
password	Varchar(10)	Password dari <i>account</i>

3.2.5 Tabel Data *User Client*

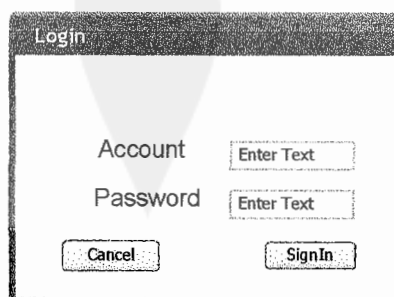
Tabel ini merepresentasikan entitas User Log.

Tabel 3.2 Tabel *userLog*

Field	Tipe Data	Deskripsi
host	Varchar(20)	Host Name dari client.
UserName	Varchar(20)	User yang melakukan log ke host tersebut.

4. Deskripsi Perancangan Antarmuka

4.2 Use Case : Login Process



Gambar 4.1
Rancangan Antarmuka *Use Case Login Process*

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada **Class Ancilla.AncillaRein.LoginUI**. Antarmuka ini digunakan pada use case *Login Process* (UC-Ancilla-01).
- TextBox atas digunakan untuk menampilkan *username*, *username default* adalah administrator.
- TextBox bawah digunakan untuk menerima masukan *password*.
- Tombol *SignIn* digunakan untuk aksi *login user*.

Event

- Login

Urutan aksi yang terjadi :

1. User memasukkan password pada TextBox bawah.
2. User menekan tombol Login.
3. Sistem memeriksa di basis data dengan mengambil informasi password.

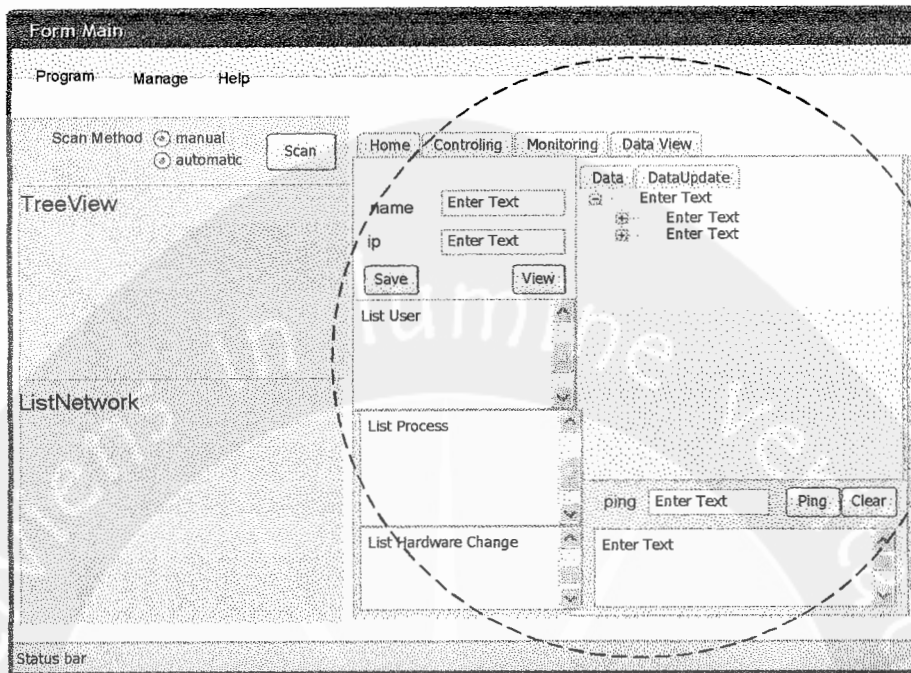
Statement SQL yang digunakan untuk mengambil data dari basis data adalah :

```
SELECT password
FROM tabel_user
WHERE username = [param1]
```

Dengan **[param1]** adalah *username default* (administrator).

Jika data sesuai teruskan ke langkah selanjutnya. Jika tidak sistem menampilkan pesan kesalahan user. Sistem membandingkan password dari query user yang diperoleh pada langkah 3 dengan password input user dari langkah 1.

4.3 Use Case : *Monitoring Client*



Gambar 4.2

Rancangan Antarmuka *Use Case Monitoring Client 1*

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada **Class Ancilla.AncillaRein.AncillaReinUI**. Antarmuka ini digunakan pada use case *Monitoring Client* (UC-Ancilla-02).
- Textbox atas untuk memasukkan nama host yang akan dilihat datanya.
- Textbox bawah untuk memasukkan alamat ip yang akan dilihat datanya.
- Tombol save untuk melakukan save perubahan konfigurasi.
- Tombol view untuk melakukan display konfigurasi dan perubahannya, jika ada.
- List user adalah list untuk menampilkan user yang sedang log maupun yang pernah log sebelumnya.
- List process adalah list untuk menampilkan seluruh proses yang sedang run di suatu host.
- List hardware change adalah list untuk menampilkan perubahan data konfigurasi.

- Tab data adalah tab yang menampilkan treeview konfigurasi.
- Tab data update adalah tab yang menampilkan treeview perubahan konfigurasi.
- Textbox ping adalah textbox yang digunakan untuk memasukkan destination ping.
- Tombol ping adalah tombol yang digunakan untuk memulai *action* proses ping.
- Tombol clear adalah tombol yang digunakan untuk mengkosongi textbox ping dan list ping.
- List ping adalah list yang digunakan untuk menampilkan result ping.

Event

- *View Data*

Urutan aksi yang terjadi :

1. User memasukkan *host destination* dan alamat ip.
2. User menekan tombol view.
3. Sistem memeriksa di basis data dengan mengambil informasi user.

Statement SQL yang digunakan untuk mengambil data dari basis data adalah :

```
SELECT *
FROM tabel sysInfo
WHERE host = hostName;
```

Jika data sesuai teruskan ke langkah selanjutnya. Jika tidak sistem menampilkan pesan kesalahan user.

4. Sistem menampilkan data konfigurasi.

- *View Data Update*

Urutan aksi :

Langkah awal sama dengan View Data, perubahan konfigurasi (jika ada) akan ditampilkan secara otomatis.

- *View User*

Urutan aksi :

1. Langkah awal sama dengan View Data.

2. Sistem memeriksa di basis data dengan mengambil informasi user.

Statement SQL yang digunakan untuk mengambil data dari basis data adalah :

```
SELECT *  
FROM tabel userLog  
WHERE host = hostName;
```

Jika data sesuai teruskan ke langkah selanjutnya.

Jika tidak sistem menampilkan pesan kesalahan user.

3. Data proses ditampilkan.

- *View Process*

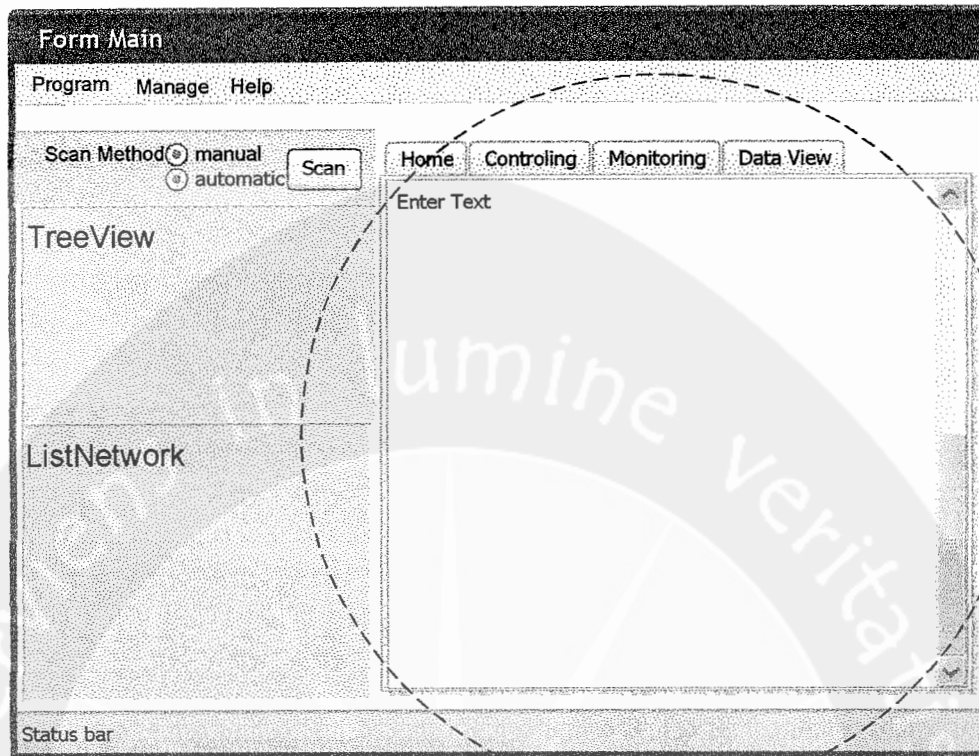
Urutan aksi :

1. Langkah awal sama dengan *View Data*, sistem akan mengirimkan paket ke *client* dan mem-filter data proses yang diterima.
2. Data proses ditampilkan.

- *Ping*

Urutan aksi :

1. User memasukkan destination
2. User menekan tombol ping.
3. Sistem mengirimkan paket data dan menampilkan respon di list ping.



Gambar 4.3

Rancangan Antarmuka Use Case Monitoring Client 2

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada **Class Ancilla.AncillaRein.AncillaReinUI**. Antarmuka ini digunakan pada use case *Monitoring Client* (UC-Ancilla-02).
- List Data All menampilkan semua data dari seluruh client, apabila ada perubahan konfigurasi tampilan informasi pada client tersebut memiliki behavior khusus (memiliki warna berbeda).

Event

- View Data All

Urutan aksi :

1. User mengaktifkan tab view data all
2. Sistem memeriksa di basis data dengan mengambil informasi client.

Statement SQL yang digunakan untuk mengambil data dari basis data adalah :

SELECT *

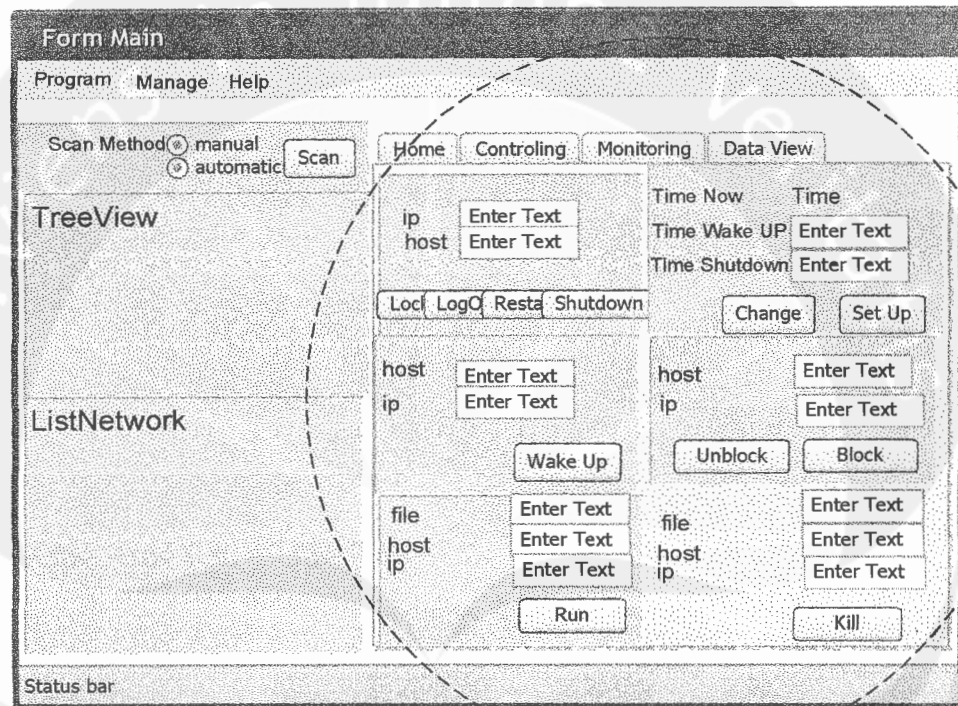
FROM tabel sysInfo

WHERE host = hostName;

Jika data sesuai teruskan ke langkah selanjutnya.
Jika tidak sistem menampilkan pesan kesalahan user.

3. Sistem menampilkan seluruh data.

4.4 Use Case : *Controlling Client*



Gambar 4.4

Rancangan Antarmuka Use Case *Controlling Client*

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada **Class Ancilla.AncillaRein.AncillaReinUI**. Antarmuka ini digunakan pada use case *Controlling Client* (UC-Ancilla-04).
- Textbox ip kiri atas memasukkan *destination ip*.
- Textbox host kiri atas memasukkan *destination ip*.
- Textbox ip kiri tengah memasukkan *destination ip*.
- Textbox host kiri tengah memasukkan *destination host*.
- Textbox ip kiri bawah memasukkan *destination ip*.

- Textbox host kiri bawah memasukkan *destination host*.
- Textbox file kiri bawah memasukkan nama *file*.
- Textbox time wakeup kanan atas variabel waktu wake up.
- Textbox time shutdown kanan atas variabel waktu shutdown.
- Textbox ip kanan tengah memasukkan *destination ip*.
- Textbox host kanan tengah memasukkan *destination host*.
- Textbox ip kiri atas memasukkan *destination ip*.
- Textbox host kiri atas memasukkan *destination host*.
- Textbox ip kanan bawah memasukkan *destination ip*.
- Textbox host kanan bawah memasukkan *destination host*.
- Textbox file kanan bawah memasukkan nama *file*.

Event

- *Lock*

Urutan langkah :

1. User memasukkan *destination host* dan *ip*.
2. User menekan tombol lock.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *LogOff*

Urutan langkah :

1. User memasukkan *destination host* dan *ip*.
2. User menekan tombol logOff.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *Restart*

Urutan langkah :

1. User memasukkan *destination host* dan *ip*.
2. User menekan tombol Restart.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *Shutdown*

Urutan langkah :

1. User memasukkan *destination host* dan *ip*.
2. User menekan tombol shutdown.

3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *Wake Up*

Urutan langkah :

1. User memasukkan *MAC address*.
2. User menekan tombol wake up.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *Run Process*

Urutan langkah :

1. User memasukkan *destination host, ip* dan nama *file*.
2. User menekan tombol run.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *Kill Process*

Urutan langkah :

1. User memasukkan *destination host, ip* dan nama *file*.
2. User menekan tombol kill.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *Set automode*

Urutan langkah :

1. User memasukkan variabel waktu.
2. Sistem akan melakukan eksekusi berdasar variabel waktu.

- *Block Input*

Urutan langkah :

1. User memasukkan *destination host* dan *ip*.
2. User menekan tombol block.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

- *Unblock input*

Urutan langkah :

1. User memasukkan *destination host* dan *ip*.

2. User menekan tombol unblock.
3. Sistem akan mengirimkan paket data ke client berisi perintah eksekusi.

4.5 Use Case : Account Management

Gambar 4.5

Rancangan Antarmuka Use Case Account Management

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada **Class Ancilla.AncillaRein.AccountManagementUI**. Antarmuka ini digunakan pada use case *Controlling Client* (UC-Ancilla-04).
- Textbox atas untuk menampilkan data password lama.
- Textbox tengah untuk memasukkan data password baru.
- Textbox bawah untuk memasukkan data password baru konfirmasi.

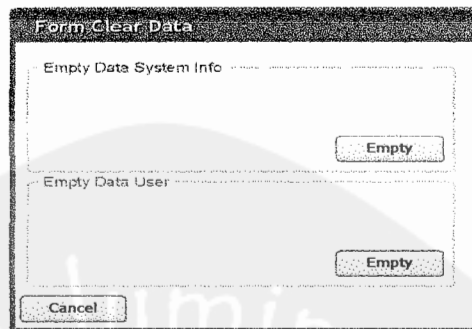
Event

- *Data Account Change*

Urutan langkah :

1. Sistem menampilkan password lama (jika ada).
2. User memasukkan data password baru.
3. User menekan tombol save.
4. Sistem memeriksa di basis data dengan mengambil informasi user dan melakukan update data.

4.6 Use Case :Data Management



Gambar 4.6

Rancangan Antarmuka Use Case Data Management

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada **Class Ancilla.AncillaRein.DataManagementUI**. Antarmuka ini digunakan pada use case *Controlling Client* (UC-Ancilla-05).
- Button empty atas untuk melakukan eksekusi clean data client.
- Button empty bawah untuk melakukan eksekusi clean data user.

Event

- *Clean Data Client*

Urutan langkah :

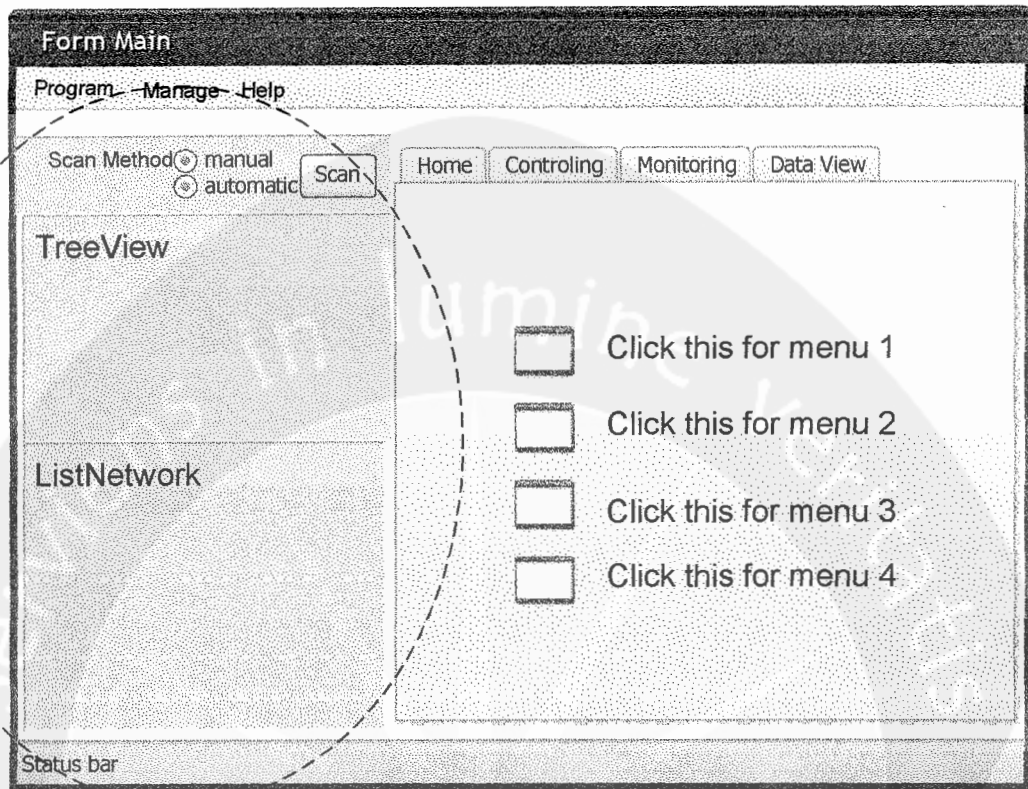
1. User menekan *button empty*.
2. Sistem akan melakukan pengosongan basis data tabel *sysInfo*.

- *Clean Data User*

Urutan langkah :

1. User menekan *button empty*.
2. Sistem akan melakukan pengosongan basis data tabel *userLog*.

4.7 Use Case : Scanning Active Client



Gambar 4.7

Rancangan Antarmuka Use Case Scanning Active Client

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada **Class Ancilla.AncillaRein.ScanActiveClientUI**. Antarmuka ini digunakan pada use case *Controlling Client* (UC-Ancilla-06).
- *Radiobutton* manual untuk memilih metode scan manual.
- *Radiobutton* otomatis untuk memilih metode scan otomatis.
- *Button scan* untuk memulai pengiriman paket data.
- *List Network* untuk menampilkan informasi log jaringan.
- *TreeView* untuk menampilkan informasi client yang aktif.

Event

- *Scan Manual*

Urutan langkah :

1. User memilih *radiobutton* manual.
2. User menekan *button Scan*.
3. Sistem akan melakukan pengiriman paket data.

4. Sistem akan memfilter respon dan menampilkan client yang aktif.

- *Scan* Otomatis

Urutan langkah :

1. User memilih *radiobutton* otomatis.
2. User menekan *button Scan*.
3. Sistem akan melakukan pengiriman paket data secara kontinyu.
4. Sistem akan memfilter respon dan menampilkan client yang aktif.



Apendiks A : Daftar Istilah dan Singkatan

Administrator adalah operator yang berhak mengakses seluruh fungsionalitas Ancilla.

C# adalah bahasa pemrograman yang sederhana, moderen, berorientasi objek dan type-safe, yang diturunkan dari C dan C++. C# menggabungkan produktifitas dari Visual Basic dan kemampuan dari C++.

Microsoft Visual Studio .NET adalah sebuah *platform* untuk membangun, menjalankan dan meningkatkan generasi lanjut dari aplikasi terdistribusi. Referensi lengkap mengenai Microsoft Visual Studio .NET dapat dilihat pada <http://msdn.microsoft.com/vstudio/>

Microsoft Ms Access 2003 adalah Database Management System yang membangun dan mengelola basis data. Referensi lengkap mengenai Ms Access 2003 dapat dilihat pada <http://www.microsoft.com/>

PDHUPL

PERANCANGAN, DESKRIPSI DAN HASIL UJI PERANGKAT LUNAK

Ancilla 1.0

Disusun oleh:


Aryanto Adi Nugroho

02 07 03605

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen	Halaman
	Universitas Atma Jaya Yogyakarta	<i>PDHUPL-Ancilla</i>	
		Revisi	

DAFTAR PERUBAHAN

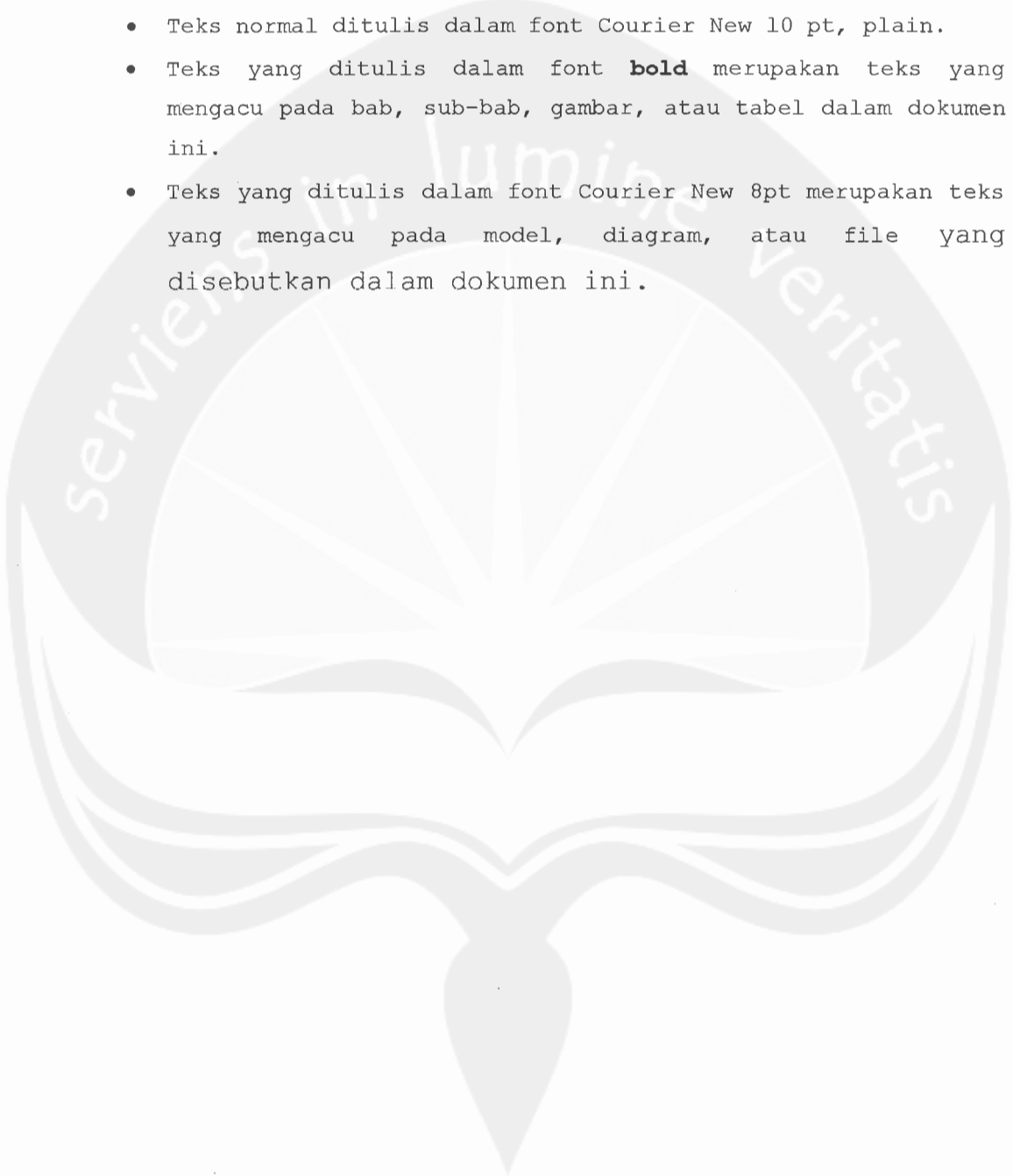
Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E
Ditulis oleh	AR					
Diperiksa oleh	KSN KA					
Disetujui oleh	KSN KA					

NOTASI DOKUMEN

Notasi yang digunakan dalam dokumen ini adalah sebagai berikut :

- Teks normal ditulis dalam font Courier New 10 pt, plain.
- Teks yang ditulis dalam font **bold** merupakan teks yang mengacu pada bab, sub-bab, gambar, atau tabel dalam dokumen ini.
- Teks yang ditulis dalam font Courier New 8pt merupakan teks yang mengacu pada model, diagram, atau file yang disebutkan dalam dokumen ini.



DAFTAR ISI

1	Pendahuluan	6
1.1	Tujuan	6
1.2	Deskripsi Umum Sistem	6
1.3	Istilah dan singkatan	6
1.4	Referensi	6
1.5	Deskripsi Umum Dokumen	7
2	Lingkungan Pengujian Perangkat Lunak	7
2.1	Perangkat Lunak Pengujian	7
2.2	Perangkat Keras Pengujian	7
2.3	Sumber Daya Manusia	7
2.4	Prosedur Umum Pengujian	8
2.4.1	Pengenalan dan Latihan	8
2.4.2	Persiapan Perangkat Keras	7
2.4.3	Persiapan Perangkat Lunak	7
2.4.4	Pelaksanaan	7
2.4.5	Pelaporan Hasil	7
3	Identifikasi dan Rencana Pengujian	9
3.1	Identifikasi Pengujian	9
3.2	Rencana Pengujian	9
3.2.1	Urutan Pelaksanaan Pengujian	9
3.2.2	Data Pengujian	9
4	Identifikasi Pengujian	9
5	Hasil Pengujian	15
	Apendiks A : Daftar Istilah dan Singkatan	30

1.5 Deskripsi Umum Dokumen

Dokumen ini terdiri dari lima bab, yaitu :

1. Bab pertama adalah **Pendahuluan**, yang akan memberikan deskripsi dokumen.
2. Bab kedua adalah **Lingkungan Pengujian Perangkat Lunak**, yang akan menggambarkan lingkungan tempat berjalannya perangkat lunak (perangkat keras dan perangkat lunak), sumber daya manusia, serta prosedur umum pengujian.
3. Bab ketiga adalah **Identifikasi dan Rencana Pengujian**, yang berisi deskripsi umum kelas-kelas dan butir-butir pengujian.
4. Bab keempat adalah **Identifikasi Pengujian**, yang berisi deskripsi rinci kelas-kelas dan butir-butir pengujian.
5. Bab kelima adalah **Hasil Pengujian**, yang berisi langkah-langkah dan hasil pengujian kelas-kelas dan butir-butir pengujian.

2 Lingkungan Pengujian Perangkat Lunak

2.1 Perangkat Lunak Pengujian

Perangkat lunak yang digunakan untuk pengujian berupa :

1. Sistem operasi Microsoft Windows XP Professional Edition.
2. Ms Access 2003.
3. Data-data input program, berupa data *account*, data *user* dan data *client* yang telah dibuat sebelumnya.

2.2 Perangkat Keras Pengujian

Perangkat keras yang digunakan untuk pengujian berupa :

Komputer tempat aplikasi Ancilla berjalan, dengan spesifikasi Pentium 4 1.7 Ghz, 512 MB DDRAM.

2.3 Sumber Daya Manusia

Sumber daya manusia yang digunakan untuk pengujian berupa :

Pembuat perangkat lunak, dengan pengalaman pemrograman berbasis GUI 1,5 tahun.

1 Pendahuluan

1.1 Tujuan

Dokumen PDHUPL ini dibuat untuk menyediakan perencanaan, deskripsi, dan hasil pengujian perangkat lunak Ancilla 1.0. Dokumen ini ditujukan untuk pembuat perangkat lunak, dan orang-orang lain yang tertarik untuk mengembangkan perangkat lunak ini lebih lanjut.

1.2 Deskripsi Umum Sistem

Ancilla adalah sebuah perangkat lunak yang digunakan untuk melakukan pemantauan dan pengendalian komputer secara terpusat di Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta. Ancilla dikembangkan dengan menggunakan bahasa pemrograman C#.NET 2.0 pada Ms. Visual Studio 2005 serta menggunakan Ms. Access 2003 sebagai DBMS untuk data-datanya.

Perangkat lunak ini dibagi menjadi dua bagian, yaitu bagian *AncillaRein* dan bagian *AncillaRemote*. Fungsi dari *AncillaRein* adalah melakukan pemantauan dan pengendalian terhadap *client(AncillaRemote)*, bagian ini berinteraksi langsung dengan *user*. Sedangkan fungsi dari *AncillaRemote* adalah memberikan layanan atas *request* baik berupa perintah eksekusi maupun pengiriman informasi ke *server(AncillaRein)*.

1.3 Istilah dan singkatan

Untuk definisi istilah dan singkatan yang digunakan dalam dokumen ini dapat mengacu pada **Apendiks A : Daftar Istilah dan Singkatan**.

1.4 Referensi

Referensi yang digunakan dalam pembuatan dokumen ini adalah :

- Nugroho, Aryanto Adi. *Spesifikasi Kebutuhan Perangkat Lunak Ancilla 1.0*. Universitas Atma Jaya Yogyakarta. 2007.
- Nugroho, Aryanto Adi. *Deskripsi Perancangan Perangkat Lunak Ancilla 1.0*. Universitas Atma Jaya Yogyakarta. 2007.

2.4 Prosedur Umum Pengujian

2.4.1 Pengenalan dan Latihan

Pengenalan dan Latihan perangkat lunak Ancilla diharapkan tidak memerlukan waktu lama. Ancilla diharapkan dapat dipelajari langsung dari antarmuka bantuan, tanpa melalui pelatihan khusus. Pengguna Ancilla adalah seseorang yang telah familiar dengan penggunaan program berbasis GUI.

2.4.2 Persiapan Perangkat Keras

Persiapan perangkat keras yang digunakan dalam pengujian berupa :

1. Komputer yang terhubung dengan jaringan dan server basis data.
2. Keyboard.
3. Mouse.

2.4.3 Persiapan Perangkat Lunak

Persiapan perangkat lunak yang digunakan dalam pengujian berupa :

1. Instalasi Ms. Access 2003.
2. Instalasi aplikasi Ancilla v.1.0.

2.4.4 Pelaksanaan

Pelaksanaan pengujian akan dilakukan untuk masing-masing *use case*, *basic path* dan *alternative path*. Untuk deskripsi *use case* dapat mengacu ke Spesifikasi Kebutuhan Perangkat Lunak Ancilla.

2.4.5 Pelaporan Hasil

Hasil pengujian akan diserahkan kepada Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta.

3 Identifikasi dan Rencana Pengujian

3.1 Identifikasi Pengujian

Tabel 3.1 Identifikasi Pengujian *Use Case Login*

Kelas Uji	Pengujian <i>use case Login</i>
Butir Uji	<i>Login</i>
Identifikasi	
SKPL	UC-Ancilla-01
PDHUPL	AU-01-01

Tingkat Pengujian	Pengujian unit
Jenis Pengujian	<i>Black box</i>
Jadwal	Juli 2007

Tabel 3.2 Identifikasi Pengujian Use Case Monitoring Client

Kelas Uji	Pengujian Use Case Monitoring Client					
Butir Uji	<i>Data View Last</i>	<i>Data View Update</i>	<i>Data View All</i>	<i>View Process</i>	<i>View User</i>	<i>Ping Module</i>
Identifikasi						
SKPL	UC- Ancilla-02	UC- Ancilla-02	UC- Ancilla-02	UC- Ancilla-02	UC- Ancilla-02	UC- Ancilla-02
PDHUPL	AU-02-01	AU-02-03	AU-02-04	AU-02-05	AU-02-06	AU-02-07
Tingkat Pengujian	Pengujian unit	Pengujian unit	Pengujian unit	Pengujian unit	Pengujian unit	Pengujian unit
Jenis Pengujian	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>
Jadwal	Juli 2007	Juli 2007	Juli 2007	Juli 2007	Juli 2007	Juli 2007

Tabel 3.3 Identifikasi Pengujian Use Case Controlling Client Bag.1

Kelas Uji	Pengujian use case Controlling Client				
Butir Uji	<i>Lock</i>	<i>LogOff</i>	<i>Restart</i>	<i>Shutdown</i>	<i>Wake Up</i>
Identifikasi					
SKPL	UC- Ancilla-03	UC- Ancilla-03	UC- Ancilla-03	UC- Ancilla-03	UC- Ancilla-03
PDHUPL	AU-03-01	AU-03-02	AU-03-03	AU-03-04	AU-03-05
Tingkat Pengujian	Pengujian unit	Pengujian unit	Pengujian unit	Pengujian unit	Pengujian unit
Jenis Pengujian	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>
Jadwal	Juli 2007	Juli 2007	Juli 2007	Juli 2007	Juli 2007

Tabel 3.4 Identifikasi Pengujian Use Case Controlling Client Bag.2

Kelas Uji	Pengujian use case Controlling Client				
Butir Uji	<i>Run Process</i>	<i>Kill Process</i>	<i>Automode</i>	<i>Block Input</i>	<i>Block Input</i>
Identifikasi					
SKPL	UC- Ancilla-03	UC-Ancilla-03	UC- Ancilla-03	UC- Ancilla-03	UC- Ancilla-03
PDHUPL	AU-03-06	AU-03-07	AU-03-08	AU-03-09	AU-03-10

Tingkat Pengujian	Pengujian unit	Pengujian unit	Pengujian unit	Pengujian unit	Pengujian unit
Jenis Pengujian	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>	<i>Black box</i>
Jadwal	Juli 2007	Juli 2007	Juli 2007	Juli 2007	Juli 2007

Tabel 3.5 Identifikasi Pengujian Use Case Account Management

Kelas Uji	Pengujian Use Case Account Management
Butir Uji	<i>Change Password</i>
Identifikasi	
SKPL	UC-Ancilla-04
PDHUPL	AU-04-01
Tingkat Pengujian	Pengujian unit
Jenis Pengujian	<i>Black box</i>
Jadwal	Juli 2007

Tabel 3.6 Identifikasi Pengujian Use Case Data Management

Kelas Uji	Pengujian Use Case Data Management	
Butir Uji	<i>Clean Data Client</i>	<i>Client Data User</i>
Identifikasi		
SKPL	UC-Ancilla-05	UC-Ancilla-05
PDHUPL	AU-05-01	AU-05-02
Tingkat Pengujian	Pengujian unit	Pengujian unit
Jenis Pengujian	<i>Black box</i>	<i>Black box</i>
Jadwal	Juli 2007	Juli 2007

Tabel 3.7 Identifikasi Pengujian Use Case Scan Active Client

Kelas Uji	Pengujian use case Scan Active Client	
Butir Uji	<i>Manual</i>	<i>Automatic</i>
Identifikasi		
SKPL	UC-Ancilla-06	UC-Ancilla-06
PDHUPL	AU-06-01	AU-06-02
Tingkat Pengujian	Pengujian unit	Pengujian unit
Jenis Pengujian	<i>Black box</i>	<i>Black box</i>
Jadwal	Juli 2007	Juli 2007

3.2 Rencana Pengujian

3.2.1 Urutan Pelaksanaan Pengujian

Urutan pengujian sesuai dengan nomor identifikasi pengujian yang telah ditentukan pada bab 3.1.

3.2.2 Data Pengujian

Data pengujian meliputi data *user*, data dan data status jaringan. Data *user* adalah data dari setiap *user* yang memiliki akses untuk masuk ke dalam sistem. Sedangkan data status jaringan merupakan data konektivitas dari setiap *client* yang terhubung dengan sistem dalam suatu jaringan.

4 Identifikasi Pengujian

4.1 Identifikasi Kelas Pengujian Use Case Login

Kelas pengujian ini meliputi pengujian-pengujian yang melibatkan fungsi antarmuka *use case Login* dengan aktor (*user/administrator*) sebagai penggunanya.

4.1.1 Identifikasi Butir Pengujian Login (AU-01-01)

Butir pengujian ini melakukan pengujian terhadap antarmuka login, dengan masukan berupa *password* yang dimasukkan melalui *TextBox*.

4.2 Identifikasi Kelas Pengujian Use Case Monitoring Client

Kelas pengujian ini meliputi pengujian-pengujian yang melibatkan fungsi antarmuka *use case monitoring client* dengan aktor administrator sebagai penggunanya.

4.2.1 Identifikasi Butir Pengujian Data View Last (AU-02-01)

Butir pengujian ini melakukan pengujian terhadap antarmuka *monitoring* untuk fungsi *data view last*. Masukan untuk pengujian ini adalah *hostname*. Data akan ditampilkan dalam *treeview*.

4.2.2 Identifikasi Butir Pengujian Data View Update (AU-02-02)

Butir pengujian ini melakukan pengujian terhadap antarmuka *monitoring* untuk fungsi *data view update*. Masukan untuk pengujian ini adalah *hostname*. Data akan ditampilkan dalam *treeview*.

4.2.3 Identifikasi Butir Pengujian Data View ALL (AU-02-03)

Butir pengujian ini melakukan pengujian terhadap antarmuka *monitoring* untuk fungsi *data view all*. Masukan untuk pengujian ini adalah *hostname*. Data akan ditampilkan dalam *list*. Apabila ada perubahan konfigurasi ditampilkan dengan warna berbeda.

4.3 Identifikasi Kelas Pengujian Use Case Controlling Client.

Kelas pengujian ini meliputi pengujian-pengujian yang melibatkan fungsi antarmuka *use case controlling client* dengan aktor *user/administrator* sebagai penggunaannya.

4.3.1 Identifikasi Butir Pengujian Lock (AU-03-01)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-lock*. Masukan untuk pengujian ini adalah *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.2 Identifikasi Butir Pengujian LogOff (AU-03-02)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-logOff*. Masukan untuk pengujian ini adalah *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.3 Identifikasi Butir Pengujian Restart (AU-03-03)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-restart*. Masukan untuk pengujian ini adalah *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.4 Identifikasi Butir Pengujian Shutdown (AU-03-04)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-shutdown*. Masukan untuk pengujian ini adalah *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.5 Identifikasi Butir Pengujian Wake Up (AU-03-05)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-wake up*. Masukan untuk pengujian ini adalah

destination. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.6 Identifikasi Butir Pengujian *Run Process* (AU-03-06)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-run process*. Masukan untuk pengujian ini adalah *file name* dan *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.7 Identifikasi Butir Pengujian *Kill Process* (AU-03-07)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client kill process*. Masukan untuk pengujian ini adalah *file name* dan *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.8 Identifikasi Butir Pengujian *Automode* (AU-03-08)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-automode*. Masukan untuk pengujian ini adalah *time wake up* dan *time shutdown*. Masukkan dimasukkan melalui *TextBox*.

4.3.9 Identifikasi Butir Pengujian *Block Input* (AU-03-09)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-block input*. Masukan untuk pengujian ini adalah *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.3.10 Identifikasi Butir Pengujian *Unblock Input* (AU-03-10)

Butir pengujian ini melakukan pengujian terhadap antarmuka *controlling client-unblock input*. Masukan untuk pengujian ini adalah *destination*. Masukkan dimasukkan melalui *TextBox*. *User* dapat memilih proses untuk 1 *client* atau *broadcast* ke semua *client*.

4.4 Identifikasi Kelas Pengujian Use Case Account Management

Kelas pengujian ini meliputi pengujian-pengujian yang melibatkan fungsi antarmuka *use case Account Management* dengan aktor user/administrator sebagai penggunanya.

4.4.1 Identifikasi Butir Pengujian Change Password (AU-04-01)

Butir pengujian ini melakukan pengujian terhadap antarmuka *account management-change password*. Masukan untuk pengujian ini adalah *password* baru. Masukkan dimasukkan melalui *TextBox*.

4.5 Identifikasi Kelas Pengujian Use Case Data Management

Kelas pengujian ini meliputi pengujian-pengujian yang melibatkan fungsi antarmuka *use case data management* dengan aktor user/administrator sebagai penggunanya.

4.5.1 Identifikasi Butir Pengujian Clean Data User (AU-05-01)

Butir pengujian ini melakukan pengujian terhadap antarmuka *clean data user*.

4.5.2 Identifikasi Butir Pengujian Clean Data Client (AU-05-02)

Butir pengujian ini melakukan pengujian terhadap antarmuka *clean data client*.

4.6 Identifikasi Kelas Pengujian Scan Active Client

Kelas pengujian ini meliputi pengujian-pengujian yang melibatkan fungsi antarmuka *use case data management* dengan aktor user/administrator sebagai penggunanya. User dapat memilih 2 metode, yaitu : *manual* dan *automatic*.

4.6.1 Identifikasi Butir Pengujian Manual (AU-06-01)

Butir pengujian ini melakukan pengujian terhadap antarmuka proses *scan* secara *manual*. User dalam melakukan *scan* harus menekan tombol *scan* terlebih dahulu.

4.6.2 Identifikasi Butir Pengujian Automatic (AU-06-02)

Butir pengujian ini melakukan pengujian terhadap antarmuka proses *scan* secara *automatic*. Sistem akan melakukan *scan* secara otomatis dalam satuan waktu.

5 Hasil Pengujian

5.1 Hasil Pengujian Use Case Login

5.1.1 Hasil Pengujian Login (AU-01-01)

Tabel 5.1 Hasil Pengujian Login (AU-01-01)

Identifikasi					
AU-01-01					
Pengujian terhadap antarmuka login oleh aktor user/administrator.					
Deskripsi	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
<ul style="list-style-type: none"> - Masukkan password administrator yang valid - Tekan tombol "SignIn" 	<ul style="list-style-type: none"> - Password : administrator pada textbox password - Tekan tombol "SignIn" 	<ul style="list-style-type: none"> - Menu utama Ancilla aktif 	<ul style="list-style-type: none"> - Semua menu yang ada menjadi aktif 	<ul style="list-style-type: none"> - Menu utama Ancilla aktif 	Handal
<ul style="list-style-type: none"> - Kosongi textbox password - Tekan tombol "SignIn" 	<ul style="list-style-type: none"> - Password : "" - Tekan tombol "SignIn" 	<ul style="list-style-type: none"> - Message "Insert your password!" 	<ul style="list-style-type: none"> - Keluar message "Insert your password!" 	<ul style="list-style-type: none"> - Message "Insert your password!" 	Handal
<ul style="list-style-type: none"> - Masukkan password salah - Tekan tombol "SignIn" 	<ul style="list-style-type: none"> - Password : "rajjoko" - Tekan tombol "SignIn" 	<ul style="list-style-type: none"> - Message "You insert incorrect password" 	<ul style="list-style-type: none"> - Keluar message "You insert incorrect password" 	<ul style="list-style-type: none"> - Message "You insert incorrect password" 	Handal

5.2 Hasil Pengujian Use Case Monitoring Client

5.2.1 Hasil Pengujian Data View Last (AU-02-01)

Tabel 5.2 Hasil Pengujian Data View Last (AU-02-01)

Identifikasi AU-02-01					
Pengujian terhadap antarmuka data View Last					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
<ul style="list-style-type: none"> - Aktifkan tab monitoring - Aktifkan tab System Info dalam tab monitoring. - Masukkan destination 	<ul style="list-style-type: none"> - Name : "MYCOMP" - Tekan tombol "View" 	<ul style="list-style-type: none"> - Data last MYCOMP ditampilkan pada treeview 	<ul style="list-style-type: none"> - Tampilan pada treeview 	<ul style="list-style-type: none"> - Data last MYCOMP ditampilkan pada treeview 	Handal
<ul style="list-style-type: none"> - Sama dengan pengujian pertama, hostname invalid. 	<ul style="list-style-type: none"> - Name : MYCOMPI - Tekan tombol "View" 	<ul style="list-style-type: none"> - Message "You insert unknown client" 	<ul style="list-style-type: none"> - Keluar message "You insert unknown client" 	<ul style="list-style-type: none"> - Message "You insert unknown client" 	Handal
<ul style="list-style-type: none"> - Sama dengan pengujian pertama, tapi text box destination tidak diisi 	<ul style="list-style-type: none"> - Name : " " - Tekan tombol "View" 	<ul style="list-style-type: none"> - Message "Please insert your destination" 	<ul style="list-style-type: none"> - Keluar message "Please insert your destination" 	<ul style="list-style-type: none"> - Message "Please insert your destination" 	Handal

5.2.2 Hasil Pengujian Data View Update (AU-02-01)

Tabel 5.3 Hasil Pengujian Data View Last (AU-02-01)

Identifikasi AU-02-02	
Pengujian terhadap antarmuka data view update	
Identifikasi	Deskripsi

Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
<ul style="list-style-type: none"> - Aktifkan tab <i>monitoring</i> - Aktifkan tab <i>System Info Update</i> dalam tab <i>monitoring</i>. - Masukkan <i>destination</i> 	<ul style="list-style-type: none"> - Name : "MYCOMP" - Tekan tombol "View" 	<ul style="list-style-type: none"> - Data <i>last update</i> MYCOMP ditampilkan pada <i>treeview</i> - Konfigurasi tertampil di <i>List Configuration Change</i> 	<ul style="list-style-type: none"> - Tampilan pada <i>treeview</i> - Tampilan di <i>List Configuration Change</i> 	<ul style="list-style-type: none"> - Data <i>last MYCOMP</i> ditampilkan pada <i>treeview</i> - Konfigurasi tertampil di <i>List Configuration Change</i> 	Handal
<ul style="list-style-type: none"> - Sama dengan prosedur pengujian pertama, tapi <i>hostname</i> tidak valid. 	<ul style="list-style-type: none"> - Name : MYCOMP1 - Tekan tombol "View" 	<ul style="list-style-type: none"> - Message "You insert unknown client" 	<ul style="list-style-type: none"> - Keluar message "You insert unknown client" 	<ul style="list-style-type: none"> - Message "You insert unknown client" 	Handal
<ul style="list-style-type: none"> - Sama dengan prosedur pengujian pertama, tapi text <i>box destination</i> tidak diisi 	<ul style="list-style-type: none"> - Name : " " - Tekan tombol "View" 	<ul style="list-style-type: none"> - Message "Please insert your destination" 	<ul style="list-style-type: none"> - Keluar message "Please insert your destination" 	<ul style="list-style-type: none"> - Message "Please insert your destination" 	Handal
<ul style="list-style-type: none"> - Sama dengan prosedur pengujian pertama, tapi tidak ada status perubahan konfigurasi 	<ul style="list-style-type: none"> - Name : " MYCOMP" - Tekan tombol "View" 	<ul style="list-style-type: none"> - Message "No configuration changed" di <i>treeview data update</i> dan <i>list configuration changed</i>. 	<ul style="list-style-type: none"> - Keluar message "Please insert your destination" di <i>treeview data update</i> dan <i>list configuration changed</i>. 	<ul style="list-style-type: none"> - Message "No configuration changed" di <i>treeview data update</i> dan <i>list configuration changed</i>. 	Handal

5.2.3 Hasil Pengujian Data View All (AU-02-03)

Tabel 5.4 Hasil Pengujian Data View All(AU-02-03)

Identifikasi					
	AU-02-01				
Deskripsi	Pengujian terhadap antarmuka data view all				
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Aktifkan tab Data View	-	<ul style="list-style-type: none"> - Data view semua client ditampilkan pada treeview - Client yang berubah konfigurasinya ditampilkan dengan warna merah. 	<ul style="list-style-type: none"> - Tampilan pada treeview 	<ul style="list-style-type: none"> - Data view semua client ditampilkan pada treeview - Client yang berubah konfigurasinya ditampilkan dengan warna merah. 	Handal

5.2.4 Hasil Pengujian View Process (AU-02-04)

Tabel 5.5 Hasil Pengujian View Process(AU-02-04)

Identifikasi					
	AU-02-01				
Deskripsi	Pengujian terhadap antarmuka view process				
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
<ul style="list-style-type: none"> - Aktifkan tab monitoring - Aktifkan tab System Info dalam tab monitoring. - Masukkan destination - Sama dengan prosedur pengujian pertama, hostname invalid. 	<ul style="list-style-type: none"> - Name : "MYCOMP" - Tekan tombol "View" - Name : MYCOMPi - Tekan tombol "View" 	<ul style="list-style-type: none"> - Data process MYCOMP ditampilkan pada list process - Message "Failed get process list" 	<ul style="list-style-type: none"> - Tampilan pada list process - Keluar message "Failed get process list" 	<ul style="list-style-type: none"> - Data process MYCOMP ditampilkan pada list process - Message "Failed get process list" 	Handal

- Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi	- Name : " " - Tekan tombol "View"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"		Handal

5.2.5 Hasil Pengujian View User (AU-02-05)

Tabel 5.5 Hasil Pengujian View User (AU-02-05)

Identifikasi		AU-02-05			
Deskripsi					
Pengujian terhadap antarmuka view user					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Aktifkan tab monitoring - Aktifkan tab System Info dalam tab monitoring. - Masukkan destination	- Name : "MYCOMP" - Tekan tombol "View"	- Data user MYCOMP ditampilkan pada list user	- Tampilan pada list process	- Data user MYCOMP ditampilkan pada list user	Handal
- Sama dengan prosedur pengujian pertama, hostname invalid.	- Name : MYCOMPI - Tekan tombol "View"	- Message "Failed get user list"	- Keluar message "Failed get user list"	- Message "Failed get user list"	Handal
- Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi	- Name : " " - Tekan tombol "View"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"	Handal

5.3 Hasil Pengujian Use Case Controlling Client

5.3.1 Hasil Pengujian Lock (AU-03-01)

Tabel 5.3 Hasil Pengujian Lock (AU-03-01)

Identifikasi					
AU-03-01					
Pengujian terhadap antarmuka Lock					
Deskripsi	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
Prosedur pengujian - Aktifkan tab controlling - Masukkan destination	- Name : "MYCOMP" - Tekan tombol "Lock"	- Message "MYCOMP lock" (untuk all option : Message "All client lock")	- Keluar message "MYCOMP lock"	- Message "MYCOMP lock" (untuk all option : Message "All client lock")	Handal
	- Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi	- Name : " - Tekan tombol "Lock"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"

5.3.2 Hasil Pengujian LogOff (AU-03-02)

Tabel 5.4 Hasil Pengujian LogOff (AU-03-02)

Identifikasi					
AU-03-02					
Pengujian terhadap antarmuka Logoff					
Deskripsi	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
Prosedur pengujian - Aktifkan tab controlling - Masukkan	- Name : "MYCOMP" - Tekan tombol "LogOff"	- Message "MYCOMP LogOff" (untuk all option : Message "All client	- Keluar message "MYCOMP LogOff"	- Message "MYCOMP LogOff" (untuk all option : Message "All	Handal

destination		LogOff")	client LogOff")	
- Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi	- Name : " " - Tekan tombol "LogOff"	- Message "Please insert your destination"	- Message "Please insert your destination"	Handal

5.3.3 Hasil Pengujian Restart (AU-03-03)

Tabel 5.5 Hasil Pengujian restart(AU-03-03)

Identifikasi					
	AU-03-03				
Deskripsi	Pengujian terhadap antarmuka restart				
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Aktifkan tab controlling - Masukkan destination	- Name : "MYCOMP" - Tekan tombol "restart"	- Message "MYCOMP restart" (untuk all option : Message "All client restart")	- Keluar message "MYCOMP restart"	- Message "MYCOMP restart" (untuk all option : Message "All client restart")	Handal
- Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi	- Name : " " - Tekan tombol "restart"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"	Handal

5.3.4 Hasil Pengujian Shutdown (AU-03-04)

Tabel 5.6 Hasil Pengujian shutdown (AU-03-04)

Identifikasi					
AU-03-04					
Pengujian terhadap antarmuka shutdown					
Deskripsi	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
Prosedur pengujian					
- Aktifkan tab controlling	- Name : "MYCOMP"	- Message "MYCOMP shutdown"	- Keluar message "MYCOMP shutdown"	- Message "MYCOMP shutdown"	Handal
- Masukkan destination	- Tekan tombol "shutdown"	(untuk all option : Message "All client shutdown")		(untuk all option : Message "All client shutdown")	
- Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi	- Name : "" - Tekan tombol "shutdown"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"	Handal

5.3.5 Hasil Pengujian wake up (AU-03-05)

Tabel 5.7 Hasil Pengujian wake up (AU-03-05)

Identifikasi					
AU-03-05					
Pengujian terhadap antarmuka wake up					
Deskripsi	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
Prosedur pengujian					
- Aktifkan tab controlling	- MAC : "00-A0-D1-57-74-CE"	- Message "wake up process"	- Keluar message "wake up process"	- Message "wake up process"	Handal
- Masukkan mac address	- Tekan tombol "wake"	(untuk all option : Message "All client wake up process")		(untuk all option : Message "All client wake up process")	

- Sama dengan prosedur pengujian pertama, tapi text box <i>destination</i> tidak diisi	- Name : " " - Tekan tombol "wake"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"	Handal
--	---------------------------------------	--	---	--	--------

5.3.6 Hasil Pengujian run process (AU-03-06)

Tabel 5.8 Hasil Pengujian run process (AU-03-06)

Identifikasi		AU-03-06			
Deskripsi		Pengujian terhadap antarmuka run process			
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Aktifkan tab <i>controlling</i> - Masukkan nama <i>file destination</i>	- File : "notepad" - Name : "MYCOMP" - Tekan tombol "run"	- Message "MYCOMP notepad run" (untuk all option : Message "All client notepad run")	- Keluar message "MYCOMP notepad run"	- Message "MYCOMP notepad run" (untuk all option : Message "All client notepad run")	Handal
- Sama dengan prosedur pengujian pertama, tapi text box <i>destination</i> tidak diisi	- File : "notepad" - Name : " " - Tekan tombol "run"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"	Handal
- Sama dengan prosedur pengujian pertama, tapi text box <i>file name</i> tidak diisi	- File : " " - Name : "MYCOMP" - Tekan tombol "run"	- Message "Please insert your file name"	- Keluar message "Please insert your file name"	- Message "Please insert your file name"	Handal

<ul style="list-style-type: none"> - Aktifkan tab controlling - Masukkan time wake up dan time shutdown 	<ul style="list-style-type: none"> - Wake Up : "12:00:00" - Shutdown : "12:10:00" - Tekan tombol "set up" 	<ul style="list-style-type: none"> - Message "Automode was active" 	<ul style="list-style-type: none"> - Keluar Message "Automode was active" 	<ul style="list-style-type: none"> - Message "Automode was active" 	Handal
<ul style="list-style-type: none"> - Sama dengan prosedur pengujian pertama, tapi text box time tidak diisi 	<ul style="list-style-type: none"> - Wake Up : "" - Shutdown : "" - Tekan tombol "set up" 	<ul style="list-style-type: none"> - Message "Please insert your time" 	<ul style="list-style-type: none"> - Keluar message "Please insert your time" 	<ul style="list-style-type: none"> - Message "Please insert your time" 	Handal

5.3.9 Hasil Pengujian block input (AU-03-09)

Tabel 5.11 Hasil Pengujian bLock input(AU-03-09)

Identifikasi	AU-03-09	Penguujian terhadap antarmuka block input			
Deskripsi	Penguujian terhadap antarmuka block input				
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
<ul style="list-style-type: none"> - Aktifkan tab controlling - Masukkan destination 	<ul style="list-style-type: none"> - Name : "MYCOMP" - Tekan tombol "block" 	<ul style="list-style-type: none"> - Message "MYCOMP blocked" (untuk all option : Message "All client blocked") 	<ul style="list-style-type: none"> - Keluar message "MYCOMP blocked" 	<ul style="list-style-type: none"> - Message "MYCOMP blocked" (untuk all option : Message "All client blocked") 	Handal
<ul style="list-style-type: none"> - Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi 	<ul style="list-style-type: none"> - Name : "" - Tekan tombol "block" 	<ul style="list-style-type: none"> - Message "Please insert your destination" 	<ul style="list-style-type: none"> - Keluar message "Please insert your destination" 	<ul style="list-style-type: none"> - Message "Please insert your destination" 	Handal

5.3.10 Hasil Pengujian unblock input (AU-03-10)

Tabel 5.12 Hasil Pengujian unblock input(AU-03-10)

Identifikasi					
AU-03-10					
Deskripsi					
Pengujian terhadap antarmuka block input					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Aktifkan tab controlling - Masukkan destination	- Name : "MYCOMP" - Tekan tombol "unblock"	- Message "MYCOMP unblocked" (untuk all option : Message "All client unblocked")	- Keluar message "MYCOMP blocked"	- Message "MYCOMP unblocked" (untuk all option : Message "All client unblocked")	Handal
- Sama dengan prosedur pengujian pertama, tapi text box destination tidak diisi	- Name : " " - Tekan tombol "unblock"	- Message "Please insert your destination"	- Keluar message "Please insert your destination"	- Message "Please insert your destination"	Handal

5.4 Hasil Pengujian use case Account Management

5.4.1 Hasil Pengujian use case Account Management(AU-04-01)

Tabel 5.13 Hasil Pengujian Change Password (AU-04-01)

Identifikasi					
AU-04-01					
Deskripsi					
Pengujian terhadap antarmuka change password					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Pilih menu "Manage > Account" - Masukkan password	- Old Password : "arie" - New Password :	- Message "Update password success"	- Keluar message "Update password success"	- Message "Update password success"	Handal

<p>lama</p> <ul style="list-style-type: none"> - Masukkan password baru - Masukkan konfirmasi - Tekan tombol "Save" 	<p>"admin"</p> <ul style="list-style-type: none"> - Confirm password : "admin" - Tekan tombol "Save" 				
<ul style="list-style-type: none"> - Sama dengan prosedur pengujian pertama, tapi password baru tidak sesuai dengan konfirmasi password baru 	<ul style="list-style-type: none"> - Old Password : "admin" - New Password : "administrator" - Confirm password : "adminis" - Tekan tombol "Save" 	<p>- Message "Please insert same new word!"</p>	<p>- Keluar message "Please insert same new word!"</p>	<p>- Message "Please insert same new word!"</p>	Handal
<ul style="list-style-type: none"> - Sama dengan prosedur pengujian pertama, tapi ada satu atau lebih field yang tidak diisi 	<ul style="list-style-type: none"> - Old Password : "admin" - New Password : "" - Confirm password : "" - Tekan tombol "Save" 	<p>- Message "Please insert all textbox"</p>	<p>- Keluar message "Please insert all textbox"</p>	<p>- Message "Please insert all textbox"</p>	Handal

5.5 Hasil Pengujian Use Case Data Management

5.5.1 Hasil Pengujian Clean Data Client (AU-05-01)

Tabel 5.14 Hasil Pengujian Clean Data Client (AU-05-01)

Identifikasi AU-05-01					
Deskripsi Pengujian terhadap antarmuka clean data client					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Pilih menu "manage > Clean Data" - Tekan tombol "Empty"	- Tekan tombol "Empty"	- Message "Your data has been cleaned"	- Keluar message "Your data has been cleaned"	- Message "Your data has been cleaned"	Handal

5.5.2 Hasil Pengujian Clean Data User (AU-05-02)

Tabel 5.15 Hasil Pengujian Clean Data User (AU-05-02)

Identifikasi AU-05-01					
Deskripsi Pengujian terhadap antarmuka clean data user					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Pilih menu "manage > Clean Data" - Tekan tombol "Empty"	- Tekan tombol "Empty"	- Message "Your data has been cleaned"	- Keluar message "Your data has been cleaned"	- Message "Your data has been cleaned"	Handal

5.6 Hasil Pengujian Use Case Scan Active Client

5.6.1 Hasil Pengujian Scan Manual (AU-06-01)

Tabel 5.7 Hasil Pengujian Scan Manual (AU-06-01)

Identifikasi					
AU-06-01					
Deskripsi					
Pengujian terhadap antarmuka scan manual					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Pilih radiobutton manual - Tekan tombol "Scan"	- Pilih radiobutton : "manual" - Tekan tombol "Scan"	- Data client tertampil treeview client	- Data treeview client	- Data client tertampil treeview client	Handal

5.6.2 Hasil Pengujian Scan Manual (AU-06-02)

Tabel 5.7 Hasil Pengujian Scan Manual (AU-06-02)

Identifikasi					
AU-06-01					
Deskripsi					
Pengujian terhadap antarmuka scan automatic					
Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
- Pilih radiobutton automatic - Tekan tombol "Scan"	- Pilih radiobutton : "automatic" - Tekan tombol "Scan"	- Data client tertampil treeview client	- Data treeview client	- Data client tertampil treeview client	Handal

Apendiks A : Daftar Istilah dan Singkatan

Administrator adalah operator yang berhak mengakses seluruh fungsionalitas Ancilla 1.0.

