

BAB VI

KESIMPULAN DAN SARAN

VI.1 Kesimpulan

Berdasarkan pembahasan bab-bab sebelumnya dan dokumen-dokumen penunjang seperti SKPL dan DPPL, maka dapat diambil kesimpulan seperti berikut:

1. Cara mengkomunikasikan sistem server dan sistem klien dapat dilakukan dengan pemrograman *socket*.
2. Pengendalian aktifitas chatting, browsing, copy paste, USB, dan firewall dapat dilakukan dengan perintah block dari komputer server.
3. Permintaan data sistem komputer client dapat dilakukan dengan pengiriman data string dari client ke server.
4. Pengiriman file yang berukuran lebih dari 60KB, dilakukan proses pemecahan menjadi part-part berukuran 60KB. Yang kemudian dikirim satu-persatu ke komputer klien. Ketika part-part tersebut telah lengkap diterima klien maka dilakukan proses penggabungan kembali file tersebut.
5. Proses pengendalian untuk menjalankan aplikasi tertentu. Komputer server akan mengirimkan perintah dan letak aplikasi yang akan dijalankan ke komputer klien.

VI.2 Saran

Beberapa saran dari penulis untuk pengembangan perangkat lunak ini lebih lanjut adalah:

1. Dengan aplikasi ini tidak dapat melihat komputer klien, sehingga dapat dikembangkan dengan menambahkan fungsi untuk melihat komputer klien.
2. Dengan menggunakan protokol UDP, maka data dapat hilang atau rusak. Sehingga dapat dibuat fungsi untuk membuat pengiriman data lebih handal.

Daftar Pustaka

- Fedrik. 2010. Pengembangan Sistem Pengiriman File Dan Pengendalian Jarak Jauh Menggunakan User Datagram Protocol. Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Idris, 2009, 20 Mei 2011 20.00 WIB , "Socket Programming Delphi", "<http://master-komputer.com/tutorial/41-programming/86-socket-programming-delphi.pdf>".
- Kristanto, Andri, *Jaringan Komputer*, Graha Ilmu, Yogyakarta, 2003.
- Nugroho, Aryanto Adi. 2009. Pengembangan Sistem Pemantauan Dan Pengendalian Komputer Terpusat (Studi Kasus Laboratorium Jaringan Komputer Teknik Informatika UAJY). Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Postel, J., 1980, "*User Datagram Protocol*".
- Purbo, Onno W., dkk, "TCP/IP Standar, Desain, dan Implementasi", PT Elex Media Komputindo, 2000.
- Ramos, Emilio, et.al, "*Computer Networking Concepts*", Prentice-Hall, Inc, New Jersey, 1996.
- Richolas, Tjhai. 2009. Pembangunan Aplikasi Lab Discipline Control. Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta, Yogyakarta.

Shetty, Nikhil, 2006, 26 Juni 2011 18.30 WIB , "Socket Programming", " <http://inst.eecs.berkeley.edu/~eel22/sp06/LectureNotes/Socket%20Programming.pdf>"

Wijaya, I Made Yonatan dan Utomo, Ndon Eko Satriyo. 2010. Implementasi Visual Basic 6 untuk Pembuatan Aplikasi Remote Access Berbasis TCP/IP. Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Amikom, Yogyakarta.



SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

LabAssist (Laboratorium Assist)

Untuk :
Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:
Maria Nila Anggia Rini/060704938

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>SKPL-LabAssist</i>		1/34
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

Halaman Judul.....	1
Daftar Perubahan.....	2
Daftar Halaman Perubahan.....	3
Daftar Isi.....	4
Daftar Gambar.....	5
1 Pendahuluan	6
1.1 Tujuan	6
1.2 Lingkup Masalah	6
1.3 Definisi, Akronim dan Singkatan	6
1.4 Referensi	7
1.5 Deskripsi umum (Overview)	7
2 Deskripsi Kebutuhan	8
2.1 Perspektif produk	8
2.2 Fungsi Produk	9
2.3 Karakteristik Pengguna	13
2.4 Batasan-batasan	13
2.5 Asumsi dan Ketergantungan	14
3 Kebutuhan khusus	14
3.1 Kebutuhan antarmuka eksternal	14
3.2 Kebutuhan Fungsionalitas Perangkat Lunak	15
4 Spesifikasi Rinci Kebutuhan	16
4.1 Spesifikasi Kebutuhan Fungsionalitas	16
5 Entity Relationship Diagram (ERD)	34

Daftar Gambar

1. Arsitektur Perangkat Lunak LabAssist	9
2. Use Case Diagram	15
3. Entity Relationship Diagram	34



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan *LabAssist* untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna) dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-*LabAssist* ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak *LabAssist* dikembangkan dengan tujuan untuk :

1. Mempermudah mengontrol komputer *client* di laboratorium tanpa harus mengecek satu persatu komputer *client*
2. Menangani pengelolaan jadwal praktikum.
3. Mengendalikan komputer *client* untuk melakukan *blok* aplikasi tertentu.
4. Menginstal program di komputer klien dari komputer server.
5. Melihat data sistem dari komputer *client*.
6. Melakukan pengiriman *file* ke komputer *client*.

Dan berjalan pada lingkungan ber-*platform* desktop.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi	
Program Studi Teknik Informatika	SKPL – LabAssist	6/ 34
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL- LabAssist-XXX	Kode yang merepresentasikan kebutuhan pada LabAssist dimana XXX merupakan nomor fungsi produk.
LabAssist	Perangkat lunak untuk melakukan monitoring dan pengendalian pada komputer <i>client</i>
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Sapta, Juli, *Spesifikasi Kebutuhan Perangkat Lunak SC3 (Smart Client for Cyber Community)*, Universitas Atma Jaya Yogyakarta, 2006.
2. Fedrik, *Spesifikasi Kebutuhan Perangkat Lunak SPUAJJ (Sistem Pengiriman File Dan Pengendalian Jarak Jauh)*, Universitas Atma Jaya Yogyakarta, 2010

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

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

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

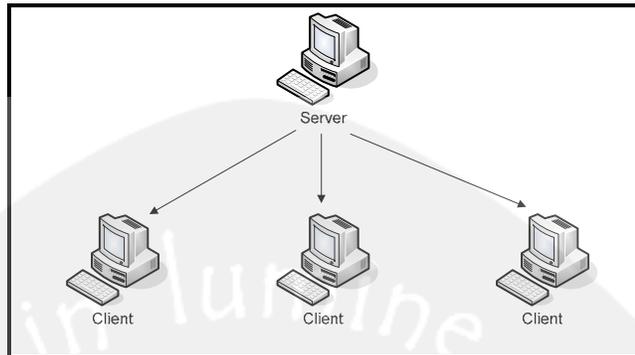
2 Deskripsi Kebutuhan

2.1 Perspektif produk

LabAssist merupakan perangkat lunak yang dikembangkan untuk membantu user mengendalikan komputer *client* dan melakukan pengelolaan jadwal praktikum dalam suatu laboratorium.

Perangkat lunak *LabAssist* ini berjalan pada platform *Desktop Application* dan dibuat menggunakan bahasa pemrograman Microsoft Visual C# dalam lingkungan pemrograman Microsoft Visual Studio 2010.

Pengguna akan berinteraksi dengan sistem melalui antarmuka Graphical User Interface. Pada sistem ini, seperti terlihat pada Gambar 1, arsitektur perangkat lunak yang digunakan berupa client server. Pada sistem ini data user akan disimpan dan user dapat melakukan login. Sistem ini akan dapat melakukan pengelolaan jadwal praktikum, mengendalikan komputer *client* dengan melakukan *blok* pada aplikasi yang dilarang, dan mengirimkan *file* dari komputer *server* ke computer *client*.



Gambar 1. Arsitektur perangkat lunak LabAssist

2.2 Fungsi Produk

Fungsi produk perangkat lunak LabAssist adalah sebagai berikut :

1. Fungsi *Login* (**SKPL-LabAssist-01**)

Merupakan fungsi yang digunakan oleh user untuk dapat masuk dalam sistem yang akan digunakan.

2. Fungsi Kelola User (**SKPL-LabAssist-02**)

Merupakan fungsi yang digunakan oleh user untuk mengelola data user.

Fungsi Pengelolaan User terdiri dari:

a. Fungsi *Tambah User* (**SKPL-LabAssist-02-01**)

Merupakan fungsi untuk menambah user yang menggunakan sistem.

b. Fungsi *Edit User* (**SKPL-LabAssist-02-02**)

Merupakan fungsi untuk mengedit data *password* user yang menggunakan system.

3. Fungsi Jalankan Aplikasi (**SKPL-LabAssist-03**)

Merupakan fungsi yang digunakan oleh user untuk mengendalikan komputer client agar menjalankan aplikasi tertentu.

4. Fungsi Blok (**SKPL-LabAssist-04**)

Merupakan fungsi yang digunakan oleh user untuk mengendalikan komputer client agar melakukan blocking terhadap aplikasi tertentu.

Fungsi Blok terdiri dari:

a. Fungsi Blok Aplikasi **(SKPL-LabAssist-04-01)**

Merupakan fungsi yang digunakan oleh user untuk mengendalikan komputer client agar melakukan blocking terhadap aplikasi tertentu yang ditandai user.

b. Fungsi Blok Aplikasi Selain **(SKPL-LabAssist-04-02)**

Merupakan fungsi yang digunakan oleh user untuk mengendalikan komputer client agar melakukan blocking terhadap aplikasi selain yang ditandai user.

c. Fungsi Blok Copy-Paste **(SKPL-LabAssist-04-03)**

Merupakan fungsi yang digunakan oleh user untuk mengendalikan komputer client agar melakukan blocking terhadap copy-paste.

d. Fungsi Blok USB **(SKPL-LabAssist-04-04)**

Merupakan fungsi yang digunakan oleh user untuk melakukan blok terhadap flashdisk.

e. Fungsi Blok Firewall **(SKPL-LabAssist-04-05)**

Merupakan fungsi yang digunakan oleh user untuk mematikan windows firewall client

5. Fungsi Kirim File **(SKPL-LabAssist-05)**

Merupakan fungsi yang digunakan oleh user untuk melakukan pengiriman file ke komputer.

6. Fungsi Kelola Laboratorium **(SKPL-LabAssist-06)**

Merupakan fungsi yang digunakan oleh user untuk melakukan pengelolaan jadwal praktikum.

Fungsi Pengelolaan Laboratorium mencakup :

a. Fungsi Tambah Jadwal Harian (**SKPL-LabAssist-06-01**)

Merupakan fungsi yang digunakan oleh user untuk menambah penambahan jadwal praktikum harian.

b. Fungsi Ubah Jadwal Harian (**SKPL-LabAssist-06-02**)

Merupakan fungsi yang digunakan oleh user untuk mengubah jadwal praktikum harian

c. Fungsi Hapus Jadwal Harian (**SKPL-LabAssist-06-03**)

Merupakan fungsi yang digunakan oleh user untuk menghapus jadwal praktikum harian tertentu.

d. Fungsi Tambah Acara Khusus (**SKPL-LabAssist-06-04**)

Merupakan fungsi yang digunakan oleh user untuk menambah jadwal khusus, yang dilaksanakan sekali.

e. Fungsi Ubah Acara Khusus (**SKPL-LabAssist-06-05**)

Merupakan fungsi yang digunakan oleh user untuk mengubah jadwal khusus.

f. Fungsi Hapus Acara Khusus (**SKPL-LabAssist-06-06**)

Merupakan fungsi yang digunakan oleh user untuk menghapus jadwal khusus.

7. Fungsi Pantau Data Sistem (**SKPL-LabAssist-07**)

Merupakan fungsi yang digunakan oleh user untuk memantau data sistem di komputer client.

Fungsi Pemantauan Data Sistem mencakup :

a. Fungsi Tampil Data Hardware **(SKPL-LabAssist-07-01)**

Merupakan fungsi yang digunakan user untuk menampilkan data hardware komputer client.

b. Fungsi Tampil Data Software **(SKPL-LabAssist-07-02)**

Merupakan fungsi yang digunakan user untuk menampilkan nama software komputer client.

c. Fungsi Tampil Data Process **(SKPL-LabAssist-07-03)**

Merupakan fungsi yang digunakan oleh user untuk menampilkan nama proses dan *username* dari proses yang terjadi komputer client.

d. Fungsi Cari Data **(SKPL-LabAssist-07-04)**

Merupakan fungsi yang digunakan oleh user untuk menampilkan nama proses dan *username* dari proses yang terjadi komputer client.

8. Fungsi Ubah Bahasa **(SKPL-LabAssist-08)**

Merupakan fungsi yang digunakan oleh user untuk mengubah bahasa pada Region and Language di Control Panel pada computer client.

9. Fungsi Kelola Aplikasi **(SKPL-LabAssist-09)**

Merupakan fungsi yang digunakan oleh user untuk melakukan pengelolaan terhadap aplikasi yang nantinya akan dipilih user untuk diblok. Dengan menyimpan jenis aplikasi dan nama aplikasi.

a. Fungsi Tambah Daftar Aplikasi **(SKPL-LabAssist-09-01)**

Merupakan fungsi yang digunakan oleh user untuk menambah daftar aplikasi.

b. Fungsi Ubah Daftar Aplikasi (**SKPL-LabAssist-09-02**)

Merupakan fungsi yang digunakan oleh user untuk mengubah data aplikasi tertentu.

c. Fungsi Hapus Daftar Aplikasi (**SKPL-LabAssist-09-03**)

Merupakan fungsi yang digunakan oleh user untuk menghapus data aplikasi tertentu.

10. Fungsi Kelola Alamat IP Klien (**SKPL-LabAssist-09**)

Merupakan fungsi yang digunakan oleh user untuk mendapatkan semua alamat IP klien yang ditangani oleh server.

11. Fungsi Instal Program (**SKPL-LabAssist-11**)

Merupakan fungsi yang digunakan oleh user untuk menginstal program di seluruh komputer klien.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak LabAssist adalah sebagai berikut :

1. Memahami pengoperasian komputer/PC.
2. Memahami pengoperasian fungsi-fungsi yang terdapat dalam LabAssist.
3. Memahami penggunaan jaringan LAN

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak LabAssist tersebut adalah :

1. Kebijakan Umum
Berpedoman pada tujuan dari pengembangan perangkat lunak LabAssist.
2. Keterbatasan perangkat keras

Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada komputer yang terhubung dalam jaringan dengan didukung:

1. Sistem operasi Windows. (Minimal Windows XP SP 2 atau Windows Server 2003)
2. Framework 4.0

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak LabAssist meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak LabAssist adalah:

1. PC Client
2. PC Server
3. Modem atau LAN Card

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak LabAssist adalah sebagai berikut :

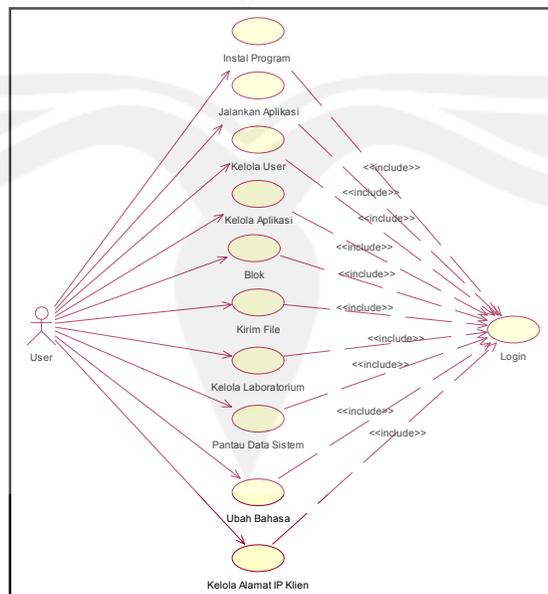
1. Nama : .Net Framework (framework 4.0)
Sumber : Microsoft
Sebagai framework dari perangkat lunak.
2. Nama : Windows XP (minimal XP SP2)
Sumber : Microsoft.
Sebagai sistem operasi komputer.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak LabAssist menggunakan protokol UDP untuk melakukan broadcast dari komputer server ke komputer client dan pengiriman data dari komputer client ke komputer server.

3.2 Kebutuhan Fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram

4 Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Specification : Login

1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah id unik yaitu username dan password yang berupa rangkaian karakter.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor memasukkan username dan password.
4. Sistem memeriksa username dan password yang diinputkan aktor.

E-1 Username atau password tidak diisi.

E-2 Username atau password tidak sesuai.

5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

5. Alternative Flow

None

6. Error Flow

E-1 Username atau password tidak diisi.

1. Sistem menampilkan peringatan bahwa username atau password harus diisi terlebih dahulu.
2. Kembali ke Basic Flow langkah ke 3.

E-2 Username dan password tidak sesuai.

1. Sistem menampilkan peringatan bahwa username atau password tidak sesuai.
2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

None

8. PostConditions

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.

4.1.2 Use case Spesification : Kelola User

1. Brief Description

Use Case ini digunakan oleh aktor mengelola user. Aktor dapat melakukan tambah user, edit user.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Kelola User.
2. Aktor memilih untuk melakukan tambah user.
A-1 Aktor memilih untuk melakukan edit user.
3. Aktor memasukkan data user.
4. Aktor meminta sistem untuk menyimpan data user yang telah dimasukkan.
5. Sistem mengecek data user yang telah dimasukkan.
E-1 Data user yang diinputkan aktor tidak lengkap.
E-2 Nama user telah ada di sistem.
E-3 Confirm Password salah
6. Sistem menyimpan data user ke basis data.

Program Studi Teknik Informatika	SKPL – LabAssist	17/ 34
----------------------------------	------------------	--------

7. Use Case ini selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit user.

1. Aktor mengetik data user yang diedit.
2. Aktor mengubah password.
3. Aktor meminta sistem untuk menyimpan data user yang telah diedit.
4. Sistem melakukan pengecekan terhadap data user yang telah diedit.

E-3 Confirm Password salah

5. Sistem menyimpan data user yang telah diedit ke basis data.
6. Berlanjut ke Basic Flow langkah ke 7.

6. Error Flow

E-1 Data user yang diinputkan aktor tidak lengkap.

1. Sistem memberikan pesan peringatan silahkan lengkapi dahulu.
2. Kembali ke Basic Flow langkah ke 3.

E-2 Nama user yang diinputkan aktor telah ada di sistem.

1. Sistem memberikan pesan peringatan bahwa nama user telah ada di sistem.
2. Kembali ke Basic Flow langkah ke 3.

E-3 Confirm Password salah

1. Sistem memberikan pesan peringatan cek lagi password dan confirm password.
2. Kembali ke Alternative Flow A-1 langkah ke 3.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

Program Studi Teknik Informatika	SKPL – LabAssist	18/ 34
----------------------------------	------------------	--------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

1. Data user di basis data bertambah atau berubah.

4.1.3 Use case Spesification : Jalankan Aplikasi

1. Brief Description

Use Case ini digunakan oleh aktor untuk menjalankan aplikasi tertentu yang di komputer client.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Jalankan Aplikasi.
2. Aktor memasukkan nama atau lokasi aplikasi yang ingin dijalankan
3. Sistem server akan megirimkan perintah ke sisi client
4. Sistem client menerima perintah, nama dan lokasi aplikasi dari sistem server.
5. Sistem client menjalankan aplikasi yang inginkan user pada komputer client.
6. Use Case ini selesai.

5. Alternative Flow

None

6. Error Flow

None

7. PreConditions

1. Use Case Login sudah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Komputer client akan menjalankan aplikasi yang diinginkan user.

4.1.4 Use case Specification : Blok

1. Brief Description

Use Case ini digunakan oleh user untuk melakukan blok terhadap aplikasi tertentu, copy-paste, USB, dan firewall.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Blok.
2. Aktor memilih untuk Blok Aplikasi
 - A-1 Aktor memilih untuk melakukan Blok Aplikasi Selain.
 - A-2 Aktor memilih untuk melakukan Blok Copy-Paste dan Block USB.
 - A-3 Aktor memilih untuk melakukan Blok Firewall.
3. Aktor memilih jenis aplikasi yang akan diblok
4. Sistem server mengirim perintah dan jenis aplikasi yang akan di blok ke sistem client.
5. Sistem client menerima perintah tersebut kemudian memblok aplikasi yang telah dipilih oleh user dari sisi server.
6. Use Case ini selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan Blok Aplikasi Selain.
 1. Aktor memilih jenis aplikasi yang tidak ingin diblok.

2. Sistem server mengirim perintah dan jenis aplikasi lain yang tidak dipilih oleh user.

3. Berlanjut ke Basic Flow langkah ke 5.

A-2 Aktor memilih untuk melakukan Blok Copy-Paste dan Block USB.

1. Aktor meminta sistem untuk memblok copy-paste dan USB pada sisi client.

2. Sistem server akan mengirimkan perintah ke sistem client.

3. Sistem client menerima perintah kemudian memblok copy-paste dan USB.

4. Sistem client melakukan *restart* pada komputer client.

5. Berlanjut ke Basic Flow langkah ke 6.

A-3 Aktor memilih untuk melakukan Blok Firewall.

1. Aktor meminta sistem untuk memblok firewall komputer client.

2. Sistem server akan mengirimkan perintah ke sistem client.

3. Sistem client menerima perintah kemudian memblok firewall dan *me-restart* pada komputer client.

4. Berlanjut ke Basic Flow langkah ke 6.

6. Error Flow

None

7. PreConditions

1. Use Case Login sudah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Aplikasi tertentu, USB, *copy-paste*, atau firewall diblok.

4.1.5 Use case Spesification : Kirim File

1. Brief Description

Use Case ini digunakan oleh user untuk mengirimkan data dari komputer server ke seluruh komputer client.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Pengiriman File.
2. Aktor menentukan lokasi asal file dan lokasi tujuan file akan dikirim.
3. Aktor meminta sistem sever mengirimkan file lokasi asal di server ke lokasi tujuan pada client.
4. Sistem server mengirimkan data-data tersebut ke sistem client.
A-1 Sistem mengecek besar file lebih dari 60 KB
E-1 File telah berubah atau dipindah di sisi server
5. Sistem client menerima data dari server.
6. Use Case ini selesai.

5. Alternative Flow

- A-1 Sistem mengecek besar file lebih dari 60 KB.
1. Sistem server akan memecah file tersebut dan mengirimkannya ke client.
 2. Sistem client akan menggabungkan file tersebut kembali.
 3. Berlanjut ke Basic Flow langkah ke 6.

6. Error Flow

Program Studi Teknik Informatika	SKPL – LabAssist	22/ 34
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

E-1 File telah berubah atau dipindah di sisi server.

1. Sistem memberikan pesan peringatan "Pengiriman file dibatalkan karena file yang ditangani telah berubah".
2. Kembali ke Basic Flow langkah ke 6.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Komputer server menerima file dari server.

4.1.6 Use case Spesification : Kelola Laboratorium

1. Brief Description

Use Case ini digunakan oleh aktor administrator untuk mengelola jadwal pemakaian laboratorium. Aktor dapat melakukan tambah, edit, hapus atau tampil jadwal pemakaian laboratorium.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Kelola Laboratorium.
2. Sistem menampilkan jadwal kegiatan yang ada.
3. Aktor memilih untuk melakukan Tambah Jadwal Harian.
 - A-1 Aktor memilih untuk melakukan Ubah Jadwal Harian.
 - A-2 Aktor memilih untuk melakukan Hapus Jadwal Harian.

Program Studi Teknik Informatika	SKPL – LabAssist	23/ 34
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

A-3 Aktor memilih untuk melakukan Tambah Acara Khusus.

A-4 Aktor memilih untuk melakukan Ubah Jadwal Khusus.

A-5 Aktor memilih untuk melakukan Hapus Acara Khusus.

4. Aktor memasukkan data pemakaian laboratorium.

5. Aktor meminta sistem untuk menyimpan data yang telah dimasukkan.

6. Sistem mengecek data yang telah dimasukkan.

E-1 Data yang diinputkan aktor tidak lengkap.

E-2 Data jadwal yang dimasukkan tabrakan dengan jadwal yang sudah ada.

7. Sistem menyimpan data ke basis data.

8. Use Case ini selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan Ubah Jadwal Harian.

1. Aktor memilih data yang akan diedit.

2. Aktor mengedit data yang telah terpilih.

3. Aktor meminta sistem untuk menyimpan data yang telah diedit.

4. Sistem melakukan pengecekan terhadap data yang telah diedit.

E-2 Data jadwal yang dimasukkan tabrakan dengan jadwal yang sudah ada.

5. Sistem menyimpan data yang telah diedit ke basis data.

6. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan hapus Jadwal Harian.

1. Aktor memilih jadwal yang akan dihapus.
2. Aktor meminta sistem untuk menghapus data Media Masuk yang telah dipilih.
3. Sistem menghapus data yang telah dipilih aktor dari basis data.
4. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan Tambah Acara Khusus.

1. Aktor memasukkan data jadwal pemakaian khusus.
2. Aktor meminta sistem untuk menyimpan data yang telah dimasukkan.
3. Sistem mengecek data yang telah dimasukkan.

E-1 Data yang dimasukkan tidak lengkap

E-2 Jadwal khusus bertabrakan dengan jadwal Harian

4. Berlanjut ke Basic Flow langkah ke 7.

A-4 Aktor memilih untuk melakukan Ubah Jadwal Khusus.

1. Aktor memilih data yang akan diedit.
2. Aktor mengedit data yang telah dipilih.
3. Aktor meminta sistem untuk menyimpan data yang telah diedit.
4. Sistem melakukan pengecekan terhadap data yang telah diedit.

E-2 Data jadwal yang dimasukkan tabrakan dengan jadwal yang sudah ada.

5. Sistem menyimpan data yang telah diedit ke basis data.
6. Berlanjut ke Basic Flow langkah ke 8.

A-5 Aktor memilih untuk melakukan hapus Jadwal Khusus.

1. Aktor memilih data yang akan dihapus.
2. Aktor meminta sistem untuk menghapus data Media Masuk yang telah dipilih.
3. Sistem menghapus data yang telah dipilih aktor dari basis data.
4. Berlanjut ke Basic Flow langkah ke 8.

A-6 Jadwal khusus bertabrakan dengan jadwal Harian

1. Sistem akan menanyakan apakah ingin melakukan pemindahan jam jadwal harian.
2. Jika aktor menjawab ya.

A-7 Jika aktor menjawab tidak

3. Aktor akan memasukkan data pemindahan jadwal harian
4. Sistem akan menanyakan apakah user ingin mencetak jadwal yang tabrakan tadi?
5. Jika aktor menjawab ya.

A-7 Jika aktor menjawab tidak

6. Sistem akan mencetak jadwal yang tabrakan dan pergantian yang telah dimasukkan.
7. Aktor mengedit data yang telah dipilih.
8. Sistem menyimpan data perubahan ke basis data.
9. Berlanjut ke Basic Flow langkah ke 8.

A-7 Jika aktor menjawab tidak

1. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data yang dimasukkan tidak lengkap

1. Sistem memberikan pesan peringatan silahkan lengkapi dahulu.

2. Kembali ke Basic Flow langkah ke 3.

E-2 Data jadwal yang dimasukkan tabrakan dengan jadwal yang sudah ada.

1. Sistem memberikan pesan peringatan sudah ada praktikum hari itu.

2. Kembali ke Alternative Flow A-1 langkah ke 3.

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data Jadwal Pemakaian Laboratorium di basis data bertambah, berubah, atau berkurang.

4.1.7 Use case Spesification : Pantau Data Sistem

1. Brief Description

Use Case ini digunakan oleh aktor administrator untuk memantau data sistem komputer client. Aktor dapat meminta data-data software yang terinstal, hardware, dan proses yang terjadi dari komputer client.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Pantau Data Sistem.

2. Aktor memasukkan alamat IP komputer client yang diinginkan.

3. Aktor memilih untuk melakukan Tampil Data Hardware.

A-1 Aktor memilih untuk melakukan Tampil Data Software.

A-2 Aktor memilih untuk melakukan Tampil Data Process.

4. Sistem server akan mengirimkan perintah pada sistem client tertentu untuk meminta data hardware komputer client.

5. Sistem client menerima perintah dan menanggapi dengan mengirimkan data-data hardware ke sistem server.

6. Sistem server menerima data dari client kemudian menampilkannya ke layar user.

A-3 Aktor memilih untuk melakukan Cari Data.

7. Use Case ini selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan Tampil Data Software.

1. Sistem server akan mengirimkan perintah pada sistem client tertentu untuk meminta data software yang diinstal komputer client.

2. Sistem client menerima perintah dan menanggapi dengan mengirimkan nama software ke sistem server.

3. Berlanjut ke Basic Flow langkah ke 6.

A-2 Aktor memilih untuk melakukan Tampil Data Process.

1. Sistem server akan mengirimkan perintah pada sistem client tertentu untuk meminta proses yang berjalan komputer client.

2. Sistem client menerima perintah dan menanggapi dengan mengirimkan nama proses ke sistem server.

3. Berlanjut ke Basic Flow langkah ke 6.

A-3 Aktor memilih untuk melakukan Cari Data.

1. Aktor memasukkan data yang ingin dicari.

2. Sistem server mencari data yang diinginkan user dari list yang telah ditampilkan ke layar user.

3. Data ditemukan kemudian background data di list berubah warna.

E-1 Data yang dicari tidak ditemukan

4. Berlanjut ke Basic Flow langkah ke 6.

6. Error Flow

E-1 Data yang dicari tidak ditemukan

1. Muncul pesan "Data tidak ditemukan".

2. Berlanjut ke Basic Flow langkah ke 6.

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Pada layar user tertampil data hardware, software, atau proses dari komputer client.

4.1.8 Use case Spesification : Ubah Bahasa

1. Brief Description

Use Case ini digunakan oleh aktor administrator untuk mengubah bahasa pada region and Language di komputer client.

2. Primary Actor

1. User

3. Supporting Actor

Program Studi Teknik Informatika	SKPL – LabAssist	29/ 34
----------------------------------	------------------	--------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Ubah Bahasa.
2. Aktor memilih bahasa yang diinginkan.
3. Aktor meminta sistem untuk mengubah bahasa.
4. Sistem server mengirim perintah dan data bahasa ke sistem client
5. Sistem client menerima perintah kemudian mengubah bahasa komputer client sesuai dengan yang diinginkan aktor.
6. Use Case ini selesai.

5. Alternative Flow

None

6. Error Flow

None

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Bahasa pada Regional and Language berubah.

4.1.9 Use case Spesification : Kelola Aplikasi

1. Brief Description

Use Case ini digunakan oleh aktor petugas untuk mengelola Aplikasi. Aktor dapat melakukan tambah, edit, atau hapus Aplikasi.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

Program Studi Teknik Informatika	SKPL – LabAssist	30/ 34
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Kelola Aplikasi.
2. Sistem menampilkan data aplikasi.
3. Aktor memilih untuk melakukan tambah Aplikasi.
 - A-1 Aktor memilih untuk melakukan ubah Aplikasi.
 - A-2 Aktor memilih untuk melakukan hapus Aplikasi.
4. Aktor memasukkan nama dan jenis aplikasi.
5. Aktor meminta sistem untuk menyimpan data aplikasi yang telah dimasukkan.
6. Sistem mengecek data aplikasi yang telah dimasukkan.
 - E-1 Data Berita yang diinputkan sudah ada
7. Sistem menyimpan data aplikasi ke basis data.
8. Use Case ini selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah Aplikasi.
 1. Aktor memilih aplikasi yang akan diedit.
 2. Aktor mengedit data aplikasi yang telah terpilih.
 3. Aktor meminta sistem untuk menyimpan data Berita yang telah diedit.
 4. Sistem melakukan pengecekan terhadap data Berita yang telah diedit.
 - E-1 Data Berita yang diinputkan sudah ada
 5. Sistem menyimpan data Berita yang telah diedit ke basis data.
 6. Berlanjut ke Basic Flow langkah ke 8.
- A-2 Aktor memilih untuk melakukan hapus Berita.
 1. Aktor memilih Berita yang akan dihapus.
 2. Aktor meminta sistem untuk menghapus data Berita yang telah dipilih.

3. Sistem menghapus data Berita yang telah dipilih aktor dari basis data.

4. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data Berita yang diinputkan sudah ada.

1. Sistem memberikan pesan peringatan data yang anda masukkan sudah ada.

2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data Aplikasi di basis data bertambah, berubah atau berkurang.

4.1.10 Use case Spesification : Kelola Alamat IP Klien

1. Brief Description

Use Case ini digunakan oleh aktor untuk mendapatkan dan menampilkan seluruh alamat IP Klien yang ditangani.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Kelola Alamat IP Klien.

2. Sistem menampilkan antarmuka Kelola Alamat IP Klien.

3. Aktor memasukkan IP broadcast dan meminta alamat IP klien.

4. Sistem server akan mengirim data perintah untuk meminta alamat IP.
5. Sistem klien mendapatkan data tersebut kemudian mengirim alamat IP klien tersebut.
6. Sistem server mendapatkan alamat IP kemudian ditampilkan ke layar aktor.
7. Use Case ini selesai.

5. Alternative Flow

None

6. Error Flow

None

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

Pada layar komputer server ditampilkan seluruh alamat IP klien.

4.1.11 Use case Spesification : Instal Program

1. Brief Description

Use Case ini digunakan oleh aktor untuk menginstal program di seluruh komputer klien.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Instal Program.
2. Aktor memasukkan nama proses instal dan mengecek ke komputer client, apakah sudah siap?
3. Sistem server mengirim perintah ke sistem klien.

Program Studi Teknik Informatika	SKPL – LabAssist	33/ 34
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Sistem klien menerima perintah cek, kemudian mengirim status.
5. Sistem server mengirim key sebagai ganti menekan tombol.
6. Sistem klien menerima key dan menjalankan di proses instal dan mengirim status selanjutnya.
7. Berlanjut ke Basic Flow langkah ke 5.
- A-1 Proses instal selesai
8. Use Case ini selesai.

5. Alternative Flow

A-1 Proses instal selesai

1. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

None

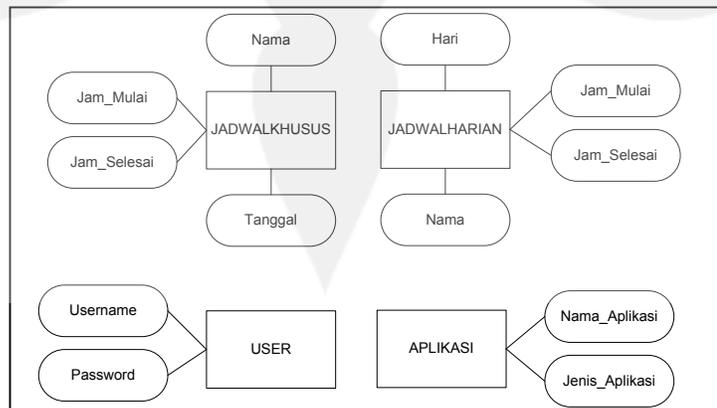
7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

Komputer klien telah terinstal program sesuai dengan perintah aktor.

5 Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

LabAssist (Laboratorium Assist)

Untuk :
Universitas Atma Jaya

Dipersiapkan oleh

Maria Nila Anggia Rini/060704938

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL- LabAssist		1/53
		Revisi		

Program Studi Teknik Informatika	DPPL – LabAssist	1/53
----------------------------------	------------------	------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperik sa oleh								
Disetuj ui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

DAFTAR ISI	4
1 PENDAHULUAN	5
1.1 TUJUAN	5
1.2 RUANG LINGKUP.....	5
1.3 DEFINISI DAN AKRONIM.....	6
SERVER.....	6
1.4 REFERENSI.....	6
2 PERANCANGAN SISTEM (SYSTEM DESIGN)	7
2.1 PERANCANGAN ARSITEKTUR (ARCHITECTURAL DESIGN)	7
2.2 PERANCANGAN RINCI (DETAIL DESIGN)	8
2.2.1 <i>Sequence Diagram</i>	8
2.2.2 <i>Diagram Kelas (Class Diagram)</i>	20
2.2.3 <i>Deskripsi Kelas</i>	21
3 PERANCANGAN DATA.....	40
3.1 DEKOMPOSISI DATA	40
3.1.1 <i>Deskripsi Entitas Data USER</i>	40
3.1.2 <i>Deskripsi Entitas Data Aplikasi</i>	40
3.1.3 <i>Deskripsi Entitas Data Praktikum</i>	40
3.2 PHYSICAL DATA MODEL.....	40
4 PERANCANGAN ANTAR MUKA (UI DESIGN)	41
4.1 ANTAR MUKA LOGIN	41
4.2 ANTAR MUKA MENU UTAMA	41
4.3 ANTAR MUKA KELOLA USER	42
4.4 ANTAR MUKA KELOLA LABORATORIUM.....	43
4.5 ANTAR MUKA DAFTAR APLIKASI	44
4.6 ANTAR MUKA BROADCAST.....	45
4.6.1 <i>Antarmuka Broadcast non Extend</i>	45
4.6.2 <i>Antarmuka Broadcast Extend</i>	48
4.7 ANTAR MUKA SYSTEM CHECK.....	49
4.8 ANTAR MUKA CLIENT	50
4.9 ANTAR MUKA INSTAL PROGRAM.....	51

1 Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Ruang Lingkup

Perangkat Lunak LabAssist dikembangkan dengan tujuan untuk :

1. Mempermudah mengontrol komputer *client* di laboratorim tanpa harus mengecek satu persatu komputer *client*
2. Menangani pengelolaan jadwal praktikum.
3. Mengendalikan komputer *client* untuk melakukan *blok* aplikasi tertentu.
4. Menginstal program di komputer klien dari komputer server.
5. Melihat data sistem dari komputer *client*.
6. Melakukan pengiriman *file* ke komputer *client*.

Dan berjalan pada lingkungan dengan *platform* desktop.

1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan:

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
LabAssist	Perangkat lunak untuk melakukan monitoring dan pengendalian pada komputer klien
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

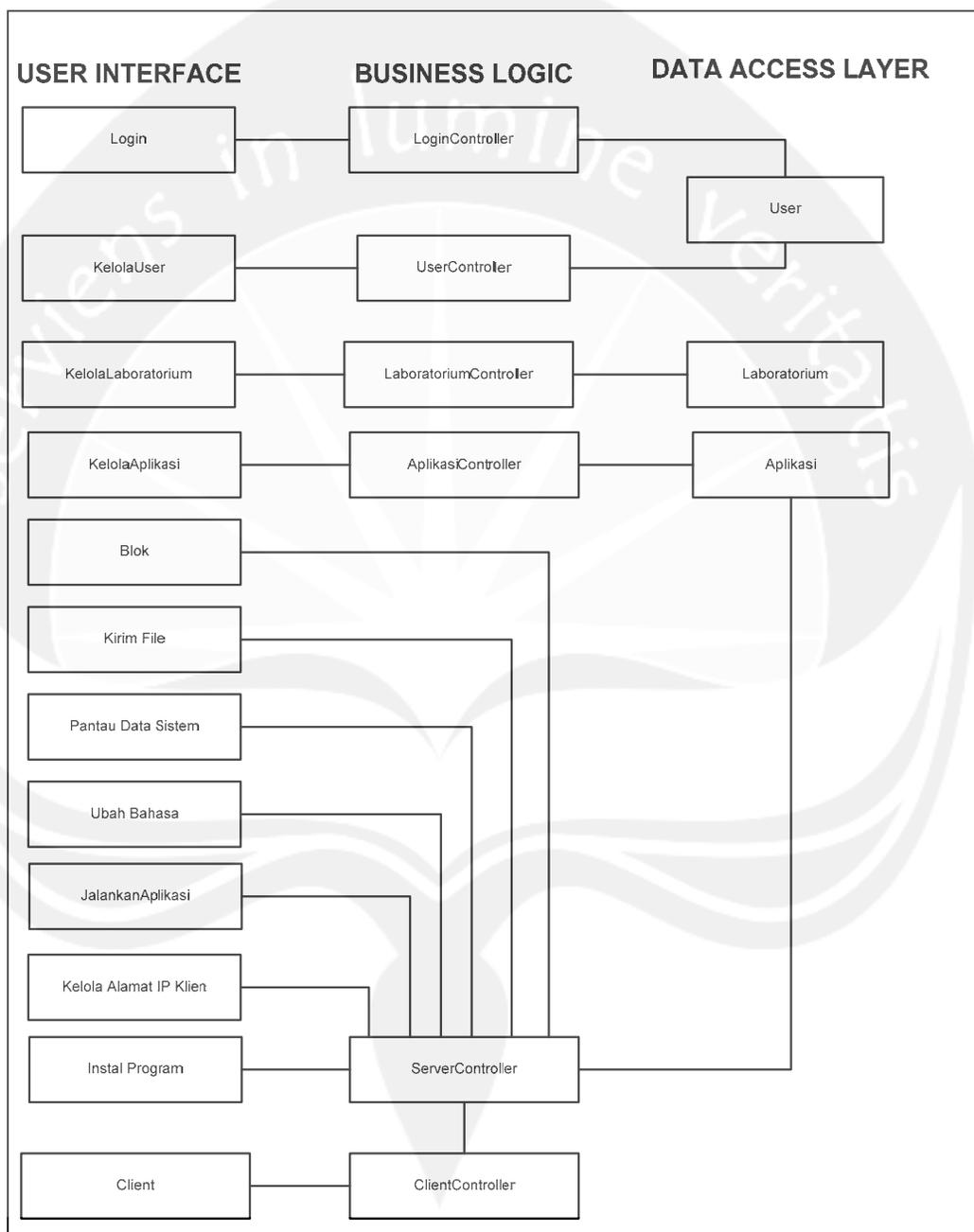
1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Fedrik, Deskripsi Perancangan Perangkat Lunak SPUAJJ (Sistem Pengiriman File Dan Pengendalian Jarak Jauh), Universitas Atma Jaya Yogyakarta, 2010
2. Rini, Maria Nila Anggia, Spesifikasi Kebutuhan Perangkat Lunak LabAssist, Universitas Atma Jaya Yogyakarta, 2011

2 Perancangan Sistem (System Design)

2.1 Perancangan Arsitektur (Architectural Design)

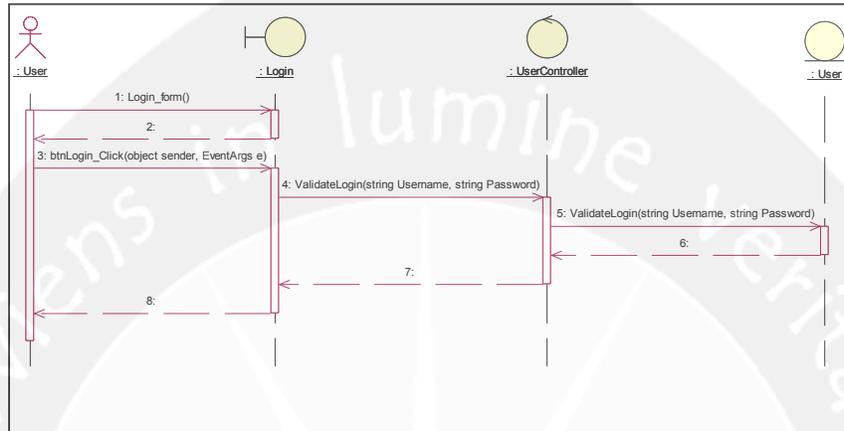


Gambar 1 Rancangan Arsitektur Aplikasi LabAssist

2.2 Perancangan Rinci (Detail Design)

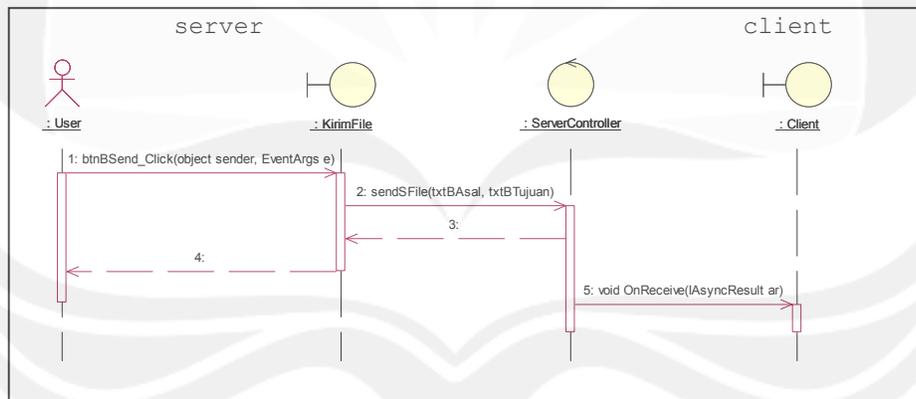
2.2.1 Sequence Diagram

2.2.1.1 Login



Gambar 2.1 Sequence Diagram : Login

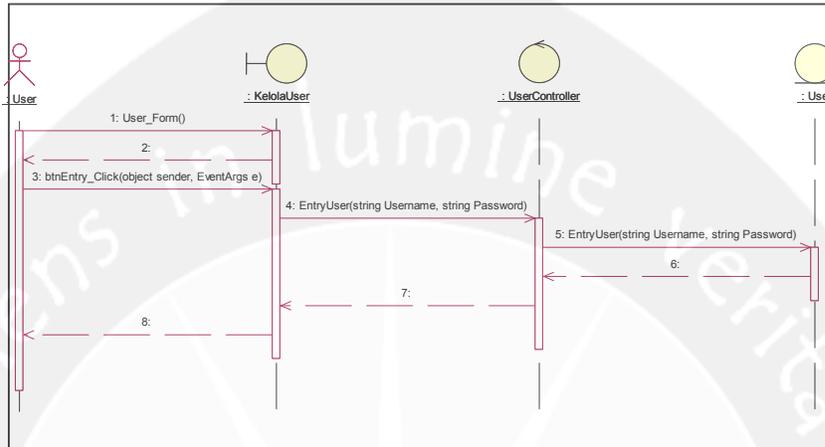
2.2.1.2 Kirim File



Gambar 2.2 Sequence Diagram : Kirim File

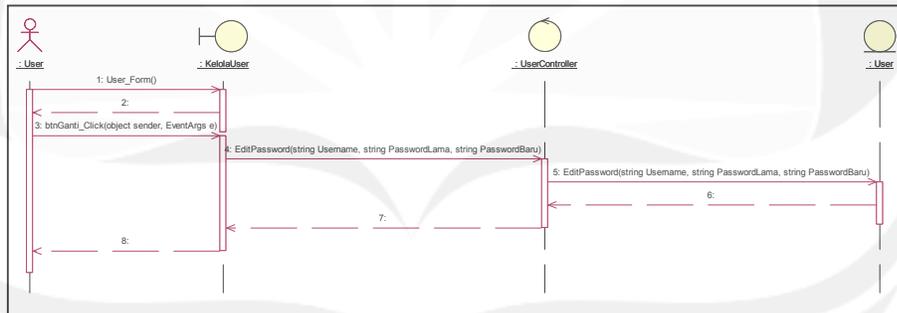
2.2.1.3 Kelola User

2.2.1.3.1 Tambah User



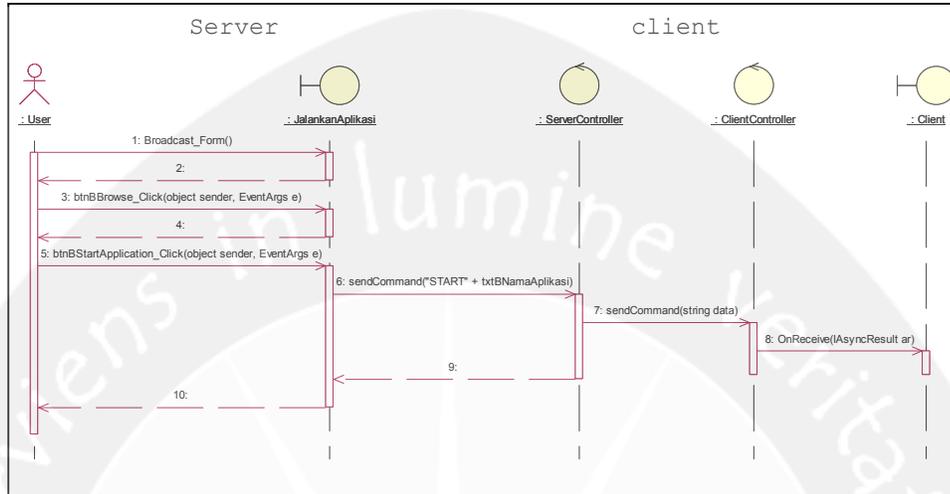
Gambar 2.3 Sequence Diagram : Tambah User

2.2.1.3.2 Edit User



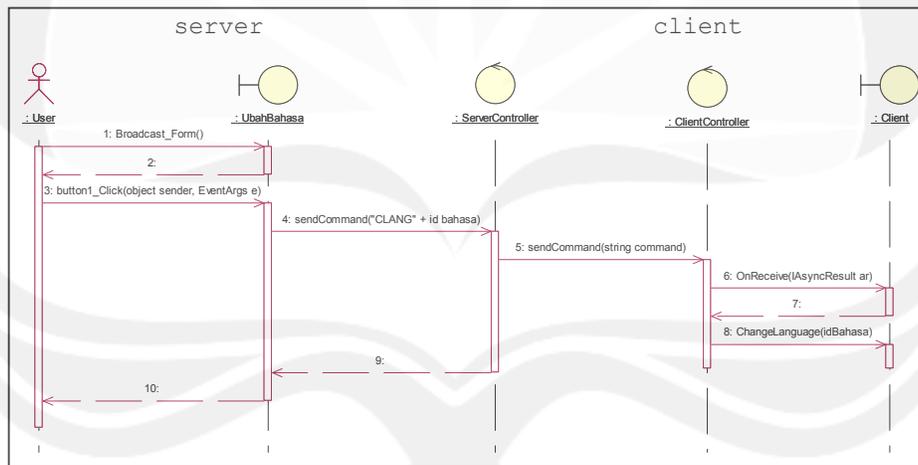
Gambar 2.4 Sequence Diagram : Edit User

2.2.1.4 Jalankan Aplikasi



Gambar 2.5 Sequence Diagram : Jalankan Aplikasi

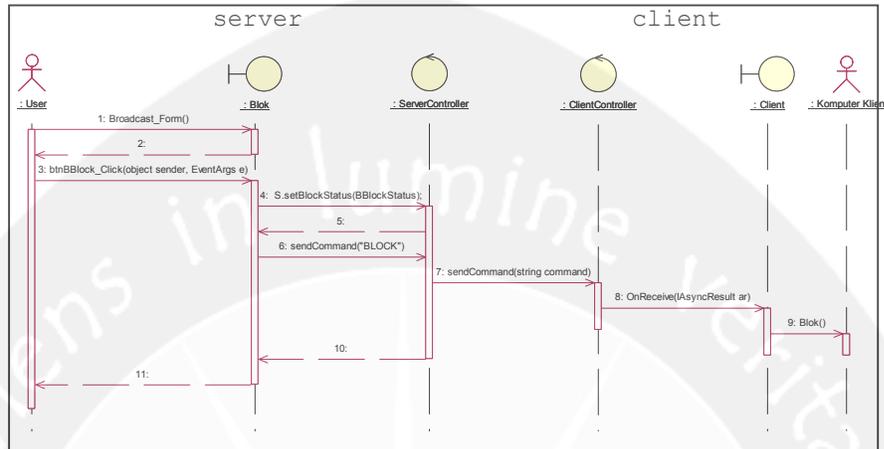
2.2.1.5 Ubah Bahasa



Gambar 2.6 Sequence Diagram : Ubah Bahasa

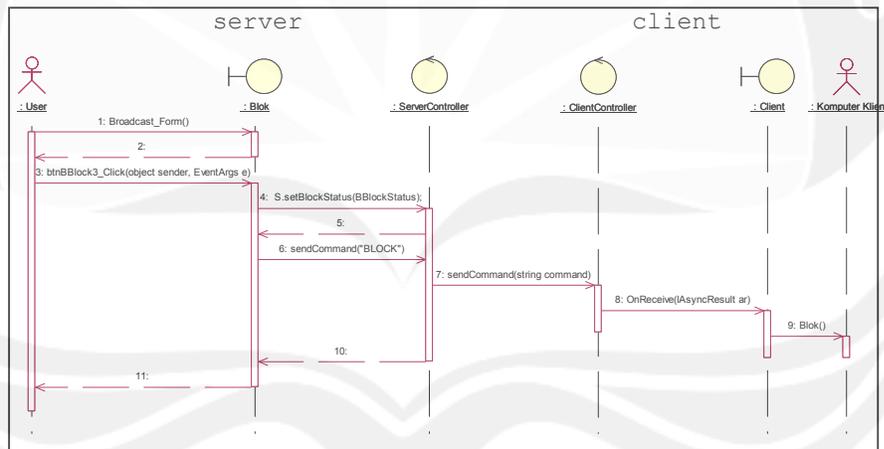
2.2.1.6 Blok

2.2.1.6.1 Blok Aplikasi



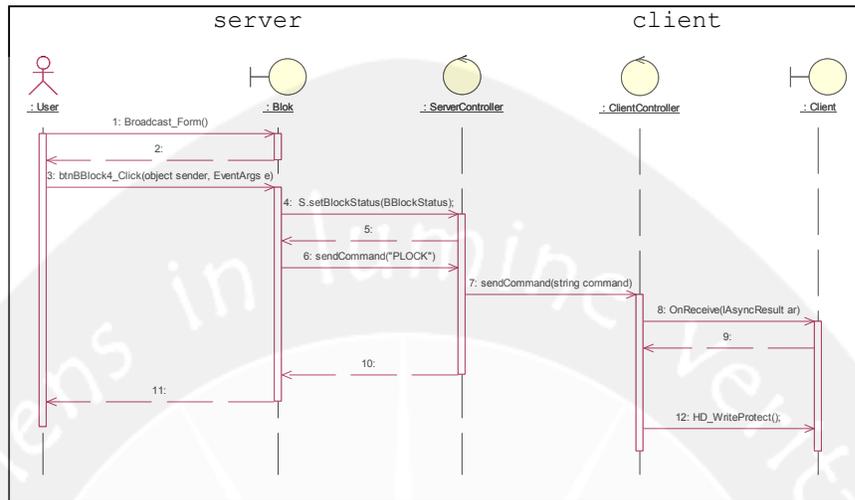
Gambar 2.7 Sequence Diagram : Blok Aplikasi

2.2.1.6.2 Blok Aplikasi Selain



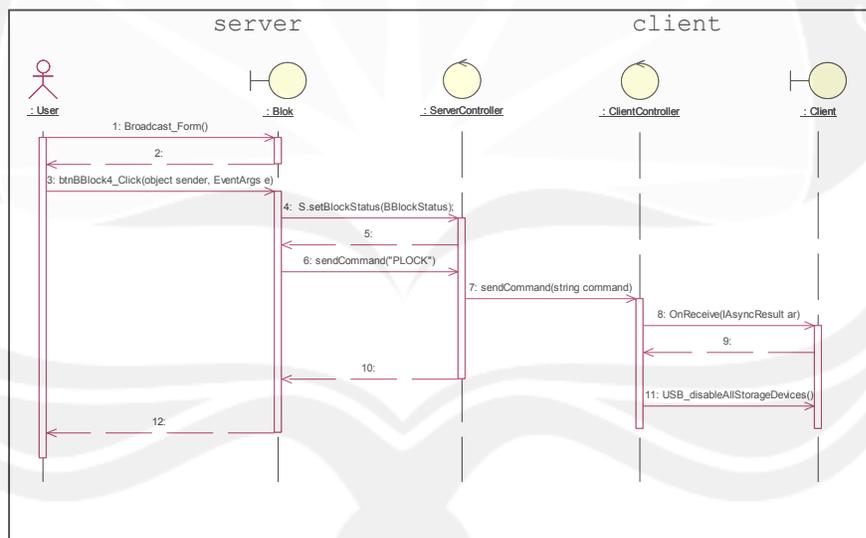
Gambar 2.8 Sequence Diagram : Blok Aplikasi Selain

2.2.1.6.3 Blok Copy-Paste



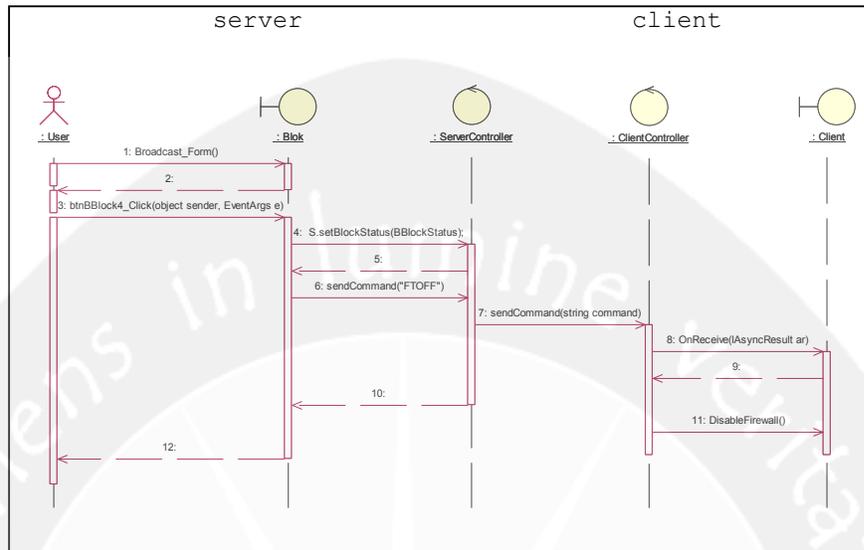
Gambar 2.9 Sequence Diagram : Blok Copy-Paste

2.2.1.6.4 Blok USB



Gambar 2.10 Sequence Diagram : Blok USB

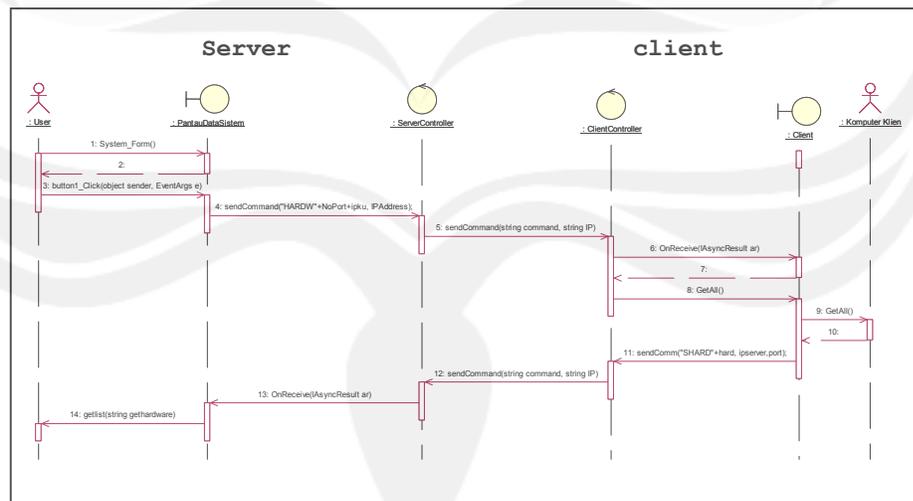
2.2.1.6.1 Blok Firewall



Gambar 2.11 Sequence Diagram : Blok Firewall

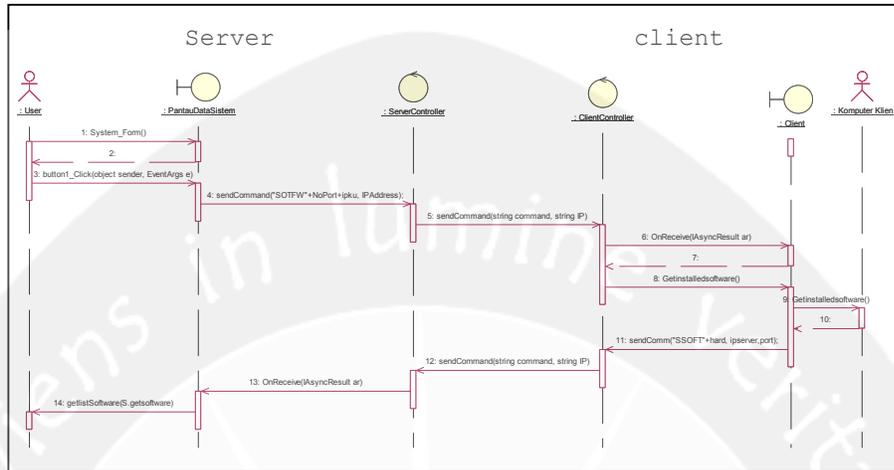
2.2.1.7 Pantau Data Sistem

2.2.1.7.1 Tampil Data Hardware



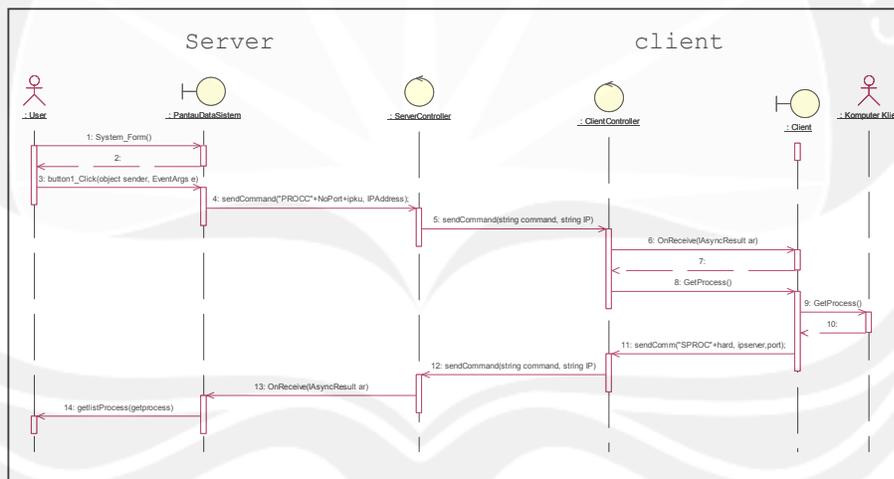
Gambar 2.12 Sequence Diagram : Tambah Media Keluar

2.2.1.7.2 Tampil Data Software



Gambar 2.13 Sequence Diagram : Tampil Data Software

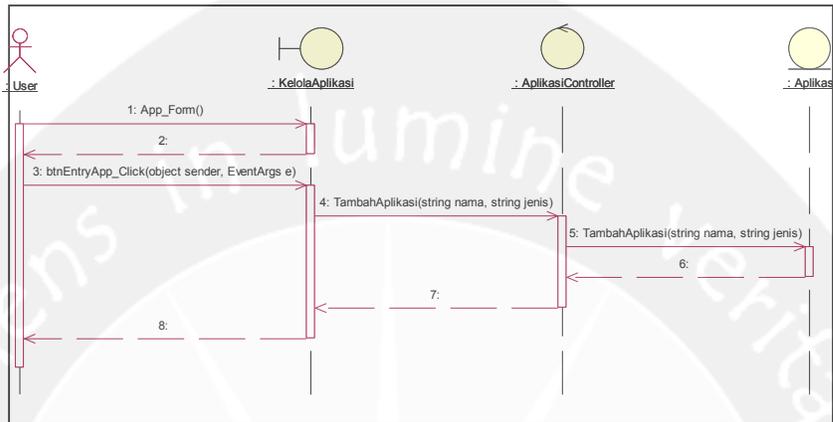
2.2.1.7.3 Tampil Data Process



Gambar 2.14 Sequence Diagram : Tampil Data Process

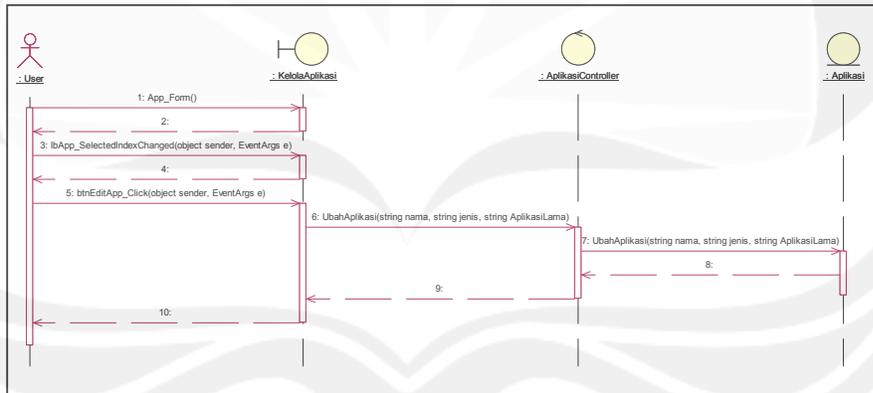
2.2.1.8 Kelola Aplikasi

2.2.1.8.1 Tambah Daftar Aplikasi



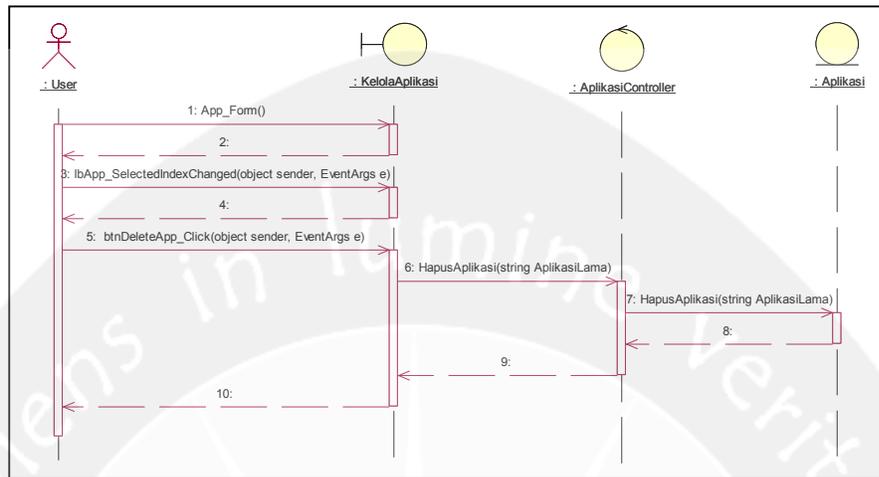
Gambar 2.15 Sequence Diagram : Tambah Daftar Aplikasi

2.2.1.8.2 Ubah Daftar Aplikasi



Gambar 2.16 Sequence Diagram : Ubah Daftar Aplikasi

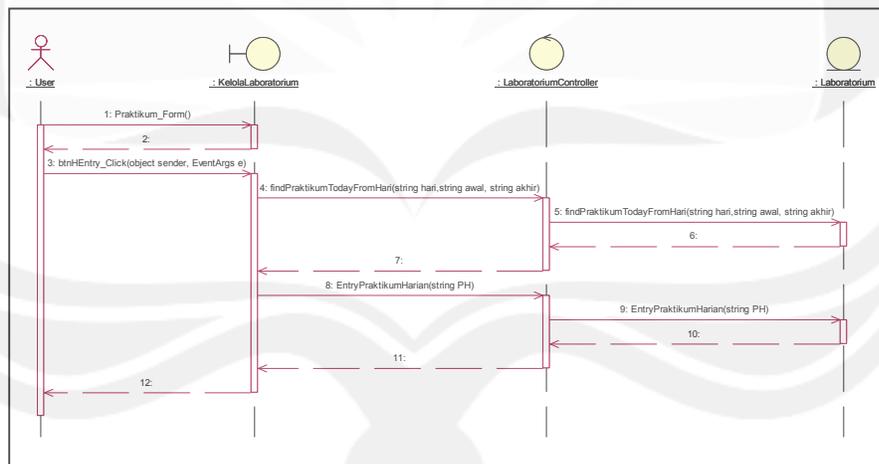
2.2.1.8.3 Hapus Daftar Aplikasi



Gambar 2.17 Sequence Diagram : Hapus Daftar Aplikasi

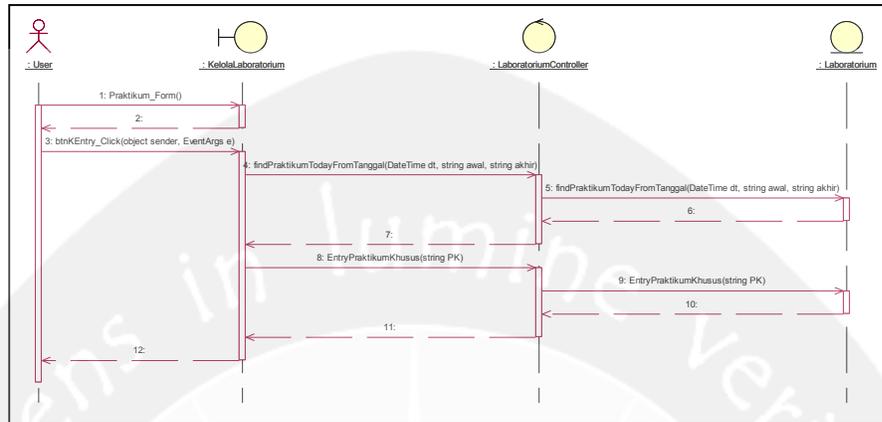
2.2.1.9 Kelola Laboratorium

2.2.1.9.1 Tambah Jadwal Harian



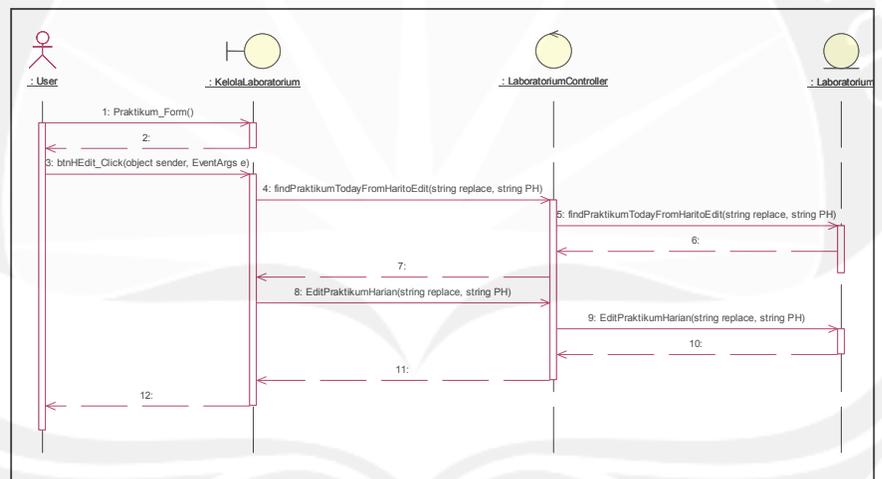
Gambar 2.18 Sequence Diagram : Tambah Jadwal Harian

2.2.1.9.2 Tambah Acara Khusus



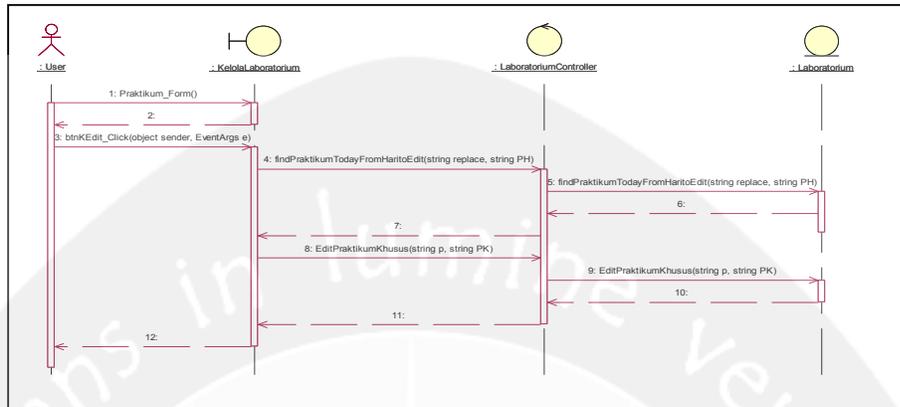
Gambar 2.19 Sequence Diagram : Tambah Acara Khusus

2.2.1.9.3 Fungsi Edit Jadwal Harian



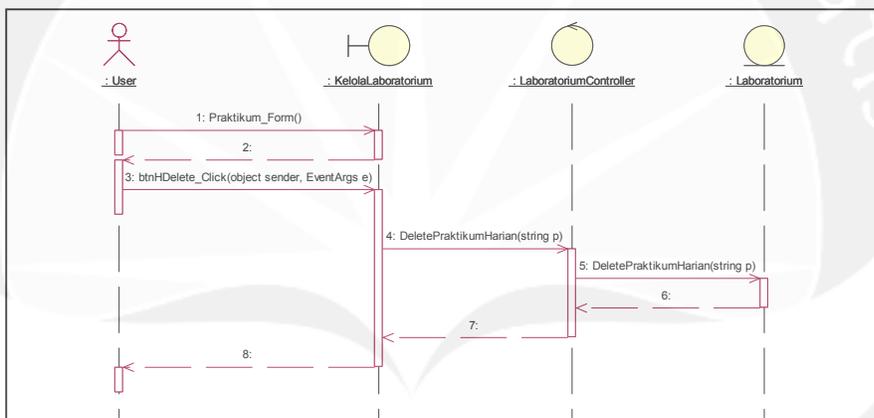
Gambar 2.20 Sequence Diagram : Edit Jadwal Harian

2.2.1.9.4 Fungsi Edit Jadwal Khusus



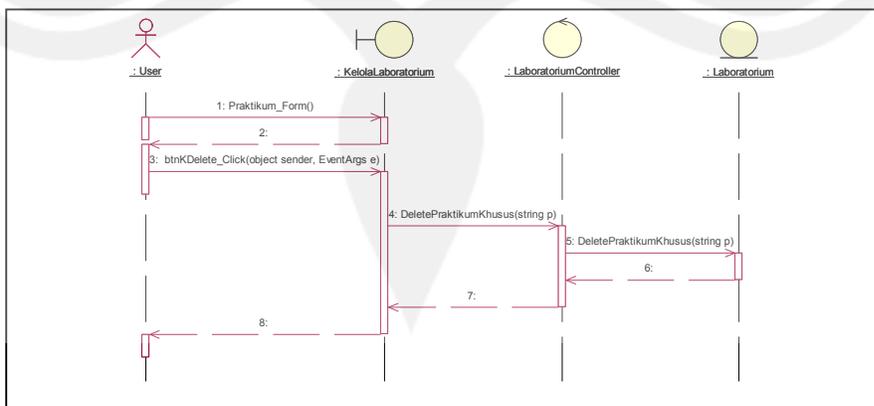
Gambar 2.21 Sequence Diagram : Edit Jadwal Khusus

2.2.1.9.5 Hapus Jadwal Harian



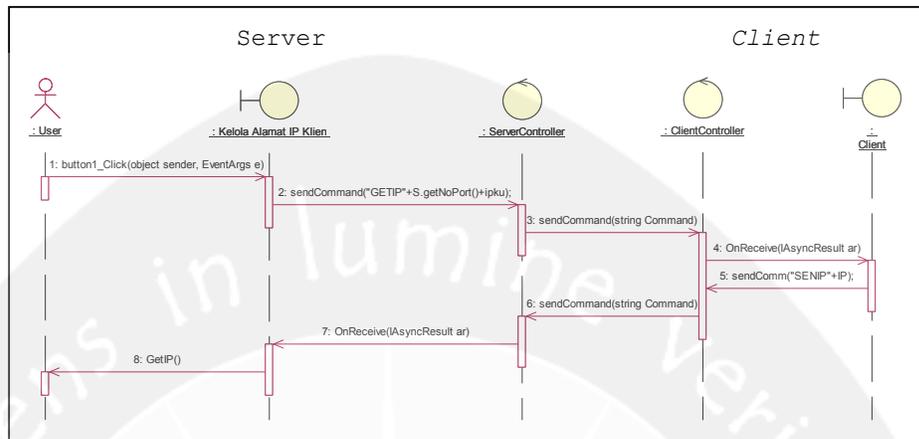
Gambar 2.22 Sequence Diagram : Hapus Jadwal Harian

2.2.1.9.6 Hapus Acara Khusus



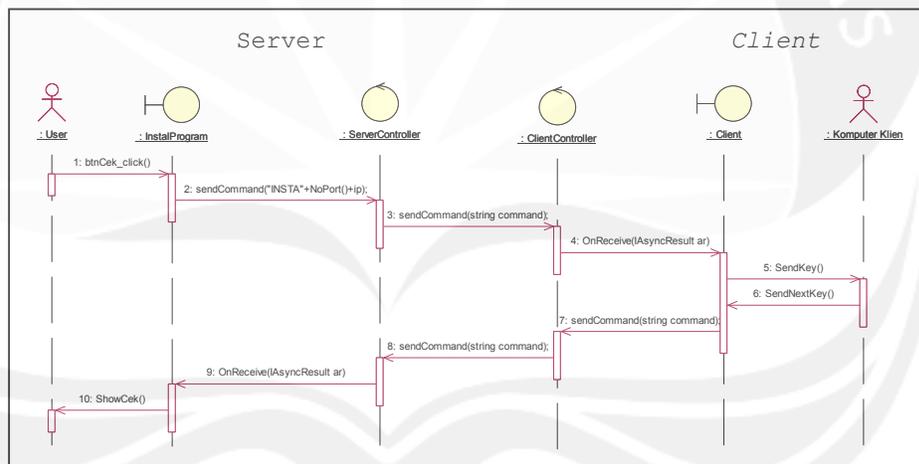
Gambar 2.23 Sequence Diagram : Hapus Acara Khusus

2.2.1.10 Kelola Alamat IP Klien



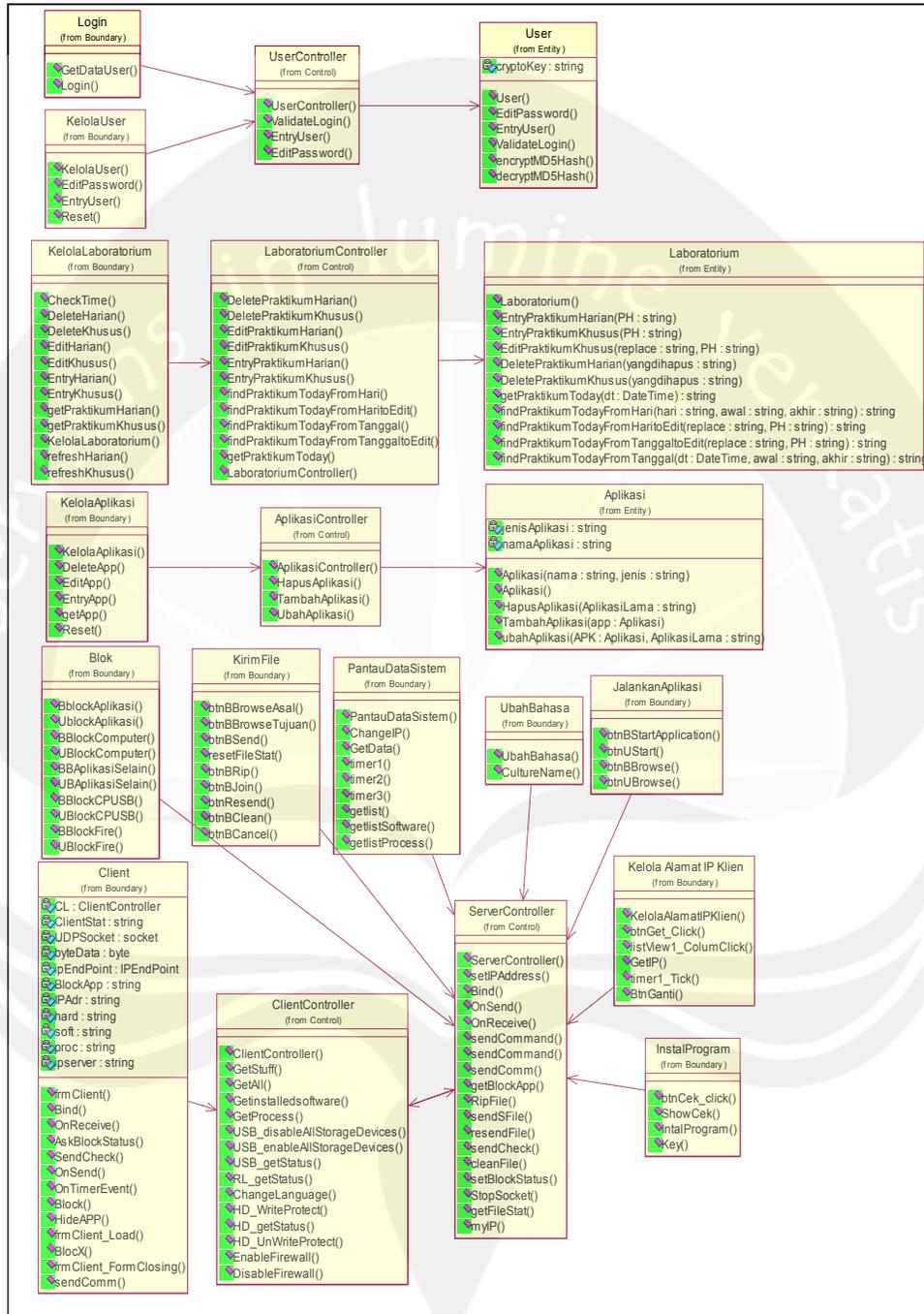
Gambar 2.15 Sequence Diagram : Kelola Alamat IP Klien

2.2.1.11 Instal Program



Gambar 2.15 Sequence Diagram : Instal Program

2.2.2 Diagram Kelas (Class Diagram)



Gambar 2.24 Class Diagram

2.2.3 Deskripsi Kelas

2.2.3.1 Specific Design Class Login

Login	<<boundary>>
+Login () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getDataLogin() Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	

2.2.3.2 Specific Design Class KelolaUser

KelolaUser	<<boundary>>
+KelolaUser() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+Reset() Operasi ini digunakan untuk mengosongkan textbox, combobox.	
+BindGrid() Operasi ini digunakan untuk menampilkan data di grid view.	
+entryUser() Operasi ini digunakan untuk menyimpan data user yang telah diinputkan ke dalam database.	
+editUser() Operasi ini digunakan untuk mengubah password user yang telah diinputkan ke dalam database.	

2.2.3.3 Specific Design Class KelolaLaboratorium

KelolaLaboratorium	<<boundary>>
<pre> +KelolaLaboratorium () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +Reset() Operasi ini digunakan untuk mengosongkan textbox, combobox. +getPraktikumHarian() Operasi ini digunakan untuk menampilkan data. +getPraktikumKhusus() Operasi ini digunakan untuk menampilkan data. +CheckTime(awal,akhir):Boolean Operasi ini digunakan untuk mengecek apakah waktu awal lebih kecil dari waktu akhir. +EntryHarian() Operasi ini digunakan untuk menyimpan data praktikum harian yang telah diinputkan ke dalam database. +EditHarian() Operasi ini digunakan untuk mengubah data praktikum harian yang telah diinputkan ke dalam database. +DeleteHarian() Operasi ini digunakan untuk menghapus data praktikum harian dari database. </pre>	

2.2.3.4 Specific Design Class KelolaAplikasi

KelolaAplikasi	<<boundary>>
<pre> +Reset() Operasi ini digunakan untuk mengosongkan textbox, </pre>	

combobox.

+getApp()

Operasi ini digunakan untuk menampilkan data.

+EntryApp()

Operasi ini digunakan untuk menyimpan data aplikasi yang telah diinputkan ke dalam database.

+EditApp()

Operasi ini digunakan untuk mengubah data aplikasi yang telah diinputkan ke dalam database.

+DeleteApp()

Operasi ini digunakan untuk menghapus data aplikasi dari database.

2.2.3.5 Specific Design Class Blok

Blok	<<boundary>>
+BblockAplikasi() Operasi ini digunakan untuk mengirim perintah blok aplikasi yang diinginkan user ke seluruh klien.	
+UblockAplikasi() Operasi ini digunakan untuk mengirim perintah blok aplikasi yang diinginkan user ke satu klien.	
+BBlockComputer() Operasi ini digunakan untuk mengirim perintah blok komputer ke seluruh klien.	
+UBlockComputer() Operasi ini digunakan untuk mengirim perintah blok komputer ke satu klien.	
+BBAplikasiSelain() Operasi ini digunakan untuk mengirim perintah blok aplikasi selain yang dipilih user ke seluruh klien.	

+UBAplikasiSelain()

Operasi ini digunakan untuk mengirim perintah blok aplikasi selain yang dipilih user ke satu klien.

+BBlockCPUSB()

Operasi ini digunakan untuk mengirim perintah blok copy paste dan USB ke seluruh klien.

+UBlockCPUSB()

Operasi ini digunakan untuk mengirim perintah blok copy paste dan USB ke satu klien.

+BBlockFire()

Operasi ini digunakan untuk mengirim perintah blok firewall ke seluruh klien.

+UBlockFire()

Operasi ini digunakan untuk mengirim perintah blok firewall ke satu klien.

2.2.3.6 Specific Design Class KirimFile

KirimFile	<<boundary>>
+btnBBrowseAsal() Operasi ini digunakan untuk menentukan lokasi asal file.	
+btnBBrowseTujuan() Operasi ini digunakan untuk menentukan lokasi tujuan file.	
+btnBSend() Operasi ini digunakan untuk mengirim file ke komputer klien.	
+resetFileStat() Operasi ini digunakan untuk menyembunyikan tombol Rip, join, dan clean.	

+btnBRip()

Operasi ini digunakan untuk melakukan RIP terhadap file yang akan dikirim.

+ btnBJoin()

Operasi ini digunakan untuk menggabungkan file yang tadi dipisah.

+btnResend()

Operasi ini digunakan untuk mengirim ulang file ke klien.

+btnBClean()

Operasi ini digunakan untuk mengirimkan perintah clean ke klien

+btnBCancel()

Operasi ini digunakan untuk mengosongkan textbox.

2.2.3.7 Specific Design Class PantauDataSistem

PantauDataSistem	<<boundary>>
<p>+PantauDataSistem() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+ChangeIP() Operasi ini digunakan untuk mengubah Alamat IP klien.</p> <p>+GetData() Operasi ini digunakan untuk mengirim perintah ke komputer klien untuk mengirim data sistem klien. Kemudian menerima data sistem dari klien dan menampilkannya ke komputer server.</p> <p>+timer1() Operasi ini digunakan untuk menampilkan data hardware yang diterima dari klien selama timer1 dinyalakan.</p>	

```
+timer2()
```

Operasi ini digunakan untuk menampilkan data software yang diterima dari klien selama timer1 dinyalakan.

```
+timer3()
```

Operasi ini digunakan untuk menampilkan data process yang diterima dari klien selama timer1 dinyalakan.

```
+getList(kata)
```

Operasi ini digunakan untuk menampilkan data hardware ke listview

```
+getListSoftware(kata)
```

Operasi ini digunakan untuk menampilkan data software ke listview

```
+getListProcess(kata)
```

Operasi ini digunakan untuk menampilkan data process ke listview

2.2.3.8 Specific Design Class UbahBahasa

UbahBahasa	<<boundary>>
<pre>+UbahBahasa()</pre> <p>Operasi ini digunakan untuk mengirim perintah ke seluruh komputer klien untuk mengubah bahasa sesuai dengan keinginan klien.</p> <pre>+CultureName(key):string</pre> <p>Operasi ini digunakan untuk mengubah bahasa yang dari combo box dengan nama id yang dikenal komputer.</p>	

2.2.3.9 Specific Design Class JalankanAplikasi

JalankanAplikasi	<<boundary>>
<pre>+btnBStartApplication()</pre>	

Operasi ini digunakan untuk mengirimkan perintah ke klien untuk menjalankan aplikasi tertentu (untuk seluruh komputer klien).

+btnUStart()

Operasi ini digunakan untuk mengirimkan perintah ke klien untuk menjalankan aplikasi tertentu (untuk satu komputer klien).

+btnBBrowse()

Operasi ini digunakan untuk mencari lokasi aplikasi yang diinginkan untuk seluruh komputer klien).

+btnUBrowse()

Operasi ini digunakan untuk mencari lokasi aplikasi yang diinginkan untuk satu komputer klien).

2.2.3.10 Specific Design Class Client

Client	<<boundary>>
<p>-CL:ClientController</p> <p>Atribut ini digunakan untuk memanggil kelas ClientController.</p> <p>-ClientStat:string</p> <p>Atribut ini digunakan untuk menampung status klien.</p> <p>-UDPSocket:socket</p> <p>Atribut ini digunakan untuk melakukan koneksi dengan protokol UDP.</p> <p>-byteData:byte</p> <p>Atribut ini digunakan untuk menampung data yang dikirimkan user.</p> <p>-ipEndPoint: IPEndPoint</p> <p>Atribut ini digunakan untuk mengkombinasikan alamat IP dan no port.</p> <p>-BlockApp:string</p>	

Atribut ini digunakan untuk menampung aplikasi yang ingin diblok.

-IPAdr:string

Atribut ini digunakan untuk menampung alamat IP.

-hard:string

Atribut ini digunakan untuk menampung data hardware komputer.

-soft:string

Atribut ini digunakan untuk menampung data software komputer.

-proc:string

Atribut ini digunakan untuk menampung data process komputer.

-ipserver:string

Atribut ini digunakan untuk menampung alamat IP komputer klien.

+frmClient()

Operasi ini merupakan konstruktor.

+Bind()

Operasi ini digunakan untuk membundel sebuah *socket* dengan alamat IP address dan nomor port.

+OnReceive(ar)

Operasi ini digunakan untuk menerima data yang dikirim oleh komputer server.

+AskBlockStatus()

Operasi ini digunakan untuk mengirim data berupa permintaan kepada server untuk mengirimkan status blok aplikasi.

+SendCheck(Chk)

Operasi ini digunakan untuk mengirimkan status pengiriman file.

+OnSend(ar)
 Operasi ini digunakan untuk mengirim suatu data ke komputer server.

+OnTimerEvent(sender,e)
 +getIPJaringan()
 Operasi ini digunakan untuk mencari IP jaringan.

+Block()
 Operasi ini digunakan untuk melakukan blok terhadap aplikasi tertentu.

+HideAPP()
 Operasi ini digunakan untuk menyembunyikan sistem klien.

+BlocX()
 Operasi ini digunakan untuk mem-blok komputer klien.

+frmClient_Load(sender, e)
 Operasi ini dijalankan ketika form ini dijalankan.

+frmClient_FormClosing(sender, e)
 Operasi ini dijalankan ketika form ini ditutup.

+sendComm(Command, IPAddr, noPort)
 Operasi ini digunakan untuk mengirimkan data ke server.

2.2.3.11 Kelola Alamat IP Klien

UbahBahasa	<<boundary>>
-ipku:string Atribut ini digunakan untuk menyimpan alamat IP komputer server	
+KelolaAlamatIPKlien Operasi ini merupakan konstruktor. +BtnGanti(sender, e) Operasi ini digunakan untuk mengganti IP Broadcast. +btnGet_Click(sender, e)	

Operasi ini digunakan untuk mengirim permintaan no IP ke komputer klien.

+listView1_ColumClick(sender, e)

Operasi ini digunakan untuk mengurutkan data di listview

+timer1_Tick(sender, e)

Operasi ini digunakan untuk menampilkan alamat IP selama timer dinyalakan.

+GetIP()

Operasi ini digunakan untuk menampilkan alamatIP klien di listview.

2.2.3.12 Specific Design Class InstalProgram

KirimFile	<<boundary>>
<p>+IntalPrgram()</p> <p>Operasi ini merupakan konstruktor kelas ini.</p> <p>+ btnCek_click()</p> <p>Operasi ini digunakan untuk mengirimkan key dan mendapatkan status instal program di komputer klien.</p> <p>+ ShowCek()</p> <p>Operasi ini digunakan untuk menampilkan status penginstalan program ke layar user.</p> <p>+ Key()</p> <p>Operasi ini digunakan untuk key yang akan dikirimkan ke komputer klien sebagai ganti penekanan tombol.</p>	

2.2.3.13 Specific Design Class UserController

UserController	<<Control>>
<p>+ValidateLogin (Username, Password)</p>	

Operasi ini digunakan untuk mengecek apakah username dan password yang dimasukkan user ada dalam database.

+EntryUser(Username, Password)

Operasi ini digunakan untuk menyimpan username dan password yang telah diinputkan ke dalam database.

+EditPassword(Username, PasswordLama, PasswordBaru)

Operasi ini digunakan untuk mengubah data password yang telah diinputkan ke dalam database.

2.2.3.14 Specific Design Class LaboratoriumController

LaboratoriumController	<<Control>>
<p>+findPraktikumTodayFromHari(hari, awal, akhir):string Operasi ini digunakan untuk mengetahui praktikum hari ini berdasarkan hari.</p> <p>+findPraktikumTodayFromHaritoEdit(replace, PH):string Operasi ini digunakan untuk mengetahui praktikum hari ini berdasarkan hari untuk melakukan edit.</p> <p>+EditPraktikumHarian(replace, PH) Operasi ini digunakan untuk mengubah data praktikum harian yang telah diinputkan ke dalam database.</p> <p>+EntryPraktikumHarian(PH) Operasi ini digunakan untuk menyimpan data praktikum harian yang telah diinputkan ke dalam database</p> <p>+EntryPraktikumKhusus(PK) Operasi ini digunakan untuk menyimpan data praktikum khusus yang telah diinputkan ke dalam database.</p> <p>+findPraktikumTodayFromTanggal(dt, awal, akhir):string Operasi ini digunakan untuk mengetahui praktikum hari ini berdasarkan tanggal.</p> <p>+findPraktikumTodayFromTanggaltoEdit(p, PK):string</p>	

Operasi ini digunakan untuk mengetahui praktikum hari ini berdasarkan tanggal untuk melakukan edit.

+EditPraktikumKhusus (praktikum, PK)

Operasi ini digunakan untuk mengubah data praktikum khusus yang telah diinputkan ke dalam database.

+DeletePraktikumHarian (Praktikum)

Operasi ini digunakan untuk menghapus data praktikum harian dari database.

+DeletePraktikumKhusus (Praktikum)

Operasi ini digunakan untuk menghapus data praktikum khusus dari database.

+getPraktikumToday (dtime):string

Operasi ini digunakan untuk mengetahui praktikum hari ini

2.2.3.15 Specific Design Class AplikasiController

AplikasiController	<<Control>>
<p>+TambahAplikasi (nama, jenis) Operasi ini digunakan untuk menyimpan data aplikasi yang telah diinputkan ke dalam database.</p> <p>+ UbahAplikasi (nama, jenis, AplikasiLama) Operasi ini digunakan untuk mengubah data praktikum aplikasi yang telah diinputkan ke dalam database.</p> <p>+ HapusAplikasi (AplikasiLama) Operasi ini digunakan untuk menghapus data aplikasi dari database.</p>	

2.2.3.16 Specific Design Class ServerController

ServerController	<<Control>>
<p>+UDPSocket:Socket</p>	

Atribut ini digunakan untuk socket UDP.

+ epServer:EndPoint

Atribut ini digunakan untuk titik akhir UDP.

+byteData:byte[]

Atribut ini digunakan untuk menyimpan data yang akan dikirim ke client.

+BlockStatus:string

Atribut ini digunakan untuk menyimpan data blok status.

+IPAddr:string;

Atribut ini digunakan untuk menyimpan alamat IP.

+NoPort:string

Atribut ini digunakan untuk menyimpan nomor port

+FileStat:string

Atribut ini digunakan untuk menyimpan status file.

+BlockApp:string

Atribut ini digunakan untuk menyimpan data untuk blok aplikasi

+gethardware:string

Atribut ini digunakan untuk menyimpan data hardware dari klien.

+getsoftware:string

Atribut ini digunakan untuk menyimpan data software dari klien.

+getprocess:string

Atribut ini digunakan untuk menyimpan data proses dari klien.

+ServerController(string IPAddr, string NP)

Operasi ini merupakan konstruktor, yang menginisialisasi atribut kelas ini dan memberi nilai untuk IPAddr dan NoPort.

+setIPAddress(IPAddr)

Operasi ini digunakan untuk memberi nilai pada atribut IPAddr.

+Bind()

Operasi ini digunakan untuk koneksi UDP.

+OnSend(ar)

Operasi ini digunakan untuk mengirim data.

+OnReceive(ar)

Operasi ini digunakan untuk menerima data dari client.

+sendCommand(Command)

Operasi ini digunakan untuk mengirim perintah ke client.

+sendCommand(Command, IPAddr)

Operasi ini digunakan untuk mengirim perintah ke client.

+sendComm(Command, IPAddr)

Operasi ini digunakan untuk mengirim perintah ke client.

+getBlockApp(BlckStatus):string

Operasi ini digunakan untuk mendapatkan daftar aplikasi dari database untuk melakukan blok.

+RipFile(AsalFile)

Operasi ini digunakan untuk memotong file yang akan dikirim ke klien.

+sendSFile(AsalFile, TujuanFile)

Operasi ini digunakan untuk mengirim file ke klien.

+resendFile(AsalFile1, AsalFile2, TujuanFile)

Operasi ini digunakan untuk mengirim ulang file ke klien.

+ sendCheck>NamaFile, TujuanFile)

Operasi ini digunakan untuk mengecek pengiriman file.

+cleanFile()

Operasi ini digunakan untuk menghapus data pengiriman.
`+setBlockStatus(String Status)`
 Operasi ini digunakan untuk memberi nilai pada atribut `BlockStatus`.
`+StopSocket()`
 Operasi ini digunakan untuk menutup koneksi.
`+getFileStat()`
 Operasi ini digunakan untuk mengambil nilai atribut `FileStat`.
`+myIP():string`
 Operasi ini digunakan untuk mendapatkan nomor IP komputer server.

2.2.3.17 Specific Design Class ClientController

ClientController	<<Control>>
<code>+ClientController()</code> Operasi ini merupakan konstruktor, yang menginisialisasi atribut kelas ini dan memberi nilai untuk <code>IPAdd</code> dan <code>NoPort</code> . <code>+GetStuff(queryObject, nama, cari):string</code> Operasi ini berisi query untuk memdapatkan data hardware. <code>+GetAll():string</code> Operasi ini digunakan untuk mendapatkan hardware klien. <code>+Getinstalledsoftware():string</code> Operasi ini digunakan untuk mendapatkan software klien. <code>+GetProcess():string</code> Operasi ini digunakan untuk mendapatkan process klien. <code>+USB_disableAllStorageDevices()</code> Operasi ini digunakan untuk menonaktifkan USB	

+USB_enableAllStorageDevices()

Operasi ini digunakan untuk mengaktifkan USB

+USB_getStatus()

Operasi ini digunakan untuk mendapatkan status USB dari register.

+RL_getStatus()

Operasi ini digunakan untuk mendapatkan status bahasa dari register.

+ChangeLanguage(bhs)

Operasi ini digunakan untuk mengganti bahasa.

+HD_getStatus()

Operasi ini digunakan untuk mendapatkan status harddisk dari register.

+HD_WriteProtect()

Operasi ini digunakan untuk mengaktifkan sifat "write protect" pada harddisk

+HD_UnWriteProtect()

Operasi ini digunakan untuk menonaktifkan sifat "write protect" pada harddisk

+EnableFirewall()

Operasi ini digunakan untuk mengaktifkan firewall.

+DisableFirewall()

Operasi ini digunakan untuk menonaktifkan firewall.

2.2.3.18 Specific Design Class User

User	<<entity>>
-cryptoKey:string	
Atribut ini merupakan atribut static untuk melakukan enkripsi dan dekripsi password.	
+User()	
Default konstruktor, digunakan untuk inisialisasi semua	

attribute dari kelas ini.

+EntryUser(Username, Password)

Operasi ini digunakan untuk menyimpan username dan password yang telah diinputkan ke dalam database.

+ValidateLogin(Username, Password):Boolean

Operasi ini digunakan untuk mengecek apakah username dan password yang dimasukkan user ada dalam database.

+EditPassword(Username, PasswordLama, PasswordBaru)

Operasi ini digunakan untuk mengubah data password yang telah diinputkan ke dalam database.

+encryptMD5Hash(password):string

Operasi ini digunakan untuk melakukan enkripsi terhadap data password.

+decryptMD5Hash(password):string

Operasi ini digunakan untuk melakukan dekripsi terhadap data password.

2.2.3.19 Specific Design Class Aplikasi

Program	<<entity>>
- namaAplikasi:string	Atribut ini digunakan untuk menyimpan data nama aplikasi.
-jenisAplikasi: String	Atribut ini digunakan untuk menyimpan data jenis aplikasi.
+Aplikasi()	Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.
+Aplikasi(nama, jenis)	konstruktor, digunakan untuk memberi nilai atribut namaAplikasi dan jenisAplikasi.

+ TambahAplikasi(aplikasi)

Operasi ini digunakan untuk menyimpan data aplikasi yang telah diinputkan ke dalam database.

+ubahAplikasi(APK, AplikasiLama)

Operasi ini digunakan untuk mengubah data aplikasi dari database dengan data masukan user yang baru.

+HapusAplikasi(AplikasiLama)

Operasi ini digunakan untuk menghapus data aplikasi tertentu ke dalam database.

2.2.3.20 Specific Design Class Laboratorium

Media	<<entity>>
<p>+Laboratorium() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+EntryPraktikumHarian(PH) Operasi ini digunakan untuk menyimpan data praktikum harian yang telah diinputkan ke dalam database.</p> <p>+EditPraktikumHarian(replace, PH) Operasi ini digunakan untuk mengubah data praktikum harian dari database dengan data inputan user.</p> <p>+EntryPraktikumKhusus(PH) Operasi ini digunakan untuk menyimpan data praktikum khusus yang telah diinputkan ke dalam database.</p> <p>+EditPraktikumKhusus(replace, PH) Operasi ini digunakan untuk mengubah data praktikum khusus dari database dengan data inputan user.</p> <p>+DeletePraktikumHarian(yangdihapus) Operasi ini digunakan untuk menghapus data praktikum harian tertentu ke dalam database.</p>	

+DeletePraktikumKhusus (yangdihapus)

Operasi ini digunakan untuk menghapus data praktikum khusus tertentu ke dalam database.

+convertDay(hari)

Operasi ini digunakan untuk mengubah hari dari bahasa Indonesia ke bahasa Inggris.

+getPraktikumToday(dt):string

Operasi ini digunakan untuk mengambil jadwal penggunaan laboratorium hari ini.

+findPraktikumTodayFromHari(hari, awal, akhir):string

Operasi ini digunakan untuk mengambil jadwal penggunaan laboratorium hari ini berdasarkan hari dan waktu pelaksanaan.

+findPraktikumTodayFromHaritoEdit(replace, PH):string

Operasi ini digunakan untuk mengambil jadwal penggunaan laboratorium hari ini berdasarkan hari dan waktu pelaksanaan untuk melakukan edit.

+findPraktikumTodayFromTanggaltoEdit(replace, PH)

Operasi ini digunakan untuk mengambil jadwal penggunaan laboratorium hari ini berdasarkan tanggal pelaksanaan untuk melakukan edit.

+findPraktikumTodayFromTanggal(DateTime dt, string awal, string akhir)

Operasi ini digunakan untuk mengambil jadwal penggunaan laboratorium hari ini berdasarkan tanggal.

3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Data USER

Nama	Tipe	Panjang	Keterangan
Username	Varchar	20	Username admin
password	Varchar	20	Password admin

3.1.2 Deskripsi Entitas Data Aplikasi

Nama	Tipe	Panjang	Keterangan
Nama_Aplikasi	VarChar	50	Nama dari Aplikasi
Jenis_Aplikasi	VarChar	50	Jenis dari Aplikasi

3.1.3 Deskripsi Entitas Data Praktikum

Nama	Tipe	Panjang	Keterangan
tanggal	VarChar	10	Tanggal praktikum
Hari	VarChar	10	hari praktikum
Jam mulai	VarChar	5	Jam dimulainya praktikum
Jam selesai	VarChar	5	Jam selesai praktikum
nama	VarChar	50	Nama praktikum

3.2 Physical Data Model

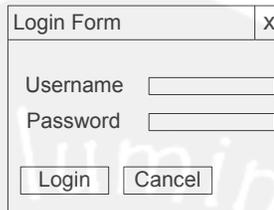
USER	APLIKASI
USERNAME varchar(20) PASSWORD varchar(20)	NAMA_APLIKASI varchar(50) JENIS_APLIKASI varchar(50)

JADWAL_KHUSUS	JADWAL_KHUSUS
TANGGAL varchar(10) JAM_MULAI varchar(5) JAM_SELESAI varchar(5) Nama varchar(50)	HARI varchar(10) JAM_MULAI varchar(5) JAM_SELESAI varchar(5) Nama varchar(50)

Gambar 3 Physical Data Model

4 Perancangan Antar Muka (UI Design)

4.1 Antarmuka Login



Login Form	X
Username	<input type="text"/>
Password	<input type="text"/>
Login	Cancel

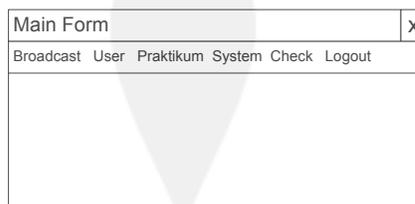
Gambar 4.1 Rancangan Antarmuka Login

Antarmuka Gambar 4.1 digunakan untuk melakukan proses login ke dalam sistem LabAssist. Untuk mendapat akses masuk ke dalam sistem, user harus memasukkan *username*, *password* dengan benar pada textbox yang telah disediakan.

Tombol login berfungsi sebagai validasi username dan password yang diinputkan dengan data username dan password yang telah tersimpan di database. Jika data username dan password benar atau cocok maka user akan masuk ke dalam sistem, sebaliknya jika id dan password salah atau tidak cocok maka akan diberikan pesan peringatan.

Tombol cancel berfungsi untuk membatalkan login dan menutup form login.

4.2 Antarmuka Menu Utama



Main Form	X
Broadcast User Praktikum System Check Logout	

Gambar 4.2 Rancangan Antarmuka Menu Utama

Antarmuka Gambar 4.2 merupakan menu utama user berhasil melakukan login yang berfungsi untuk

Program Studi Teknik Informatika	DPPL – LabAssist	41/53
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

menampilkan menu-menu yang akan menuju ke antarmuka lainnya. Menu Boardcast untuk menampilkan Form Broadcast. Menu User untuk menampilkan Form Pengelolaan User. Menu Praktikum untuk menampilkan Form Pengelolaan Jadwal Praktikum. System Check untuk menampilkan Form System Check. Menu Logout untuk keluar dari sistem dan proses logout kemudian kembali ke form Login.

4.3 Antarmuka Kelola User

Gambar 4.3 Rancangan Antarmuka Pengelolaan User

Antarmuka Gambar 4.3 memiliki tab Entry dan tab Edit.

Tab Entry digunakan untuk memasukkan data user ke database. Di tab ini user menginputkan username, password, dan confirm password. Tombol Entry digunakan untuk proses Entry data. Ketika tombol ini diklik, maka akan terjadi pengecekan terhadap data yang dimasukan. Apabila username telah dipakai atau password dan confirm password tidak sama, maka akan muncul pesan peringatan. Tombol cancel digunakan untuk membersihkan masukkan yang diinput user pada textbox.

Tab Edit digunakan untuk melakukan proses pergantian password. Tombol Ganti Password ini berfungsi untuk mengganti password. Ketika tombol ini diklik, akan melakukan validasi, bila username tidak

ada di database atau password baru dan confirm password tidak cocok, maka akan muncul pesan peringatan. Namun, bila validasi berhasil maka password yang lama akan berganti menjadi password yang baru. Tombol cancel digunakan untuk membersihkan masukkan yang diinput user pada textbox.

4.4 Antarmuka Kelola Laboratorium

Gambar 4.4 Rancangan Antarmuka Pengelolaan Laboratorium

Antarmuka Gambar 4.4 merupakan antar muka untuk melakukan pengelolaan data pemakaian laboratorium, yaitu pemakaian harian dan pemakaian khusus yang dilakukan sekali. Antarmuka ini memiliki tab Harian dan tab Khusus.

Tab Harian digunakan untuk memasukkan, mengubah, atau menghapus data jadwal penggunaan laboratorium Harian. Tombol entry digunakan untuk melakukan proses penambahan data ke database. Tombol Edit digunakan untuk melakukan proses perubahan data tertentu dari database. Tombol Delete digunakan untuk menghapus data tertentu dari database. Tombol Refresh digunakan untuk menghapus inputan user dari textbox yang tersedia. Pada awal tampilnya form ini, hanya tombol Entry dan Reset yang aktif. Tombol Edit dan Delete akan aktif ketika user memilih data dari list data harian.

Tab Khusus digunakan untuk memasukkan, mengubah, atau menghapus data jadwal penggunaan laboratorium Khusus atau hanya satu kali. Tombol entry digunakan untuk melakukan proses penambahan data ke database. Tombol Edit digunakan untuk melakukan proses perubahan data tertentu dari database. Tombol Delete digunakan untuk menghapus data tertentu dari database. Tombol Refresh digunakan untuk menghapus inputan user dari textbox yang tersedia. Pada awal tampilnya form ini, hanya tombol Entry dan Reset yang aktif. Tombol Edit dan Delete akan aktif ketika user memilih data dari list data khusus.

Untuk memasukkan dan mengubah data harus melihat data yang ada sehingga tidak terjadi bentrok antara jadwal satu dengan yang lain.

4.5 Antarmuka Daftar Aplikasi

Gambar 4.5 Rancangan Antarmuka Daftar Aplikasi

Antarmuka Gambar 4.6 merupakan antarmuka yang berfungsi untuk menambah, menghapus dan memperbaharui data aplikasi. Jenis aplikasi merupakan combo box yang berisi Browser, Messenger, Games, dan Others. Tombol Entry digunakan untuk menambah data aplikasi di

database. Tombol Edit digunakan untuk memperbaharui data aplikasi di database. Tombol Delete digunakan untuk menghapus data aplikasi dari database. Tombol Cancel digunakan untuk membersihkan inputan user dari textbox.

4.6 Antarmuka Broadcast

4.6.1 Antarmuka Broadcast non Extend

Gambar 4.6.1 Rancangan Antarmuka Broadcast non Extend

Antarmuka Gambar 4.5.1 merupakan antarmuka yang berfungsi untuk melakukan broadcast dari komputer server ke seluruh komputer client. pada form ini terdapat group box utama yaitu Group box Broadcast yang didalamnya terdapat group box lain.

1. Group Box IP Broadcast

Group ini berfungsi untuk menentukan dan menampilkan alamat IP Broadcast yang dipakai. Tombol Ganti

digunakan untuk mengganti alamat IP dan akan ditampilkan pada label IP Broadcast. Tombol Unicast digunakan untuk melakukan extend terhadap Form Broadcast ini dan menampilkan group box Unicast.

2. Group Box Jalankan Aplikasi

Group ini berfungsi untuk melakukan broadcast ke seluruh komputer client untuk menjalankan aplikasi tertentu. Tombol '...' merupakan tombol browse untuk menentukan letak aplikasi yang akan dijalankan. Tombol Start digunakan untuk mengirim command dan nama aplikasi yang akan dijalankan di seluruh komputer client.

3. Group Box Tutup/Block Aplikasi

Group ini berfungsi untuk melakukan broadcast ke seluruh komputer client untuk menutup atau blocking terhadap aplikasi tertentu. Check Box Browser, messenger, Games, dan Other digunakan untuk menentukan jenis aplikasi yang ingin di-block. Tombol Daftar Aplikasi digunakan untuk memanggil antarmuka lain, yang berfungsi untuk menambah dan memperbaharui data aplikasi. Tombol Block aplikasi digunakan untuk melakukan broadcast command dan info aplikasi yang akan di-block ke seluruh komputer client. Tombol Block Selain fungsinya sama dengan Tombol Block Aplikasi hanya saja yang di-block adalah aplikasi selain yang dipilih user. Tombol Block Komputer digunakan untuk melakukan block terhadap penggunaan komputer client. Tombol Shutdown digunakan untuk melakukan broadcast command ke seluruh komputer client agar komputer client melakukan shutdown. Tombol Block Copy/Paste

digunakan untuk broadcast command agar komputer client tidak dapat melakukan aksi Copy dan Paste. Tombol Block Firewall digunakan untuk broadcast command agar komputer client mematikan firewall.

4. Group Box Update/Kirim File

Group ini berfungsi untuk melakukan broadcast pengiriman file ke seluruh komputer client. Tombol '...' yang pertama merupakan tombol browse untuk menentukan file yang akan dikirim dan yang kedua untuk menentukan letak direktori tempat file akan diletakkan di komputer client. Tombol Send digunakan untuk melakukan broadcast pengiriman bagian-bagian file yang sudah ditentukan ke seluruh komputer client. Tombol Check File digunakan untuk melakukan pengecekan terhadap file yang dikirimkan ke komputer client, hasilnya akan ditampilkan ke dalam listbox. Tombol Cancel digunakan untuk membatalkan pengiriman file. Tombol Rip File digunakan untuk memecah file menjadi bagian-bagian kecil berukuran 60 KB. Tombol Join File digunakan untuk melakukan broadcast ke seluruh komputer *client* agar menggabungkan bagian-bagian file berukuran 60KB menjadi satu file yang utuh. Tombol Refresh digunakan untuk melakukan refresh pada listbox pengecekan file. Tombol Resend digunakan untuk mengirim bagian yang kurang ke komputer *client* sesuai dengan pilihan pada listbox.

5. Group Box Ubah Bahasa

Group ini berfungsi untuk melakukan broadcast ke seluruh komputer client untuk mengganti bahasa. Tombol 'Ganti' untuk mengirim command dan bahasa yang akan diinginkan komputer client.

Program Studi Teknik Informatika	DPPL – LabAssist	47/ 53
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.6.2 Antarmuka Broadcast Extend

The image shows a software interface titled "Broadcast Form" with a close button (x) in the top right corner. The interface is divided into two main sections: "Broadcast" on the left and "Unicast" on the right. Both sections have a similar layout:

- IP Address Section:** A text input field for "IP Broadcast" (or "IP Address" in the Unicast section) with a "Ganti" button to its right and an "Unicast" button below it.
- Application Execution Section:** A "Jalankan Aplikasi" section with a "Nama/Lokasi Aplikasi" text input field and a "Start" button below it.
- Block/Action Section:** A "Tutup/Block Aplikasi" section with checkboxes for "Browser", "Messenger", "Games", and "Other", and a "Daftar Aplikasi" button. Below these are buttons for "Block Aplikasi", "Block Komputer", "Shutdown", "Block Selain", "Block Copy/Paste", and "Block Firewall".
- Language Section:** A "Ubah Bahasa" section with a "Bahasa" text input field and a "Ganti" button.
- File Management Section:** An "Update/Kirim File" section with "Lokasi Asal File" and "Lokasi Tujuan File" text input fields. Below these are buttons for "Send", "Cek File", "Cancel", "Rip File", "Join File", and "Clean File".
- Footer Section:** "Resend" and "Refresh" buttons at the bottom.

Gambar 4.6.2 Rancangan Antarmuka Broadcast Extend

Antarmuka Gambar 4.5.2 ini merupakan extend dari Broadcast Form yang akan muncul ketika Tombol Unicast pada Broadcast Form diklik. Broadcast Form mendapat tambahan Group Box Unicast yang berfungsi untuk melakukan broadcast command pada satu komputer client sesuai dengan IP address yang diberikan user. Group box Unicast ini terdapat group box lain di dalamnya, yaitu:

1. Group IP address

Group ini berfungsi untuk menentukan dan menampilkan alamat IP client. Tombol Ganti digunakan untuk mengganti alamat IP klien.

2. Group Jalankan Aplikasi

Group ini berfungsi untuk melakukan broadcast ke komputer client untuk menjalankan aplikasi tertentu. Tombol '...' adalah tombol browse untuk menentukan letak aplikasi yang akan dijalankan. Tombol Start

digunakan untuk mengirim command dan info letak aplikasi yang akan dijalankan di komputer client.

3. Group Tutup/Block Aplikasi

Group ini berfungsi untuk melakukan broadcast ke komputer client agar mem-block aplikasi tertentu. Check Box Browser, messenger, Games, dan Other untuk menentukan jenis aplikasi yang ingin di-block. Tombol Block aplikasi digunakan untuk melakukan broadcast command dan info aplikasi yang akan di-block ke komputer client. Tombol Block Selain fungsinya sama dengan Tombol Block Aplikasi hanya saja yang di-block adalah aplikasi selain yang dipilih user. Tombol Block Komputer digunakan untuk melakukan block terhadap penggunaan komputer. Tombol Shutdown digunakan untuk mematikan komputer client. Tombol Block Copy/Paste digunakan untuk melakukan broadcast command agar tidak dapat melakukan aksi Copy dan Paste. Tombol Block Firewall digunakan untuk broadcast command agar komputer client mematikan firewall.

4.7 Antarmuka System Check

The image shows a screenshot of a software interface titled "Data System Form". The interface is divided into several sections. At the top, there is a "Computer Client" section containing a placeholder for a logo "(LOGO)", an "IP Broadcast" input field, and a "Change" button. Below this is a "Cek Data System" section with three radio buttons labeled "Hardware", "Software", and "Process", and a "Check Now" button. At the bottom of the form is a large text area labeled "(data)".

Gambar 4.7 Rancangan Antarmuka System Check

Antarmuka Gambar 4.7 merupakan antarmuka yang berfungsi untuk melihat data sistem pada seluruh komputer client. Form ini terdapat group box yang memiliki fungsi tertentu yaitu:

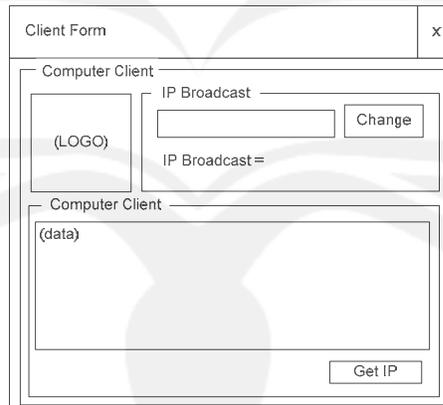
1. Group IP Broadcast

Group ini berfungsi untuk menentukan dan menampilkan IP Broadcast yang dipakai. Tombol Ganti digunakan untuk mengganti alamat IP Broadcast.

2. Group Cek Data Sistem

Group ini berfungsi untuk mengecek dan menampilkan data sistem seluruh komputer client. *Check box* Hardware, Software, dan Proses merupakan pilihan untuk data sistem yang akan ditampilkan. Tombol Check System digunakan untuk mendapatkan data sistem yang hasilnya akan ditampilkan ke listbox yang tersedia.

4.8 Antarmuka Client



Gambar 4.8 Rancangan Antarmuka Client

Antarmuka Gambar 4.8 merupakan antarmuka yang berfungsi untuk mendapatkan seluruh alamat IP komputer client. Form ini terdapat group box yang memiliki fungsi tertentu yaitu:

1. Group IP Broadcast

Group ini berfungsi untuk menentukan dan menampilkan IP Broadcast yang dipakai. Tombol Ganti digunakan untuk mengganti alamat IP Broadcast.

2. Group Computer Client

Group ini berfungsi untuk meminta dan menampilkan alamat IP seluruh komputer client. Tombol Get IP digunakan untuk meminta alamat IP kemudian hasilnya akan ditampilkan ke listbox yang tersedia.

4.9 Antarmuka Instal Program

The screenshot shows a software installation window titled "Instal Program Form". It is divided into several functional areas:

- Instal:** Contains a logo field labeled "(LOGO)", an "IP Broadcast:" label with a text input field and a "Ganti" button, and a label "IP Broadcast:=".
- Update/Kirim File:** Includes "Lokasi Asal File" and "Lokasi Tujuan File" fields with browse buttons (...), and buttons for "Send", "Rip File", "Join File", and "Clean File".
- Unrar:** Features "Lokasi Asal" and "Lokasi Tujuan" fields with browse buttons (...), and an "Unrar" button.
- Setup:** Contains "NamaLokasi Aplikasi" field with a browse button (...), a "Start" button, "Nama Proses" and "Key" fields with "Cek" and "Send" buttons respectively.

At the bottom of the main content area, there are "Resenc" and "Refresh" buttons.

Gambar 4.8 Rancangan Antarmuka Instal Program

Antarmuka Gambar 4.8 merupakan antarmuka yang berfungsi untuk mendapatkan seluruh alamat IP komputer client. Form ini terdapat group box yang memiliki fungsi tertentu yaitu:

1. Group IP Broadcast

Group ini berfungsi untuk menentukan dan menampilkan IP Broadcast yang dipakai. Tombol Ganti digunakan untuk mengganti alamat IP Broadcast.

2. Group Box Update/Kirim File

Group ini berfungsi untuk melakukan broadcast pengiriman file ke seluruh komputer client. Tombol '...' yang pertama merupakan tombol browse untuk menentukan file yang akan dikirim dan yang kedua untuk menentukan letak direktori tempat file akan diletakkan di komputer client. Tombol Send digunakan untuk melakukan broadcast pengiriman bagian-bagian file yang sudah ditentukan ke seluruh komputer client. Tombol Check File digunakan untuk melakukan pengecekan terhadap file yang dikirimkan ke komputer client, hasilnya akan ditampilkan ke dalam listbox. Tombol Cancel digunakan untuk membatalkan pengiriman file. Tombol Rip File digunakan untuk memecah file menjadi bagian-bagian kecil berukuran 60 KB. Tombol Join File digunakan untuk melakukan broadcast ke seluruh komputer *client* agar menggabungkan bagian-bagian file berukuran 60KB menjadi satu file yang utuh. Tombol Refresh digunakan untuk melakukan refresh pada listbox pengecekan file. Tombol Resend digunakan untuk mengirim bagian yang kurang ke komputer *client* sesuai dengan pilihan pada listbox.

3. Group Unrar

Group ini berfungsi untuk melakukan broadcast ke komputer client untuk melakukan unrar terhadap file tertentu. Tombol '...' untuk menentukan letak lokasi file. Tombol Unrar digunakan untuk mengirim command dan info letak file yang akan di-unrar.

4. Group Jalankan Aplikasi

Group ini berfungsi untuk melakukan broadcast ke komputer client untuk menjalankan aplikasi tertentu. Tombol '...' adalah tombol browse untuk menentukan

letak aplikasi yang akan dijalankan. Tombol Start digunakan untuk mengirim command dan info letak aplikasi yang akan dijalankan di komputer client.

5. Group Setup

Group ini berfungsi untuk melakukan broadcast ke komputer client untuk melakukan setup program. Tombol Cek digunakan untuk mencari kesiapan proses (yang dimasukkan oleh user ke dalam textbox nama proses) di komputer klien. Tombol Send digunakan untuk mengirimkan key ke komputer klien pada proses tertentu. Tombol Resend digunakan untuk mengirim ulang perintah yang hilang.