

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan pembahasan pada bab-bab sebelumnya, dapat ditarik kesimpulan :

1. Perangkat lunak IPFilter berhasil dibangun dan dapat digunakan untuk melakukan blok terhadap IP serta *port* yang dikehendaki yang sedianya akan dilakukan akses dari suatu koneksi yang dilakukan didalam suatu jaringan komputer lokal dalam *subnet* yang sama dan kemudian membuat *summary*-nya.
2. Aplikasi IPFilter dapat dijalankan di sisi server sebuah jaringan komputer, sebagai filter arus koneksi yang dilakukan terhadap server melalui *ip address* maupun *port address*.

#### **5.2. Saran**

Beberapa saran yang dapat ditarik dari proses analisa sampai pembuatan laporan tugas akhir ini adalah sebagai berikut :

1. Perangkat lunak IPFilter dapat dikembangkan tidak hanya untuk melakukan Blocked IP di jaringan komputer lokal saja tetapi juga dapat untuk melakukan blok pada jaringan komputer di luar jaringan tersebut.
2. Jika memang harus berjalan pada sebuah jaringan komputer lokal sebaiknya dapat ditambahkan pula fungsi untuk pengenalan

subnet terhadap koneksi yang terjadi, jika mempunyai subnet yang sama berarti proses blok dapat dilakukan karena masih dalam satu jaringan, tetapi jika tidak / mempunyai subnet yang berbeda / berbeda jaringan maka proses blok-IP tidak dapat dilakukan.



## DAFTAR PUSTAKA

### Tutorial :

- <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds.msp?mfr=true>
- <http://java.sun.com/docs/books/tutorial/uiswing/components/index.html>
- NetBeans, Java and JfreeChart.pdf from [Carl.David@uconn.edu](mailto:Carl.David@uconn.edu)

### Jfree :

- <http://sourceforge.net/projects/jfreechart>
- <http://www.jfree.org/jfreechart/download.html>
- [http://sourceforge.net/project/showfiles.php?group\\_id=15494&package\\_id=12428&release\\_id=516325](http://sourceforge.net/project/showfiles.php?group_id=15494&package_id=12428&release_id=516325)
- <http://www.jfree.org/jfreechart/>
- [http://www.sideofsoftware.com/reports\\_ChartExample.htm](http://www.sideofsoftware.com/reports_ChartExample.htm)
- <http://forum.java.sun.com/thread.jspa?threadID=638913&messageID=4021213>
- [http://genweb.ostg.com/google/radlinks/sf\\_bottom\\_right\\_results.html?google\\_kw=X+Bar+and+R+Chart&google\\_page\\_url=http://sourceforge.net/project/showfiles.php?group\\_id=15494&package\\_id=12428&release\\_id=516325](http://genweb.ostg.com/google/radlinks/sf_bottom_right_results.html?google_kw=X+Bar+and+R+Chart&google_page_url=http://sourceforge.net/project/showfiles.php?group_id=15494&package_id=12428&release_id=516325)

### Teori :

- *Pemrograman GUI Swing Java dengan Netbeans.5*, G Sri Hartati, B Herry Suharto, M Soesilo Wiyono, Penerbit ANDI, Edisi 1, 2006.
- <http://www.netbeans.org>
- <http://java.sun.com>
- *Keamanan Jaringan Internet*, Onno W Purbo, Tony Wiharjito, Penerbit PT. Elex Media Komputindo, 2000.
- *TCP/IP-Standar, Desain dan Implementasi*, Onno W Purbo, Penerbit PT. Elex Media Komputindo, 2002.
- <http://www.teknisoft.net/download/artikel/network/penghitungan-subneting.dpr>
- <http://yoyok.wordpress.com/2007/09/14/penghitungan-subneting>
- <http://d3tkjdepok.wordpress.com/2007/09/12/belajar-subneting>
- <http://antoklaten.tripod.com/id5.html>
- [surya-kubiwa.tripod.com/rangkuman/Subneting.pdf](http://surya-kubiwa.tripod.com/rangkuman/Subneting.pdf)
- CCNA1 v3.0 Module 10 "Routing Fundamentals and Subnets."

**SKPL**

**SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK**

**Internet Protocol Blocked  
( IPBlocked )**

untuk :


**Pengguna Jaringan Komputer**

Dipersiapkan oleh :

**Yohanes David Widyatama**

**01 07 03093**

**Program Studi Teknik Informatika – Fakultas Teknologi Industri  
Universitas Atma Jaya Jogjakarta**

	<b>Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Atma Jaya Jogjakarta</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<b>GL01-G08</b>		<b>1 / 26</b>
	<b>Revisi</b>			<b>Tgl : 20-08-2008</b>

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis Oleh	David							
Diperiksa Oleh	EDJ							
Disetujui Oleh	EDJ							

## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

# Daftar Isi

1	Pendahuluan .....	7
1.1	Tujuan .....	7
1.2	Lingkup Masalah .....	7
1.3	Definisi, Akronim dan Singkatan.....	8
1.4	Referensi .....	9
1.5	Deskripsi Umum (Overview).....	9
2	Deskripsi Kebutuhan.....	10
2.1	Perspektif Produk.....	10
2.2	Kebutuhan Fungsionalitas Perangkat Lunak.....	12
2.2.1	Use Case : Login.....	12
2.2.2	Use Case : Ubah Password .....	12
2.2.3	Use Case : Setting Blocked.....	12
2.2.4	Use Case : Report Statistic .....	13
2.2.5	Use Case : Report Grafik .....	13
2.2.6	Use Case : Tes Koneksi .....	13
2.3	Karakteristik Aktor .....	13
3	Spesifikasi Rinci Kebutuhan .....	13
3.1	Kebutuhan Fungsionalitas.....	13
3.1.1	Spesifikasi Use Case : Login .....	13
3.1.2	Spesifikasi Use Case : Ubah Password.....	14
3.1.3	Spesifikasi Use Case : Setting Blocked .....	15
3.1.4	Spesifikasi Use Case : Report Statistic .....	16
3.1.5	Spesifikasi Use Case : Report Grafik .....	17
3.1.5	Spesifikasi Use Case : Tes Koneksi.....	18
3.2	Kebutuhan Non-fungsionalitas .....	18
3.2.1	Kebutuhan Antarmuka Eksternal.....	19
3.2.2	Antarmuka Pemakai.....	19
3.2.3	Antarmuka Perangkat keras .....	19
3.2.4	Antarmuka Perangkat Lunak .....	19
3.2.5	Persistent Data .....	20
3.2.6	ERD .....	21
4	Realisasi Use Case .....	21
4.1	Statistic Structure Diagram .....	21
4.1.1	Analysis Class Diagram : Package Dependencies .....	21
4.1.2	Analysis Class Diagram : Use Case Login .....	22
4.1.3	Analysis Class Diagram : Use Case Ubah Password.....	22
4.1.4	Analysis Class Diagram : Use Case Setting Blocked .....	22
4.1.5	Analysis Class Diagram : Use Case Report Statistic .....	23
4.1.6	Analysis Class Diagram : Use Case Report Grafik .....	23
4.1.7	Analysis Class Diagram : Use Case Tes Koneksi.....	23

4.2	Interaction Diagram .....	24
4.2.1	Analysis Collaboration Diagram : Login.....	24
4.2.2	Analysis Collaboration Diagram : UbahPassword .....	24
4.2.3	Analysis Collaboration Diagram : SettingBlocked.....	25
4.2.4	Analysis Collaboration Diagram : Report Statistic.....	25
4.2.5	Analysis Collaboration Diagram : Report Grafik .....	25
4.2.6	Analysis Collaboration Diagram : Tes Koneksi .....	26





## Daftar Gambar

Gambar 1.1	Skema Aplikasi IPBlocked .....	7
Gambar 1.2	Skema Aplikasi IPBlocked.....	11
Gambar 2.1	Use Case Diagram IPBlocked .....	12
Gambar 4.1	Entity Relationship Diagram .....	21
Gambar 4.2	Analysis Class Diagram : Package Dependencies IPBlocked .....	21
Gambar 4.3	Analysis Class Diagram : UseCase Login.....	22
Gambar 4.4	Analysis Class Diagram : UseCase UbahPassword .....	22
Gambar 4.5	Analysis Class Diagram : UseCase SettingBlocked.....	22
Gambar 4.6	Analysis Class Diagram : UseCase Report Statistic.....	23
Gambar 4.7	Analysis Class Diagram : UseCase Report Grafik .....	23
Gambar 4.8	Analysis Class Diagram : UseCase Tes Koneksi .....	23
Gambar 4.2.1	Analysis Collaboration Diagram : Login .....	24
Gambar 4.2.2	Analysis Collaboration Diagram : UbahPassword .....	24
Gambar 4.2.3	Analysis Collaboration Diagram : SettingBlocked .....	25
Gambar 4.2.4	Analysis Collaboration Diagram : Report Statistic .....	25
Gambar 4.2.5	Analysis Collaboration Diagram : Report Grafik.....	25
Gambar 4.2.6	Analysis Collaboration Diagram : Tes Koneksi .....	26

## Daftar Tabel

Tabel 3.1	Spesifikasi UseCase : Login.....	13
Tabel 4.2	Spesifikasi UseCase : Ubah Password .....	14
Tabel 4.3	Spesifikasi UseCase : Setting Blocked.....	15
Tabel 4.4	Spesifikasi UseCase : Report Statistic .....	16
Tabel 4.5	Spesifikasi UseCase : Report Grafik .....	17
Tabel 4.6	Spesifikasi UseCase : Tes Koneksi .....	18

## Daftar Lampiran

-

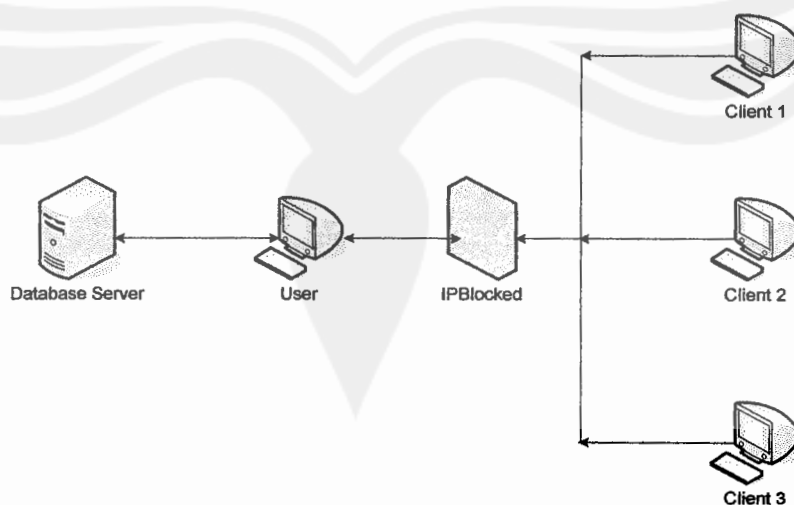
## 1. Pendahuluan

### 1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau *Software Requirement Specification* (SRS) ini dibuat sebagai dokumen spesifikasi kebutuhan perangkat lunak **IPBlocked** ( Internet Protocol Blocked ) untuk mendefinisikan kebutuhan perangkat lunak, yang meliputi antarmuka eksternal, dan juga atribut, serta mendefinisikan fungsi perangkat lunak, juga mendefinisikan batasan perangkat lunak untuk mempermudah melakukan berbagai evaluasi dan pengembangan sistem dikemudian harinya.

### 1.2 Lingkup Masalah

*Firewall* adalah sebuah sistem yang didesain untuk mencegah akses dari pihak yang tidak berhak (*unauthorized*) menuju atau dari jaringan lokal. Firewall dapat diimplementasikan, dalam bentuk hardware, software, atau kombinasi keduanya. Firewall biasanya digunakan untuk mencegah/mengendalikan aliran data tertentu. Semua data yang masuk ataupun yang keluar jaringan harus melewati firewall, yang akan memeriksa setiap paket apakah cocok atau tidak dengan kriteria/aturan yang ada pada standar keamanan yang didefinisikan dalam firewall.



Gambar 1.1 Skema Aplikasi IPBlocked  
(Sumber : David, 2007)

Aplikasi IPBlocked ini juga dapat dikatakan sebagai suatu firewall, karena aplikasi ini digunakan *user* untuk membatasi serta melakukan pengamanan terhadap hal-hal yang tidak diinginkan, mem-**blok IP** dan juga **port** yang dikehendaki agar tidak dapat di akses oleh *client* dan mengetahui traffic yang sedang berlangsung pada konektivitas yang sedang berlangsung.

### 1.3 Definisi, Akronim dan Singkatan

- **SKPL** adalah Spesifikasi Kebutuhan Perangkat Lunak, yang juga disebut sebagai Software Requirement Specification (SRS), yang merupakan spesifikasi dari perangkat lunak yang digunakan.
- **SKPL- IPSec -XX** adalah Kode Internet Protocol security berbasis Java Technology yang sedang dikembangkan, yang merepresentasikan kebutuhan pada *IPSec* .
- **IPBlocked**, merupakan aplikasi yang dikembangkan untuk jaringan komputer.
- **Firewall**, adalah sebuah sistem yang didesain untuk mencegah akses dari pihak yang tidak berhak (*unauthorized*) menuju atau dari jaringan lokal
- **IP**, *Internet Protocol* merupakan alamat tiap-tiap komputer yang berada pada suatu jaringan komputer yang menggunakan protokol jaringan TCP/IP..
- **Netbeans** , merupakan tools yang digunakan untuk membuat antar-muka dari aplikasi IPBlocked.
- **GUI**, *Graphical User Interface*, merupakan antarmuka pemakai dengan sistem.
- **User**, merupakan pengguna aplikasi IPBlocked untuk melakukan filtering terhadap komunikasi data yang sedang berlangsung dengannya, dalam hal ini lebih tepat dinamakan Admin (Administrator).

## 1.4 Referensi

1. GLO1, *Spesifikasi Kebutuhan Perangkat Lunak*, Jurusan Teknik Informatika – UAJY
2. David, Yohanes. *Proposal Aplikasi Firewall (Internet protocol Security)*, Universitas Atma Jaya Yogyakarta, 2006.
3. Dharwiyanti, Sri, *Pengantar Unified Modeling Language (UML)*, Kuliah Umum Ilmu Komputer, 2003.

## 1.5 Deskripsi Umum ( Overview )

Secara umum Dokumen SKPL ini dibagi menjadi tiga bagian utama. Bagian pertama berisi tentang penjelasan dokumen SKPL yang mencakup tujuan pembuatan dokumen SKPL ini beserta lingkup masalah yang melatar-belakangi pembuatan perangkat lunak ini guna pengembangan perangkat lunak tersebut, definisi, referensi, dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan secara umum mengenai perangkat lunak yang akan dikembangkan, meliputi perspektif produk, fungsi produk, karakteristik pengguna, batasan-batasan dalam penggunaan perangkat lunak serta asumsi serta ketergantungan yang dipakai dalam pengembangan perangkat.

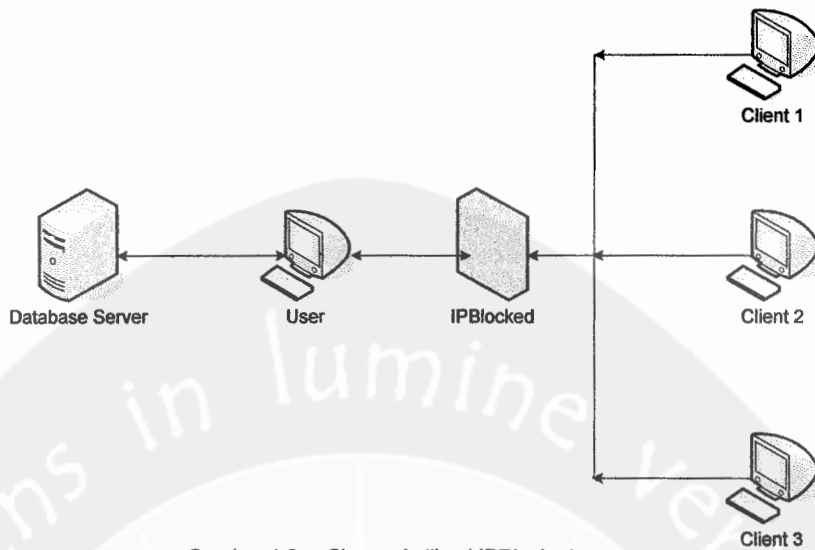
Bagian ketiga berisi tentang kebutuhan perangkat lunak secara lebih khusus, diantaranya kebutuhan antarmuka eksternal, kebutuhan performansinya, batasan perancangannya, atribut sistem perangkat lunak serta kebutuhan-kebutuhan lainnya.

## 2. DESKRIPSI KEBUTUHAN

### 2.1 Perspektif Produk.

Aplikasi *IPBlocked* merupakan aplikasi yang dikembangkan untuk membantu para user jaringan komputer terutama admin (administrator) untuk lebih mengamankan PC/jaringan tempat user tersebut melakukan pekerjaannya. *IPBlocked* nantinya akan berfungsi tidak berbeda dengan firewall, yakni sebagai blocked, yang akan membatasi pengguna jaringan komputer untuk melakukan koneksi dengan admin melalui pembatasan pada *IP address* (internet protocol) dan juga *port address*, *IPBlocked* juga akan mencatat berapa kali IP tersebut melakukan koneksi dengan admin dan menampilkannya dalam sebuah grafik.

Aplikasi ini nantinya akan dijalankan pada sebuah jaringan komputer lokal. Semua proses koneksi yang dilakukan terhadap admin akan dicatat dan diawasi oleh aplikasi *IPBlocked* ini, dimana rule (aturan) yang digunakan pada *IPBlocked* tersebut tergantung dari user yang menjalankannya (administrator). Aplikasi ini hanya berjalan satu arah, yakni hanya ditempatkan pada sebuah PC yang bertindak sebagai *server* yang terhubung pada jaringan lokal (LAN).

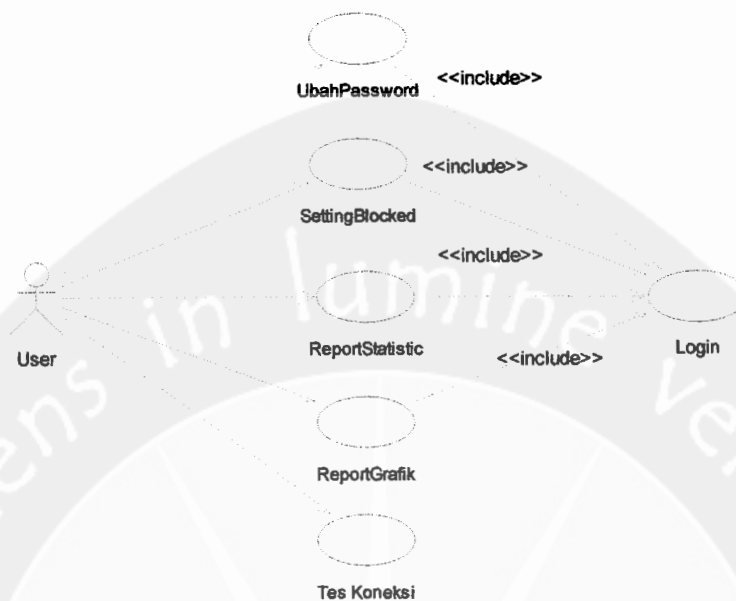


Gambar 1.2 Skema Aplikasi IPBlocked  
(Sumber : David, 2007)

Pada saat program dijalankan, program akan menjalankan perintah untuk mendeteksi komputer-komputer mana aja yang mengakses komputer *server*, jadi, ini akan dicatat sebagai waktu awal, tapi pada saat dideteksi lagi sudah tidak ada, maka koneksi sebelumnya dianggap selesai. Untuk fungsi blok ip dan juga port, ini tergantung pada permintaan user (hal ini dapat disesuaikan pada saat melakukan setting blocked), begitu disimpan sebagai *rule* yang diinginkan , maka fungsi ini pun akan berjalan sebagaimana mestinya. Untuk mengetahui apakah proses blok yang dilakukan berhasil atau tidak, maka dari sisi *client* digunakan sebuah *network tester*.

Aplikasi IPBlocked ( Internet Protocol Blocked ) ini akan dibuat menggunakan bahasa pemrograman Java NetBeans 5.5 untuk rancangan antarmukanya. IPBlocked akan dijalankan pada sebuah jaringan komputer yang menggunakan sistem operasi Windows XP SP 2.

## 2.2 Kebutuhan Fungsionalitas Perangkat Lunak



Gambar 2.1 Use Case Diagram IPBlocked

### 2.2.1 Use Case Login

Deskripsi : Use Case ini digunakan oleh *user* untuk memperoleh akses ke sistem. Login didasarkan pada sebuah *user\_name* dan *password* yang berupa rangkaian karakter yang telah tersimpan pada basis data.

Lihat : Spesifikasi Use Case : Login(UC- IPBlocked -01).

### 2.2.2 Use Case UbahPassword

Deskripsi : Use Case ini digunakan oleh *user* untuk mengganti *password* dari user pengguna aplikasi IPBlocked.

Lihat : Spesifikasi Use Case : UbahPassword(UC-IPBlocked-02).

### 2.2.3 Use Case SettingBlocked

Deskripsi : Use Case ini digunakan oleh *user* untuk mengubah ataupun menambah aturan / rule yang digunakan sebagai *filtering* dari aplikasi IPBlocked ini.

Lihat : Spesifikasi Use Case : SettingBlocked (UC-IPBlocked-03)

## 2.2.4 Use Case ReportStatistic

Deskripsi : Use Case ini digunakan oleh *user* untuk melihat keseluruhan fungsi dari aplikasi IPsec tersebut port-port mana saja yang melakukan koneksi, paket data yang dikirim serta menampilkan grafiknya.

Lihat : Spesifikasi Use Case : ReportStatistic(UC- IPBlocked -04).

## 2.2.5 Use Case ReportGrafik

Deskripsi : Use Case ini digunakan oleh *user* untuk melihat grafik yang dihasilkan dari data statistik yang telah tersimpan pada sistem.

Lihat : Spesifikasi Use Case : Report Statistik (UC- IPBlocked -05).

## 2.2.6 Use Case TesKoneksi

Deskripsi : Use case ini digunakan *client* untuk melakukan test apakah proses blok terhadap *ip address* maupun *port address client* yang dilakukan oleh server berhasil atau tidak.

Lihat : Spesifikasi Use Case : Report TesKoneksi (UC- IPBlocked -06).

## 2.3 Karakteristik Aktor

### 2.3.1 User

User harus memenuhi hal-hal berikut :

1. Memahami pengoperasian komputer secara aktif.
2. Mengerti tentang penggunaan aplikasi pada jaringan.
3. Mengerti jaringan komputer.

## 3 Spesifikasi Rinci Kebutuhan

### 3.1 Spesifikasi Kebutuhan Fungsionalitas

#### 3.1.1 Spesifikasi Use Case : Login

Tabel 3.1 Spesifikasi Use Case : Login

Use Case ID	UC-IPBlocked-01
Use Case Name	Login



<b>Use Case Type</b>	Essential
<b>Priority</b>	High
<b>Actors</b>	<i>User</i>
<b>Description</b>	Use Case ini digunakan oleh <i>aktor</i> untuk memperoleh akses ke sistem. Login didasarkan pada sebuah <i>User_ID</i> dan <i>Password</i> yang unik milik user.
<b>Preconditions</b>	-
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan antar muka untuk Login.</li> <li>2. Aktor memasukkan <i>username</i> dan <i>password</i>.</li> <li>3. Sistem memeriksa <i>username</i> dan <i>password</i>.</li> <li>4. Sistem memberikan akses kepada <i>user</i>.</li> </ol>
<b>Alternative Path</b>	-
<b>Postconditions</b>	Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1. <i>User_ID</i> dan <i>Password</i> yang dimasukkan tidak sesuai (Tabel 3.1 Basic Path 2) : <ol style="list-style-type: none"> <li>a. Sistem menampilkan peringatan bahwa <i>User_ID</i> dan <i>Password</i> tidak sesuai.</li> <li>b. Kembali ke 3.1 Basic Path 2.</li> </ol> </li> </ol>
<b>Extends</b>	-
<b>Includes</b>	-
<b>Bussines Rules</b>	-

### 3.1.2 Spesifikasi Use Case : UbahPassword

Tabel 3.2 Spesifikasi Use Case : UbahPassword

<b>Use Case ID</b>	UC-IPBlocked-02
<b>Use Case Name</b>	UbahPassword
<b>Use Case Type</b>	Essential
<b>Priority</b>	High
<b>Actors</b>	<i>User</i>
<b>Description</b>	Use Case ini digunakan <i>user</i> untuk melakukan pengelolaan data user, yang berupa penggantian password.

<b>Preconditions</b>	<b>Use Case : Login</b> sudah dilaksanakan dan user sudah berhasil memasuki sistem.
<b>Basic Path</b>	<b>Ganti password :</b> 1. Sistem menampilkan antar muka untuk ganti password. 2. Aktor memasukkan password yang lama. 3. Aktor memasukkan password yang baru. 4. Aktor memasukkan verifikasi password baru. 5. Sistem mengecek apakah data tersebut sudah digunakan atau belum. 6. Sistem menyimpan password user tersebut ke data base sebagai password baru untuk melakukan login.
<b>Alternative Path</b>	-
<b>Postconditions</b>	Data password pada basis data berubah.
<b>Exception Paths</b>	1. Password yang dimasukkan sudah ada (setelah tabel 3.2 Basic Path 5) <ul style="list-style-type: none"> <li>a. Sistem menampilkan peringatan bahwa password sama.</li> <li>b. Kembali ke Tabel 3.2 Basic Path 2.</li> </ul>
<b>Extends</b>	-
<b>Includes</b>	<b>Use Case : Login</b>
<b>Bussines Rules</b>	-

### 3.1.3 Spesifikasi Use Case : SettingBlocked

Tabel 3.3 Spesifikasi Use Case : SettingBlocked

<b>Use Case ID</b>	UC-IPBlocked-03
<b>Use Case Name</b>	SettingBlocked
<b>Use Case Type</b>	Essential
<b>Priority</b>	High
<b>Actors</b>	<i>User</i>
<b>Description</b>	Use Case ini digunakan <i>user</i> untuk melakukan setting terhadap aturan-aturan yang digunakan untuk melakukan blocked ip mana saja yang nantinya akan melakukan koneksi.

<b>Preconditions</b>	<b>Use Case : Login</b> sudah dilaksanakan dan user dapat melakukan setting block sesuai dengan yang diinginkan.
<b>Basic Path</b>	<b>Setting block :</b> <ol style="list-style-type: none"> <li>1. Sistem menampilkan antar muka untuk Setting Blocked</li> <li>2. Aktor memasukkan Alamat host / IP.</li> <li>3. Aktor memasukkan Port.</li> <li>4. Aktor memilih tombol simpan.</li> <li>5. Sistem mengecek apakah data tersebut valid atau tidak.</li> <li>6. Jika valid, maka sistem akan menyimpannya sebagai aturan yang harus dijalankan.</li> </ol>
<b>Alternative Path</b>	-
<b>Postconditions</b>	Setting block yang baru sudah tersimpan.
<b>Exception Paths</b>	
<b>Extends</b>	-
<b>Includes</b>	<b>Use Case : Login</b>
<b>Bussines Rules</b>	-

### 3.1.4 Spesifikasi Use Case : ReportStatistic

**Tabel 3.4 Spesifikasi Use Case : ReportStatistic**

<b>Use Case ID</b>	UC-IPBlocked-04
<b>Use Case Name</b>	ReportStatistic
<b>Use Case Type</b>	Essential
<b>Priority</b>	High
<b>Actors</b>	<i>User/Aktor</i>
<b>Description</b>	Use Case ini digunakan <i>User/Aktor</i> untuk melihat keseluruhan statistic yang didapat oleh sistem, mengenai ip mana saja yang melakukan koneksi dengan kita, port yang digunakan saat melakukan koneksi, waktu awal koneksi dan waktu akhir koneksi..
<b>Preconditions</b>	<b>Use Case : Login</b> sudah dilaksanakan dan user dapat melihat data statistik yang diinginkan.
<b>Basic Path</b>	1. User memilih melihat Statistic dengan menekan tombol

	Statistic pada menu yang disediakan sistem. 2. Sistem menampilkan Tabel Statistik dari blocki ip yang dilakukan.
<b>Alternative Path</b>	-
<b>Postconditions</b>	User mengetahui ststistik yang terjadi terhadap jaringan komputer yang melakukan koneksi dengan user.
<b>Exception Paths</b>	-
<b>Extends</b>	-
<b>Includes</b>	-
<b>Bussines Rules</b>	-

### 3.1.5 Spesifikasi Use Case : ReportGrafik

Tabel 3.5 Spesifikasi Use Case : ReportGrafik

<b>Use Case ID</b>	UC-IPBlocked-05
<b>Use Case Name</b>	ReportGrafik
<b>Use Case Type</b>	Essential
<b>Priority</b>	High
<b>Actors</b>	<i>User/Aktor</i>
<b>Description</b>	Use Case ini digunakan <i>User/Aktor</i> untuk melihat grafik dari keseluruhan aktifitas yang terjadi dengan sistem.
<b>Preconditions</b>	Use Case : <b>Login</b> sudah dilaksanakan dan user dapat melihat data statistik yang diinginkan.
<b>Basic Path</b>	1. User memilih menu untuk menampilkan Grafik dengan menekan tombol Grafik. 2. Sistem menampilkan Grafik yang dihasilkan berdasarkan data statistik yang ada.
<b>Alternative Path</b>	-
<b>Postconditions</b>	<i>User</i> mengetahui grafik yang terjadi dengan sistem.
<b>Exception Paths</b>	-
<b>Extends</b>	-
<b>Includes</b>	-
<b>Bussines Rules</b>	-

### 3.1.6 Spesifikasi Use Case : TesKoneksi

Tabel 3.6 Spesifikasi Use Case : TesKoneksi

<b>Use Case ID</b>	UC-IPBlocked-06
<b>Use Case Name</b>	TesKoneksi
<b>Use Case Type</b>	Essential
<b>Priority</b>	High
<b>Actors</b>	<i>User</i>
<b>Description</b>	Use case ini digunakan <i>client</i> untuk meng-test apakah proses blok terhadap ip address maupun port client yang dilakukan oleh server berhasil atau tidak
<b>Preconditions</b>	Blok telah dilakukan pada sisi server.
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. Sistem menampilkan antar muka untuk tes koneksi.</li><li>2. <i>Client</i> memasukkan alamat host server .</li><li>3. <i>Client</i> memasukkan alamat port yang akan dicek.</li><li>4. <i>Client</i> menjalankan fungsi tes koneksi dengan menekan tombol "Connect".</li><li>5. Sistem menampilkan status koneksi dari <i>client</i>, apakah Connected atau Failed.</li></ol>
<b>Alternative Path</b>	-
<b>Postconditions</b>	Blok telah dilakukan.
<b>Exception Paths</b>	-
<b>Extends</b>	-
<b>Includes</b>	-
<b>Bussines Rules</b>	-

### 3.2 Spesifikasi Kebutuhan Non-Fungsionalitas

Pengguna yang menggunakan perangkat lunak IPBlocked harus menggunakan komputer yang terkoneksi dengan komputer yang lain dalam suatu jaringan lokal. Untuk pengujiannya, digunakan sistem *client-server*, dimana Aplikasi IPBlocked dijalankan dari sisi *server* dan Network Tester dijalankan pada sisi *Client*.

### 3.2.1 Kebutuhan Antarmuka Eksternal

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

### 3.2.2 Antarmuka Pemakai

Pemakai berinteraksi langsung dengan sistem IPBlocked melalui antarmuka berbasis GUI pada layar komputer. Piranti masukan yang digunakan untuk memasukkan data login yang berupa user name dan password dengan menggunakan *keyboard* dari perangkat yang digunakan. Sedangkan keluaran dari sistem berupa data yang akan dibaca oleh *user* akan langsung di tampilkan ke layar monitor.

### 3.2.3 Antarmuka Perangkat Keras

Antarmuka perangkat keras desain sistem yang digunakan oleh aplikasi IPBlocked adalah :

1. PC / Personal Computer.
2. Keyboard.
3. Monitor.
4. Mouse.
5. LAN.

### 3.2.4 Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mendukung berjalannya perangkat lunak IPBlocked adalah:

1. Nama : *Windows XP SP 2*  
Sumber : *Microsoft*  
Fungsi : Sistem Operasi PC

2. Nama : *JDK.6 for Windows-i586*  
Sumber : *Sun Microsystem*  
Fungsi : Tool perancangan antarmuka aplikasi IPBlocked.
3. Nama : *Netbeans 5-5 for Windows*  
Sumber : *Netbeans*  
Fungsi : Tool perancangan antarmuka aplikasi IPBlocked.
4. Nama : *SQLyog-502*  
Sumber : *mySQL*  
Fungsi : Database sistem.
5. Nama : *mySQL Connector-Java-3.0.11*  
Sumber : *mySQL*  
Fungsi : Konektor sistem dengan Database.

### **3.2.5 Persistent Data**

#### **3.2.5.1 File Blocked**

File ini menyimpan data setingan alamat ip mana saja yang di blok, yang meliputi *id-blocked*, *ip-address* dan *port*.

#### **3.2.5.2 File Blocked\_Log**

File ini menyimpan data history dari alamat IP yang di-blok mulai dari awal saat aplikasi dijalankan ,yang meliputi *id\_blocked* , *ip\_address*, *port* dan *jenis..*

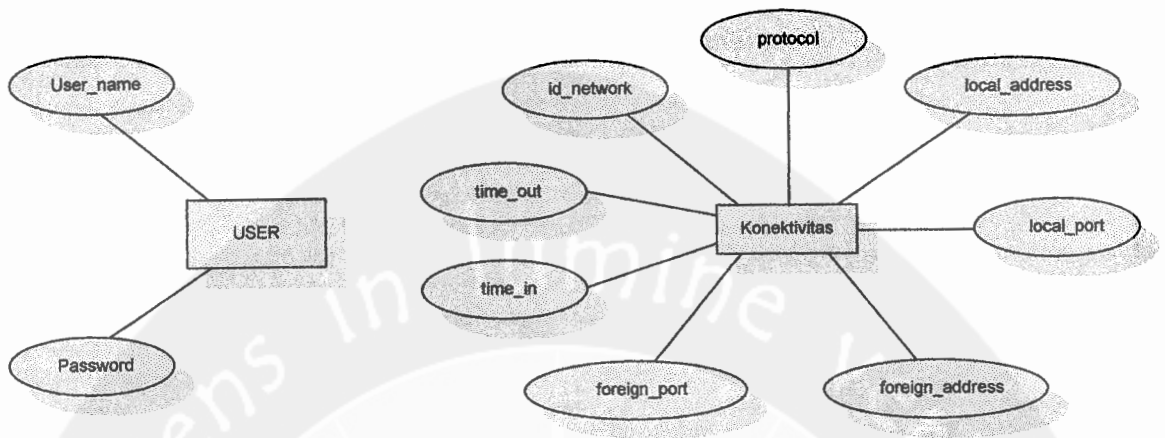
#### **3.2.5.3 File Network**

File ini menyimpan data konektivitas/history komputer yang mengakses komputer server, mulai dari protocol, local port, local address, foreign port, foreign address, time in dan time out.

#### **3.2.5.4 File Password**

File ini menyimpan data username dan password yang digunakan user untuk masuk ke dalam sistem melalui proses login.

### 3.2.6 Entity Relationship Diagram (ERD)

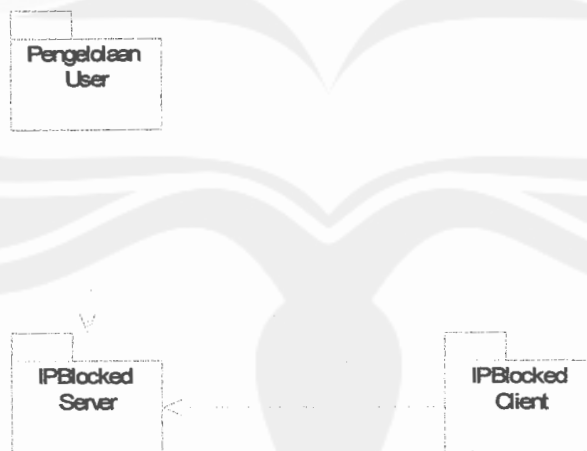


Gambar 4.1. Entity Relationship Diagram

## 4 Realisasi Use Case

### 4.1 Static Structure Diagram

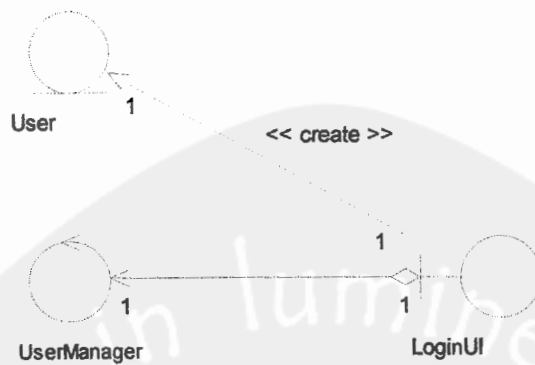
#### 4.1.1 Analysis Class Diagram : Package Dependencies



Gambar 4.2. Analysis Class Diagram : Package Dependencies IPBlocked

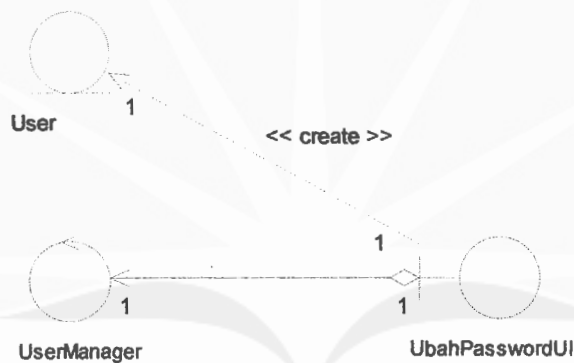


#### 4.1.2 Analysis Class Diagram : Use Case Login



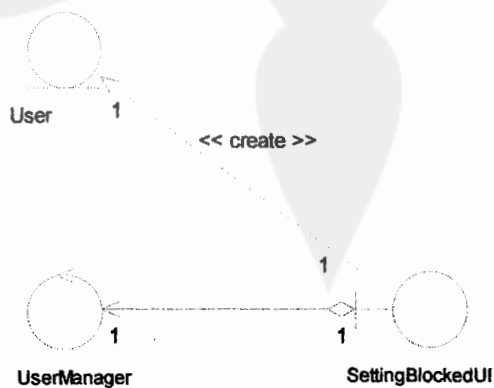
Gambar 4.2. Analysis Class Diagram : Use Case Login

#### 4.1.3 Analysis Class Diagram : Use Case UbahPassword



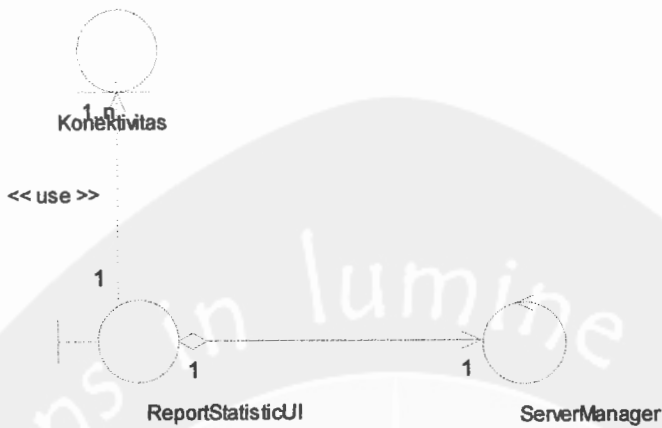
Gambar 4.3. Analysis Class Diagram : Use Case UbahPassword

#### 4.1.4 Analysis Class Diagram : Use Case SettingBlocked



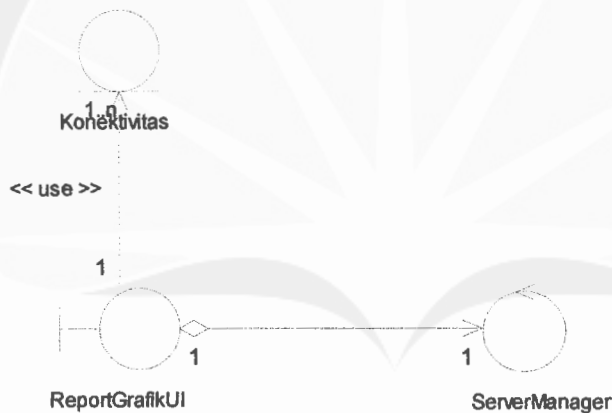
Gambar 4.4. Analysis Class Diagram : Use Case SettingBlocked

#### 4.1.5 Analysis Class Diagram : Use Case ReportStatistic



Gambar 4.5. Analysis Class Diagram : Use Case ReportStatistic

#### 4.1.6 Analysis Class Diagram : Use Case ReportGrafik



Gambar 4.6. Analysis Class Diagram : Use Case ReportGrafik

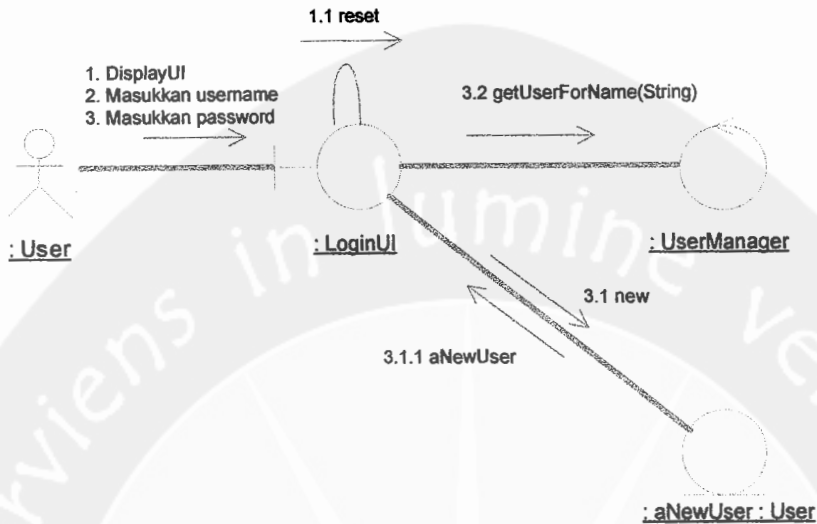
#### 4.1.7 Analysis Class Diagram : Use Case TesKoneksi



Gambar 4.8. Analysis Class Diagram : Use Case Koneksi

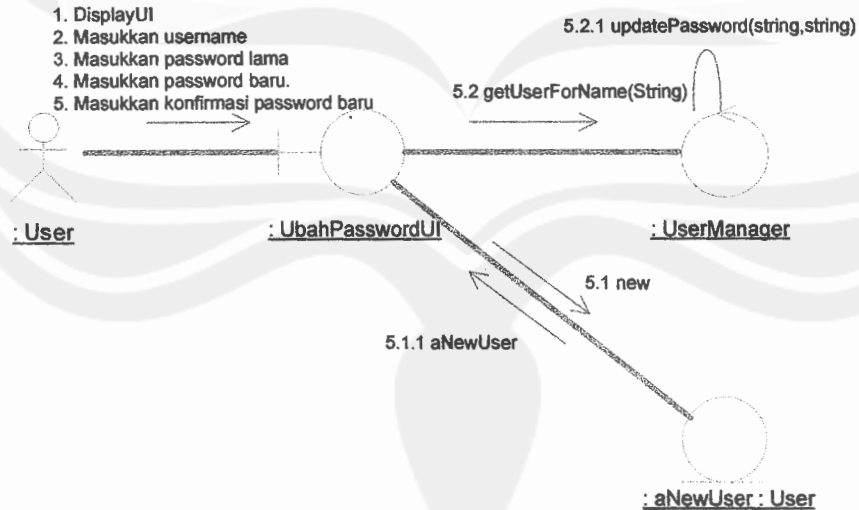
## 4.2 Interaction Diagram

### 4.2.1 Analysis Collaboration Diagram : Login



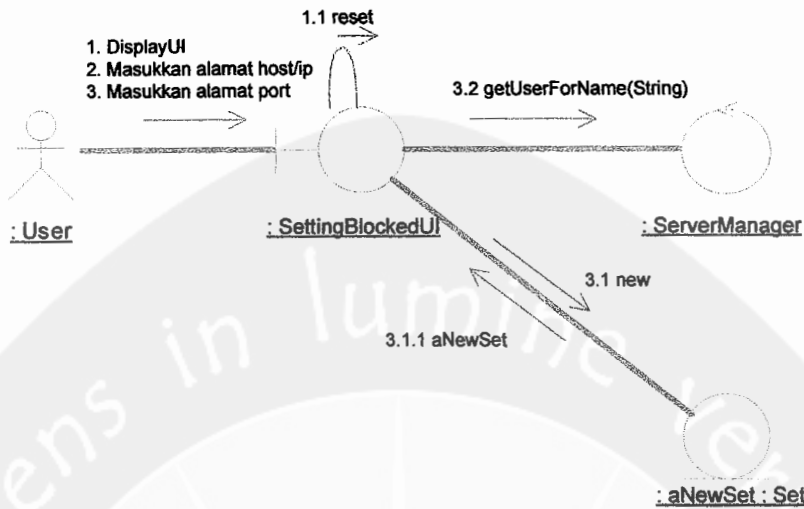
Gambar 4.2.1. Analysis Collaboration Diagram : Use Case Login

### 4.2.2 Analysis Collaboration Diagram : UbahPassword



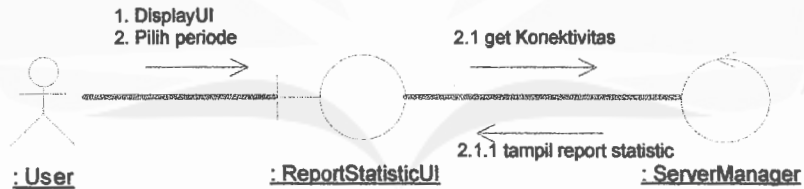
Gambar 4.2.2. Analysis Collaboration Diagram : Use Case UbahPassword

### 4.2.3 Analysis Collaboration Diagram : SettingBlocked



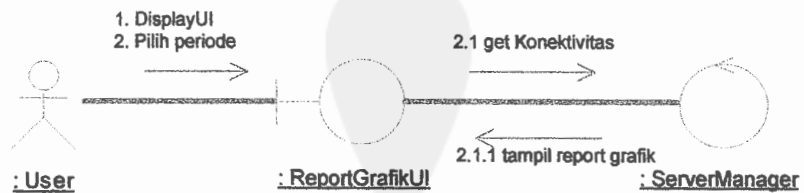
Gambar 4.2.3. Analysis Collaboration Diagram : Use Case SettingBlocked

### 4.2.4 Analysis Collaboration Diagram : ReportStatistic



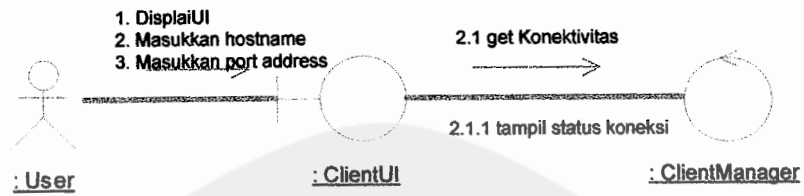
Gambar 4.2.4. Analysis Collaboration Diagram : Use Case ReportStatistic

### 4.2.5 Analysis Collaboration Diagram : ReportGrafik



Gambar 4.2.5. Analysis Collaboration Diagram : Use Case ReportGrafik

#### 4.2.6 Analysis Collaboration Diagram : TesKoneksi



Gambar 4.2.6. Analysis Collaboration Diagram : Use Case TesKoneksi

**DPPL**

## **DESKRIPSI PERANCANGAN PERANGKAT LUNAK**


**Intenet Protocol Bloked  
( IPBlocked )**

**Dipersiapkan oleh :**

**Yohanes David Widyatama**

**01 07 03093**

**Program Studi Teknik Informatika – Fakultas Teknologi Industri  
Universitas Atma Jaya Jogjakarta**

	<b>Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Atma Jaya Jogjakarta</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<b>GL01-G08</b>		<i>1 / 36</i>
	<b>Revisi</b>			<b>Tgl : 11-09-2007</b>

## DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis Oleh	David							
Diperiksa Oleh	YSP							
Disetujui Oleh	YSP							

## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi



# Daftar Isi

1	Pendahuluan.....	6
1.1	Tujuan.....	6
1.2	Lingkup Masalah.....	6
1.3	Definisi, Akronim dan Singkatan.....	7
1.4	Referensi.....	7
1.5	Deskripsi Umum (Overview).....	8
2	Deskripsi Perancangan Arsitektural.....	8
2.1	Deployment Diagram.....	8
2.1.1	Node : Client.....	8
2.1.2	Node : Server.....	8
2.2	Design Class.....	9
2.2.1	Pengantar.....	9
2.2.1	Package Dependencies.....	9
2.2.3	Package Pengelolaan User.....	10
2.2.3.1	Class Diagram Package IPBlocked.PengelolaanUser.....	10
2.2.3.2	Class IPBlocked.PengelolaanUser.User.....	10
2.2.3.3	Class IPBlocked.PengelolaanUser.UserManager.....	11
2.2.3.4	Class IPBlocked.IPGantiPassword.UbahPasswordUI.....	13
2.2.3.5	Class IPBlocked.PengelolaanUser.LoginUI.....	14
2.2.4	Package IPBlocked Server.....	14
2.2.4.1	Class Diagram Package IPBlocked.BlockedIP.....	15
2.2.4.2	Class IPBlocked.MonitoringIP.Konektivitas.....	16
2.2.4.3	Class IPBlocked.FilteringIP.ServerManager.....	18
2.2.4.4	Class IPBlocked.SettingBlocked.....	20
2.2.4.5	Class IPBlocked.FilteringIP.ReportStatisticUI.....	21
2.2.4.6	Class IPBlocked.FilteringIP.ReportGrafikUI.....	22
2.2.5	Package IPBlocked Client.....	22
2.2.5.1	Class IPBlocked.Client.Monitoring.NetworkTested.....	23
2.3	Realisasi Use Case.....	23
2.3.1	Use Case : Login.....	23
2.3.2	Use Case : Ubah Password.....	25
2.3.3	Use Case : Setting Blocked.....	26
2.3.4	Use Case : Report Statistic.....	27
2.3.5	Use Case : Report Grafik.....	28
2.3.6	Use Case : Tes Koneksi.....	29
3	Deskripsi Perancangan Persistent Data.....	30
3.1	Basis Data.....	30
3.1.1	Tabel : User.....	30
3.1.2	Tabel : Konektivitas.....	30
4	Deskripsi Perancangan Antarmuka.....	31
4.1	Use Case : Login.....	31

4.2	Use Case : UbahPassword .....	32
4.3	Use Case : Setting Blocked.....	33
4.4	Use Case : Report Statistic.....	34
4.5	Use Case : Report Grafik .....	35
4.6	Use Case : Tes Koneksi .....	36

## Daftar Gambar

Gambar 1.1	Skema Aplikasi IPBlocked.....	6
Gambar 1.2	Package Dependencies IPBlocked.....	9
Gambar 1.3	Class Diagram IPBlocked.PengelolaanUser.....	10
Gambar 1.4	Class Diagram IPBlocked.BlockedIP .....	15
Gambar 1.5	Use Case : Login .....	23
Gambar 1.6	Use Case : Ubah Password .....	25
Gambar 1.7	Use Case : Setting Blocked .....	26
Gambar 1.8	Use Case : Report Statistic .....	27
Gambar 1.5	Use Case : Report Grafik.....	28
Gambar 1.6	Use Case : Tes Koneksi.....	29
Gambar 1.7	ERD .....	30
Gambar 1.8	Gambar Use Case : Login.....	31
Gambar 1.9	Gambar Use Case : UbahPassword .....	32
Gambar 2.1	Gambar Use Case : Setting Blocked.....	33
Gambar 2.2	Gambar Use Case : ReportStatistic .....	34
Gambar 2.3	Gambar Use Case : Report Grafik.....	35
Gambar 2.4	Gambar Use Case : Tes Koneksi .....	36

## Daftar Tabel

Tabel 1.1	Tabel User Aplikasi IPBlocked.....	10
Tabel 1.2	Tabel UserManager Aplikasi IPBlocked.....	11
Tabel 1.3	Tabel GantiPassword.....	13
Tabel 1.4	Tabel LoginUI.Aplikasi IPBlocked.....	14
Tabel 1.5	Tabel Konektivitas.Aplikasi IPBlocked .....	16
Tabel 1.6	Tabel ServerManager.IPBlocked.FilteringIP .....	18
Tabel 1.7	Tabel SettingIPBlocked.IPBlocked.FilteringIP .....	20
Tabel 1.8	Tabel ReportStatistic.IPBlocked.FilteringIP .....	21
Tabel 1.9	Tabel ReportGrafik.IPBlocked.FilteringIP .....	22
Tabel 2.0	Tabel User .....	30
Tabel 2.1	Tabel Konektivitas .....	30

## Daftar Lampiran

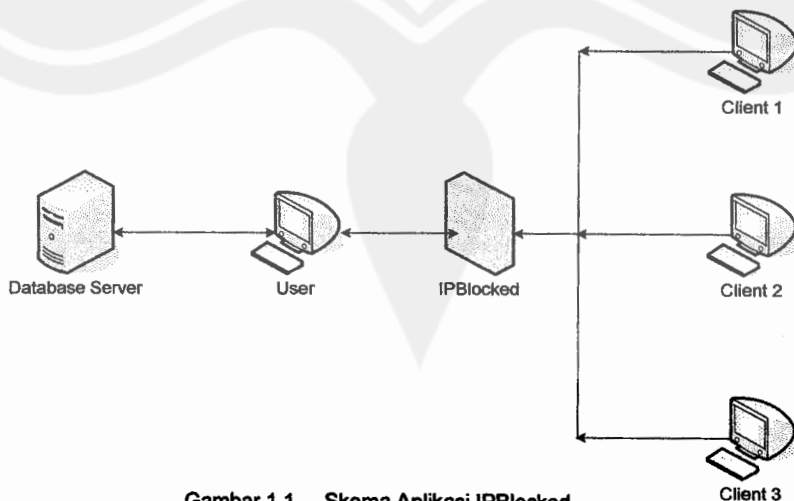
## 1. Pendahuluan

### 1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) dibuat sebagai dokumen untuk mendefinisikan deskripsi lengkap mengenai kebutuhan fungsionalitas dan kebutuhan non-fungsionalitas dari perangkat lunak *Aplikasi Internet Protocol Blocked* yang dibuat sebagai Tugas Akhir pada Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta. Dokumen ini berisi definisi fungsi perangkat lunak, juga mendefinisikan batasan perancangan perangkat lunak dan juga ditujukan untuk pembuat perangkat lunak dan orang-orang yang tertarik untuk melakukan berbagai evaluasi dan pengembangan sistem dikemudian harinya.

### 1.2 Lingkup Masalah

*Firewall* adalah sebuah sistem yang didesain untuk mencegah akses dari pihak yang tidak berhak (*unauthorized*) menuju atau dari jaringan lokal. *Firewall* dapat diimplementasikan, dalam bentuk *hardware*, *software*, atau kombinasi keduanya. *Firewall* biasanya digunakan untuk mencegah/mengendalikan aliran data tertentu. Semua data yang masuk ataupun yang keluar jaringan harus melewati firewall, yang akan memeriksa setiap paket apakah cocok atau tidak dengan kriteria/aturan yang ada pada standar keamanan yang didefinisikan dalam *firewall*.



Gambar 1.1 Skema Aplikasi IPBlocked  
(Sumber : David, 2007)

Aplikasi IPBlocked ini juga dapat dikatakan sebagai suatu firewall, karena aplikasi ini digunakan *user* untuk membatasi serta melakukan pengamanan terhadap hal-hal yang tidak diinginkan, melakukan *blok IP* yang masuk serta mengetahui *traffic* yang sedang berlangsung pada jaringan tersebut.

### 1.3 Definisi, Akronim dan Singkatan

- **DPPL** adalah Deskripsi Perancangan Perangkat Lunak, merupakan spesifikasi dari perangkat lunak yang digunakan.
- **DPPL- IPBlocked -XX** adalah Kode Internet Protocol Blocked Firewall berbasis Java Technology yang sedang dikembangkan, yang merepresentasikan kebutuhan pada **IPBlocked**.
- **IPBlocked**, merupakan aplikasi yang dikembangkan untuk pengguna jaringan komputer lokal.
- **GUI, Graphical User Interface**, merupakan antarmuka pemakai dengan sistem.
- **User**, merupakan pengguna aplikasi **IPBlocked** untuk melakukan pengamanan terhadap jaringan komputer yang digunakannya. Dalam hal ini, user dapat juga dikatakan sebagai Admin.

### 1.4 Referensi

1. GLO1, *Deskripsi Perancangan Perangkat Lunak*, Jurusan Teknik Informatika – UAJY.
2. David, Yohanes. *Proposal Aplikasi Firewall berbasis Java Technology*, Universitas Atma Jaya Yogyakarta, 2006.
3. Dharwiyanti, Sri, *Pengantar Unified Modeling Language (UML)*, Kuliah Umum Ilmu Komputer, 2003.

## 1.5 Deskripsi Umum ( Overview )

Secara umum dokumen ini terdiri dari empat bagian. Bagian pertama adalah **Pendahuluan**, yang berisi deskripsi dokumen. Bagian kedua adalah **Deskripsi Deskripsi Dekomposisi**, yang berisi deskripsi arsitektur sistem. Bagian ketiga adalah **Deskripsi Dekomposisi Modul**, yang berisi realisasi Use case – Use case yang digunakan dan Bagian 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* yang ada pada sistem, hubungan diantara mereka dan proses yang akan dijalankan masing-masing *node*.

#### 2.1.1 Node : Client

Client merupakan komputer yang digunakan untuk melihat adakah proses blok oleh *server* atau tidak terhadap *client*. Aktifitas yang dapat dilakukan Client hanyalah untuk melakukan test terhadap ip address maupun port address tertentu apakah sedang dilakukan blok atau tidak..

#### 2.1.2 Node : Server

*Server* merupakan komputer yang digunakan untuk melakukan filtering terhadap konektivitas yang terjadi dan melakukan pembuatan *report* terhadap data-data konektivitas yang ada didalamnya. Proses yang ada didalamnya adalah :

1. *IPBlocked*, merupakan program utama yang melakukan filtering dan blocked ip.
2. *SQLyog*, merupakan database yang digunakan.

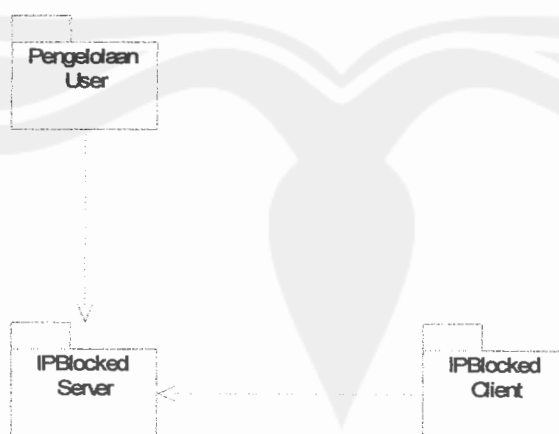
## 2.2 Design Class

### 2.2.1 Pengantar

Nama *class* yang digunakan dalam design class adalah nama class yang valid, termasuk nama packagenya. Stereotype yang digunakan dalam design *class* adalah :

- << boundary >>  
*Boundary* merupakan *class* yang berfungsi untuk menghubungkan sistem dengan user di luar sistem.
- << control >>  
*Control* class adalah suatu class yang objeknya 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*.

### 2.2.2 Package Dependencies

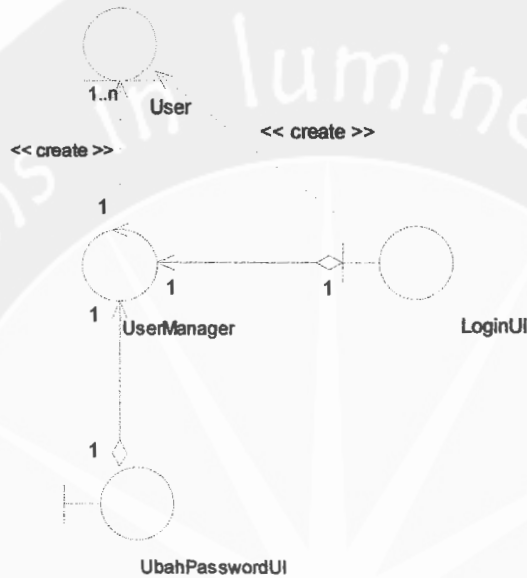


Gambar 1.2 Package Dependencies Aplikasi IPBlocked

### 2.2.3 Package Pengelolaan User (IPBlocked.PengelolaanUser)

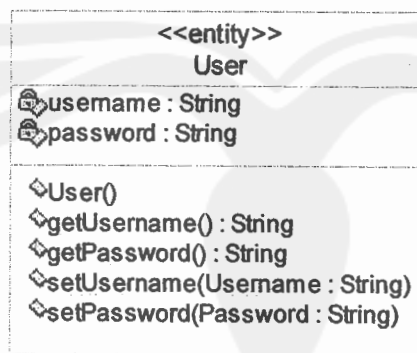
Package ini menyediakan class-class yang digunakan oleh program untuk mengelola hal-hal yang berhubungan dengan *user*.

#### 2.2.3.1 Class Diagram Package IPBlocked.PengelolaanUser



Gambar 1.3 Gambar Class Diagram IPBlocked.PengelolaanUser

#### 2.2.3.2 Class IPBlocked.PengelolaanUser.User



Tabel 1.1 Tabel User Aplikasi IPBlocked

#### Deskripsi

Class ini merepresentasikan data *user* pengguna sistem IPBlocked.

## Atribut

- - `username : string`  
Merepresentasikan *username* aktor.
- - `password : string`  
Merepresentasikan *password* aktor.

## Method

- + `User()`  
Konstruktor class `IPBlocked.PengelolaanUser.User`.  
Buat Instance baru `IPBlocked.PengelolaanUser.User` tanpa atribut terdefinisi.

Method-method berikut merupakan *accessor* untuk class `IPBlocked.PengelolaanUser.User` :

- + `getUsername () : string`
- + `getPassword () : string`

Method-method berikut merupakan *mutator* untuk atribut class `IPBlocked.PengelolaanUser.User` :

- + `setUsername(username:string)`
- + `setPassword(password:string)`

### 2.2.3.3 Class `IPBlocked.PengelolaanUser.UserManager`

<b>&lt;&lt;control&gt;&gt;</b> <b>UbahPassword</b>
◊UbahPassword(PassLama : String, PassBaru : String, PassVeri : String) : Boolean ◊KonfirmPass(PassLama : String, PassBaru ; String, PassVeri : String) : Boolean

Tabel 1.2 Tabel UserManager Aplikasi IPBlocked

## Deskripsi

Class yang berperan sebagai control class untuk aksi yang berhubungan dengan data user `IPBlocked`. Aksi yang berhubungan dengan *account* ini, antara lain :



- ganti password.
- validasi password.

### **Atribut**

-

### **Method**

- + UserManager ()  
Konstruktor *class IPBlocked.PengelolaanUser.UserManager*.  
Buat instance baru *IPBlocked.PengelolaanUser.UserManager*.

- + UbahPassword (PassLama : string, PassBaru : string, PassVeri : string) : boolean

Meng-up date *password* sistem dengan *password* baru. Untuk meng-up date *password*, harus memenuhi syarat-syarat berikut ini :

- *Password* lama harus sesuai dengan *password* sistem yang ada.
- *Password* baru (argument *newPass*) harus sesuai dengan konfirmasi *password* baru.
- Konfirmasi *Password* baru, harus sama dengan *password* baru.
- Panjang *password* baru minimal 30 karakter.

*Password* bersifat *case sensitive*.

#### **Parameters :**

*name -username* yang akan meng-up date *password* (user yang sedang melakukan LOGIN). Jika *username* dan *password* lama untuk data user ini tidak sesuai, method akan melemparkan exception, tanpa melakukan penggantian *password*.

*PassLama* - *password* lama sistem, minimal 30 karakter, bersifat *case sensitif*.

*PassBaru* - *password* baru, minimal 30 karakter, bersifat *case sensitive*.

PassVeri - konfirmasi password baru, minimal 10 karakter.

**Returns :**

True jika dan hanya jika password berhasil diubah untuk user yang sedang aktif.

- + KonfirmPassword(PassBaru : string, Pass Lama : string, PasVeri : string) : boolean

Validasi password apakah sama dengan konfirmasi passwordnya atau tidak.

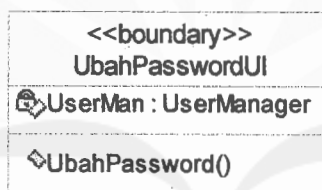
**Parameters :**

name - username yang akan di validasi data-nya.

**Returns :**

True jika dan hanya jika password baru dengan konfirmasi passwordnya sama.

### 2.2.3.4 Class IPBlocked.IPGantiPassword.UbahPasswordUI



Tabel 1.3 Tabel GantiPassword Aplikasi IPBlocked

#### Deskripsi

Class ini merupakan GUI yang digunakan untuk melakukan pengubahan *password* bagi *username* yang sedang aktif.

#### Atribut

- - userMan : IPBlocked.IPGantiPassword  
Merupakan *control class* untuk melakukan penggantian *password*.

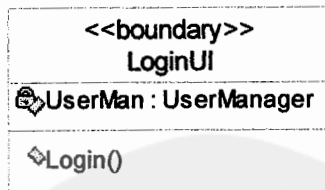
#### Method

- + UbahPasswordUI()

Buat instance baru

*IPBlocked.IPGantiPassword.UbahPasswordUI.*

### 2.2.3.5 Class IPBlocked.PengelolaanUser.LoginUI



Tabel 1.4 Tabel LoginUI Aplikasi IPBlocked

#### Deskripsi

Class ini merupakan GUI yang digunakan untuk validasi *username* dan *password* untuk Login.

#### Atribut

- - userMan : IPBlocked.IPLogin  
Merupakan *control class* untuk melakukan validasi *username* dan *password*.

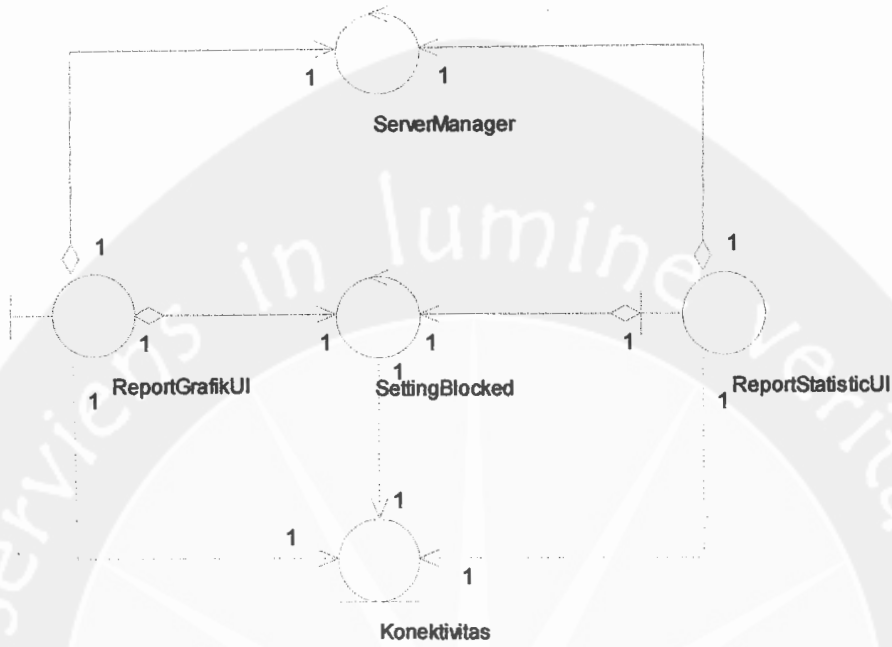
#### Method

- + LoginUI()  
Buat instance baru IPBlocked.PengelolaanUser.IPLoginUI.

### 2.2.4 Package IPBlocked Server (IPBlocked.FilteringIP)

Package ini menyediakan *class-class* yang diperlukan untuk men-set aturan yang digunakan untuk proses blocked, mengubah *password*, menampilkan statistik dari data konektivitas yang tersimpan pada data base dan menampilkannya pula dalam bentuk grafik.

### 2.2.4.1 Class Diagram Package IPBlocked.BlockedIP



Gambar 1.4 Gambar Class Diagram IPBlocked.BlockedIP

## 2.2.4.2 Class IPBlocked.MonitoringIP.Konektivitas



Tabel 1.5 Tabel Konektivitas Aplikasi IPBlocked

### Deskripsi

Class ini merepresentasikan data konektivitas sebagai data untuk reporting.

### Atribut

- - id\_network : integer  
Merepresentasikan id user pada saat konektivitas.
- - protocol : char  
Merepresentasikan protokol yang digunakan untuk koneksi.
- - local\_address : varchar  
Merepresentasikan alamat asal dari koneksi.
- - local\_port : varchar

Merepresentasikan nomor port asal yang digunakan.

- - `foreign_address` : varchar

Merepresentasikan alamat tujuan koneksi.

- - `foreign_port` : varchar

Merepresentasikan nomor port tujuan koneksi.

- - `time_in` : datetime

Merepresentasikan waktu awal koneksi.

- - `time_out` : datetime

Merepresentasikan waktu akhir koneksi.

### Method

- + `Konektivitas()`

Konstruktor *class* `IPBlocked.FilteringIP.Konektivitas`.

Buat instance baru `IPBlocked.FilteringIP.Konektivitas` tanpa atribut terdefinisi.

Method-method berikut merupakan *accessor* untuk *class* `IPBlocked.FilteringIP.Konektivitas` :

- + `getid_network():integer`
- + `getProtokol():char`
- + `getLocalAddress():varchar`
- + `getLocalPort():varchar`
- + `getForeignAddress():varchar`
- + `getForeignPort():varchar`
- + `getTime_in():datetime`
- + `getTime_out():datetime`

Method-method berikut merupakan *accessor* untuk *class* `IPBlocked.FilteringIP.Konektivitas` :

- + `setid_network(id_network : integer)`
- + `setProtokol(protokol : char)`

- + setLocalAddress(local\_address : varchar)
- + setLocalPort(local\_port : varchar)
- + setForeignAddress(foreign\_address : varchar)
- + setForeignPort(foreign\_port : varchar)
- + setTimeIn(tgltime\_in : datetime)
- + setTimeOut(time\_out : datetime)

### 2.2.4.3 Class IPBlocked.FilteringIP.ServerManager

<<control>> ServerManager
<ul style="list-style-type: none"> <li>◆ ServerManager()</li> <li>◆ getClient() : DataSet</li> <li>◆ getKonektivitasClient(Address : String, tgl1 : Date, tgl2 : Date) : DataSet</li> <li>◆ getKonektivitas(tgl1 : Date, tgl2 : Date) : DataSet</li> <li>◆ getJmlConnection(conn : String, tgl1 : Date, tgl2 : Date) : Integer</li> <li>◆ getJmlStatus(Status : String, tgl1 : Date, tgl2 : Date) : Integer</li> </ul>

Tabel 1.6 Tabel ServerManager IPBlocked.FilteringIP

#### Deskripsi

*Class* yang berperan sebagai *control class* untuk aksi yang berhubungan dengan pengelolaan data konektivitas pada *IPBlocked FilteringIP*.

#### Atribut

-

#### Method

- + ServerManager
- + getClient() : DataSet

Ambil semua data *client* pada basis data.

#### Returns :

Set data yang berisi semua data client yang tersimpan pada basis data.

- + getKonektivitasClient(address:string, tgl1:date, tgl2:date) : DataSet

**Ambil semua data client pada basis data untuk nomor client tertentu.**

**Parameters :**

address - address client yang akan diblok.

tgl1 - tanggal koneksi client yang akan diblok.

tgl2 - tanggal koneksi client yang akan diblok.

**Returns :**

Set data yang berisi semua data client yang tersimpan pada basis data.

- + getKonektivitas(tgl1:date, tgl2:date) : DataSet

**Ambil semua data konektivitas pada basis data.**

**Parameters :**

tgl1 - tanggal koneksi client yang akan diblok.

tgl2 - tanggal koneksi client yang akan diblok.

**Returns :**

Set data yang berisi semua data konektivitas yang tersimpan pada basis data.

- + getJmlConnection(con:string, tgl1:date, tgl2:date) : integer

**Menghitung jumlah koneksi dari client.**

**Parameters :**

con - nama connection yang akan dihitung jumlahnya.

tgl1 - tanggal koneksi client yang akan diambil datanya.

tgl2 - tanggal koneksi client yang akan diambil datanya.

**Returns :**

Jumlah connection dari client.

- + getJmlStatus(status:string, tgl1:date, tgl2:date) : integer

**Menghitung jumlah status dari client.**

**Parameters :**

status - status dari client yang akan dihitung jumlahnya.



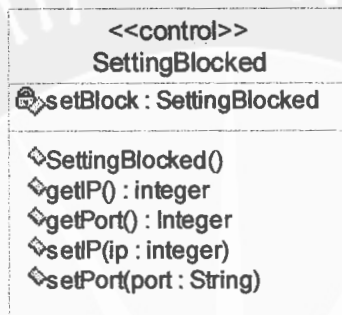
tgl1 - tanggal koneksi client yang akan diambil datanya.

tgl2 - tanggal koneksi client yang akan diambil datanya.

**Returns :**

Jumlah status dari client.

#### 2.2.4.4 Class IPBlocked.SettingBlocked



Tabel 1.7 Tabel SettingBlocked IPBlocked.FilteringIP

#### Deskripsi

Class yang berperan sebagai *control class* untuk aksi yang berhubungan dengan setting *rule/aturan* yang digunakan untuk melakukan filtering / blocking IP.

#### Atribut

-

#### Method

- + SettingBlocked
- + addRule (blocked:IPBlocked.SettingBlocked) : boolean

Digunakan untuk menambahkan rule/aturan baru.

**Parameters :**

blocked - IPBlocked.SettingBlocked yang akan ditambahkan.

**Returns :**

True jika IPBlocked.SettingBlocked berhasil ditambahkan.

- + AddBlock(ip\_address:varchar; port:int):boolean  
Digunakan untuk menambahkan data *client* baru yang akan diblok ke dalam *list client*.

**Parameters :**

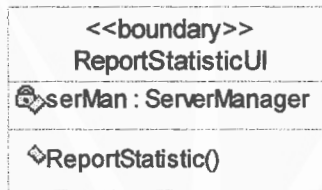
ip - ip yang akan ditambahkan dalam aturan blok.

Port - port yang akan ditambahkan dalam aturan blok.

**Returns :**

True , jika aturan baru berhasil ditambahkan.

### 2.2.4.5 Class IPBlocked.FilteringIP.ReportStatisticUI



Tabel 1.8 Tabel ReportStatistic IPBlocked.FilteringIP

**Deskripsi**

Class ini merupakan GUI yang digunakan untuk melihat data konektivitas semua *client*.

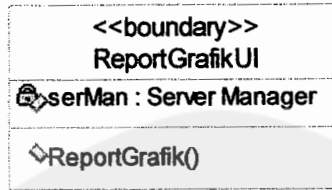
**Atribut**

- - serMan : IPBlocked.IPStatistic  
Merupakan control class untuk melihat data konektivitas semua *client*.

**Method**

- + ReportIPStatistic()  
Buat instance baru  
IPBlocked.IPStatistic.ReportIPStatisticUI.

### 2.2.4.6 Class IPBlocked.FilteringIP. ReportGrafikUI



Tabel 1.9 Tabel ReportGrafik IPBlocked.FilteringIP

#### Deskripsi

Class ini merupakan GUI yang digunakan untuk melihat data konektivitas *client* dalam bentuk grafik.

#### Atribut

- - serMan : IPBlocked.FilteringIP.ReportGrafik  
Merupakan control class untuk melihat data konektivitas *client* dalam bentuk grafik..

#### Method

- + ReportGrafikUI()  
Buat instance baru IPBlocked.FilteringIP.  
ReportGrafikUI.

### 2.2.5 Package IPBlockedClient (IPBlocked.Monitoring)

Package ini menyediakan class-class yang diperlukan untuk melakukan *monitoring* terhadap *filtering/blocking ip* yang dilakukan.



Gambar 1.1 Gambar IPBlockedClient.Monitoring

### 2.2.5.1 ClassIPBlockedClient.Monitoring.Network Tested

#### Deskripsi

Class ini merupakan GUI yang digunakan *client* untuk melihat apakah proses blok terhadap ip dan *port address* tertentu yang telah dilakukan berhasil atau tidak.

#### Atribut

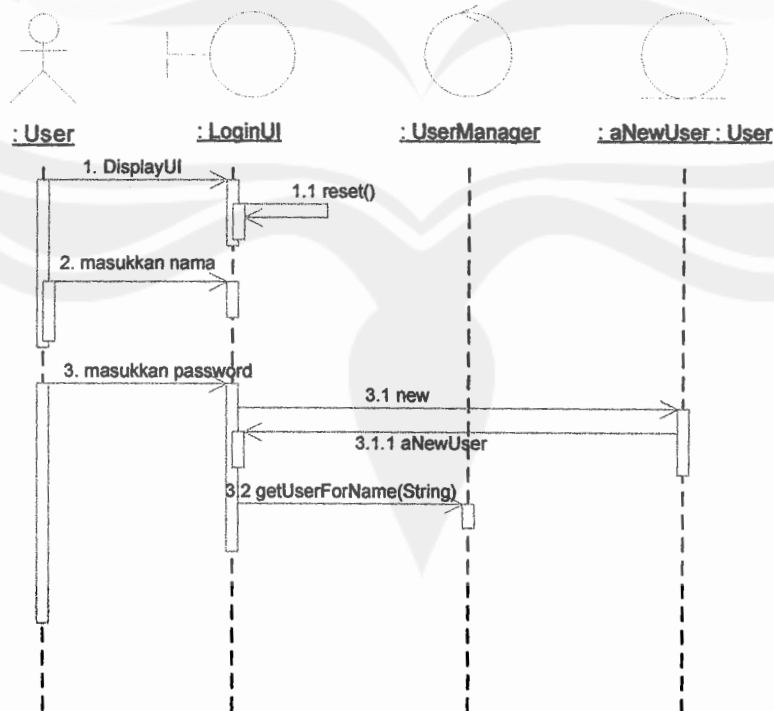
- - serMan : IPBlocked.NetworkTested.clientManager  
Merupakan control class untuk melihat proses blok yang dilakukan.

#### Method

- + Network TestedUI ()  
Buat instance baru  
IPBlocked.NetworkTested.ReportNetworkTestedUI.

## 2.3 Realisasi Use Case

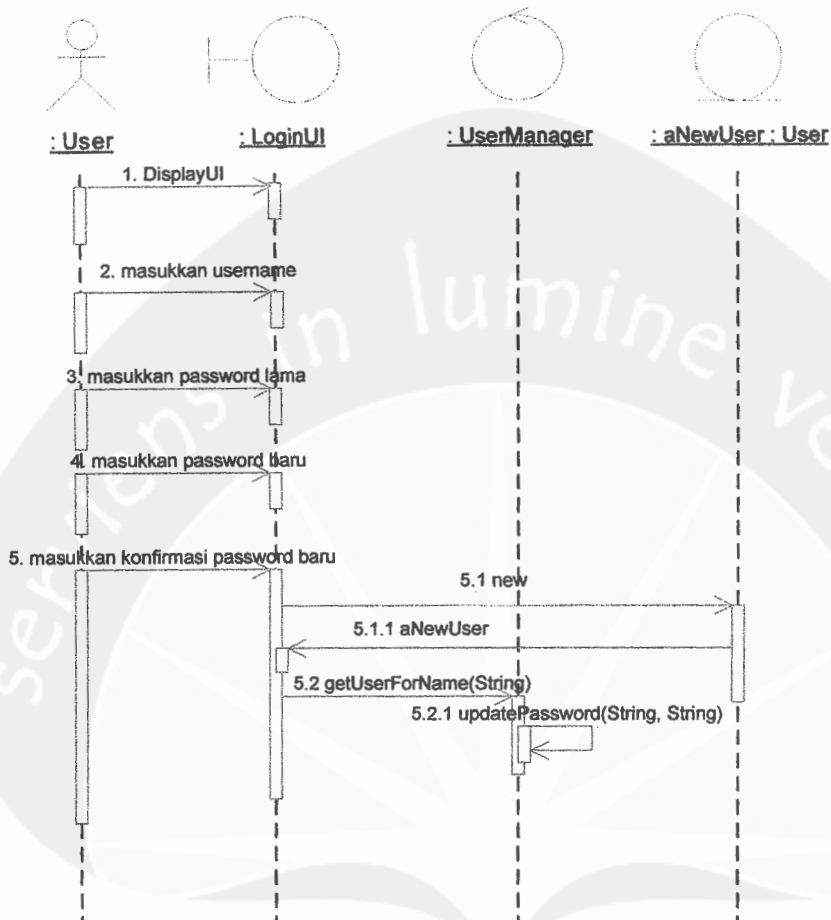
### 2.3.1 Use Case : Login



*Flow of events :*

1. Aktor menampilkan antarmuka untuk login, yaitu boundary class loginUI. LoginUI mengeset ulang (reset) tampilannya dengan mengosongkan textbox yang ada pada login.
2. Aktor memasukkan username dan password. Username ini unik untuk masing-masing administrator, dan password merupakan rangkaian karakter dengan panjang 5-20 karakter dan bersifat *case sensitive*.
3. Dengan menekan "OK", maka administrator memicu sistem untuk melakukan validasi melalui LoginUI. LoginUI membentuk aNewUser, yang merupakan instance baru class User. LoginUI melakukan validasi terhadap username dan password yang diinputkan, yang merupakan atribut dari aNewUser. UserManager memvalidasi aNewUser ini dengan memanggil method `getUserForName(string)` milik UserManager, yang akan memeriksa di basis data, apakah ada data dengan field *username* sesuai *username* aNewAccount. Jika ada, maka sistem akan membandingkan *password username* tersebut dari basis data dengan *password input user* dari aNewUser. User akan dapat memasuki sistem bila *username* dan *passwordnya* sesuai.

### 2.3.2 Use Case : UbahPassword

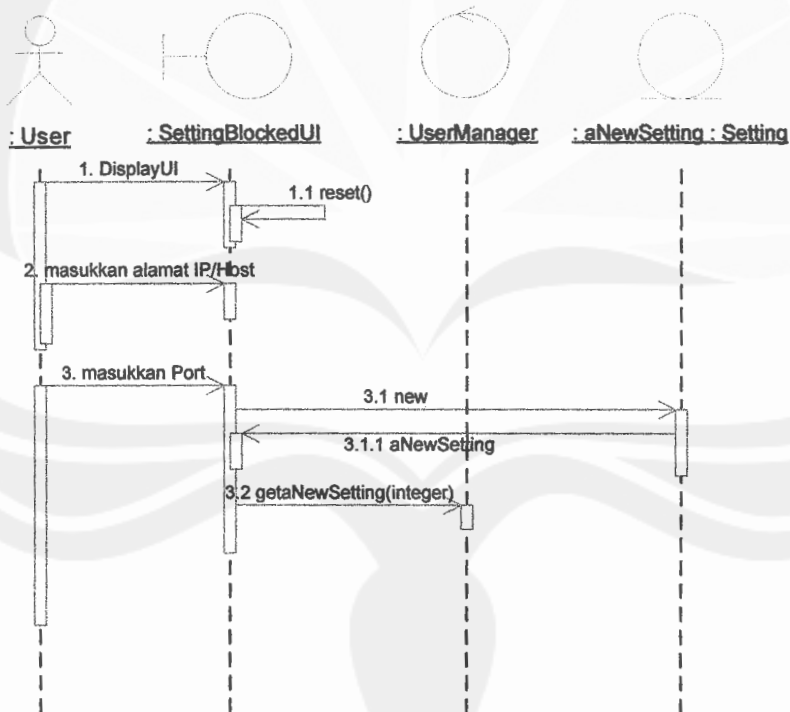


Flow of events :

1. Aktor menampilkan antarmuka untuk ubah *password*, yaitu boundary class *UbahPasswordUI*.
2. Aktor memasukkan *password* lama.
3. Aktor memasukkan *password* baru.
4. Aktor memasukkan konfirmasi password baru.
5. Aktor menekan tombol “*SAVE*” kemudian user memicu sistem untuk mengubah password. *UbahPasswordUI* membentuk objek user baru, yaitu *aNewUser*, untuk meng-enkapsulasi *username* dan *password* lama. *UserPasswordUI* kemudian memanggil method *getUserForName(string)* untuk mendapatkan data *user*. Lalu

UserManager melakukan perubahan *password* dengan memanggil method *updatePassword* (string, string). Method ini akan melakukan validasi *username* dan *password* lama milik objek *aNewUser*. Jika *username* dan *password* lama valid, maka UserManager akan melakukan *validasi* terhadap *password* baru dan konfirmasi *password* baru, yaitu membandingkan keduanya, harus sama persis (*case sensitive*). Jika validasi *password* baru dan konfirmasi *password* baru ini berhasil, maka UserManager akan mengubah data *password* pada basis data untuk *username* yang sesuai, dengan *password* baru tersebut.

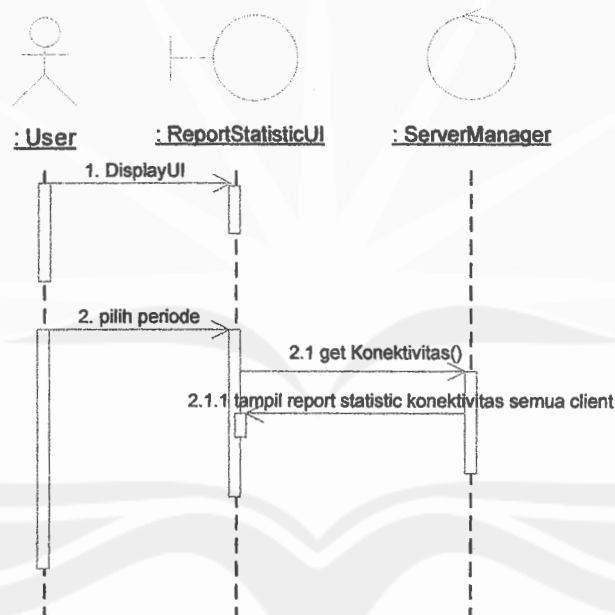
### 2.3.3 Use Case : Setting Bloked



*Flow of events :*

1. Aktor menampilkan antarmuka untuk *setting blocked*, yaitu *boundary class SettingBlockedUI*.
2. Aktor meng-inputkan *hostname*..
3. Aktor meng-inputkan *port address*.
4. Aktor menekan tombol “*SAVE*”.
5. Sistem akan menyimpan aturan tersebut sebagai aturan baru yang akan melakukan *filtering* terhadap *hostname* dan *port* yang dimaksud.

### 2.3.4 Use Case : Report Statistic

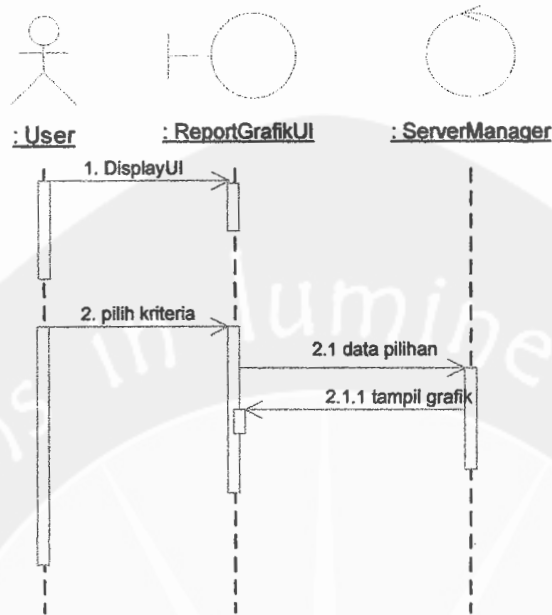


*Flow of events :*

1. Aktor menampilkan antarmuka untuk Statistik reporting, yaitu *boundary class ReportStatisticUI*.
2. Aktor memicu sistem untuk menampilkan report statistik.



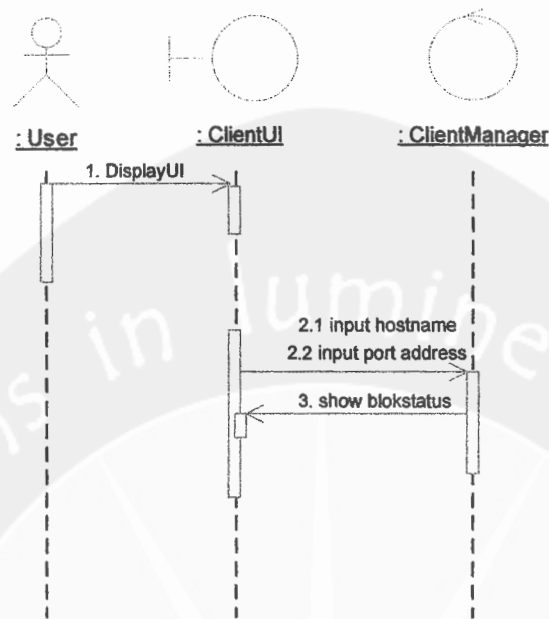
### 2.3.5 Use Case : Report Grafik



*Flow of events :*

1. Aktor menampilkan antarmuka untuk grafik reporting, yaitu boundary class *ReportGrafikUI*.
2. Aktor memilih semua kriteria yang ada, yang akan ditampilkan reportnya. Berdasarkan pilihan ini, *GrafikReportUI* akan mengirimkan data pilihannya ke *ServerManager*.
3. Aktor memicu sistem untuk menampilkan report grafik.

### 2.3.6 Use Case : Tes Koneksi

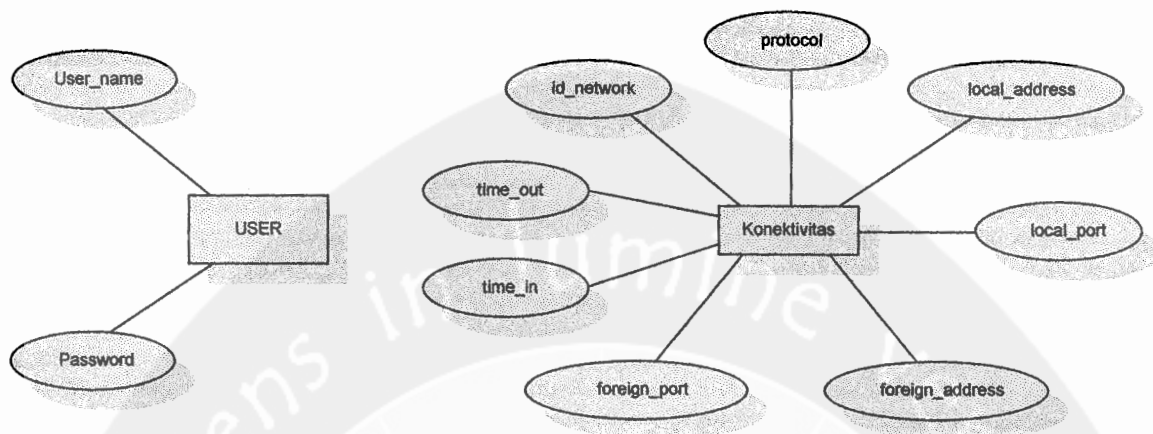


#### Flow of events :

1. Aktor menampilkan antarmuka untuk melakukan koneksi, yaitu *boundary class NetworkTestedUI*.
2. Aktor memasukkan alamat IP dan port.
3. Aktor memicu sistem untuk melakukan tes koneksi dengan menggunakan method *connect hostname* dan *connect ip address*. Kemudian setelah koneksi berhasil dilakukan, sistem akan segera mengirimkan data konektivitasnya apakah *failed* atau *connecting*.

### 3. Deskripsi Perancangan Persistent Data

#### 3.1 Basis Data



##### 3.1.1 Tabel User

Tabel ini merepresentasikan entitas user.

Field	Tipe data	NULL	Deskripsi
Uname	Varchar (20)	No	Username untuk IPBlocked. Merupakan rangkaian karakter dengan panjang 5-20 karakter. Primary key.
Passwd	Varchar (30)	No	Password untuk user IPBlocked. Merupakan rangkaian karakter dengan panjang 5-20 karakter.

##### 5.1.2 Tabel Konektivitas

Tabel ini merepresentasikan entitas Konektivitas.

Field	Tipe data	NULL	Deskripsi
Id_network	int(11)	No	Nomor id client pada jaringan.
Protokol	char (3)	YES	Protokol yang digunakan client untuk melakukan koneksi.

Local_address	varchar (15)	YES	Alamat asal dari suatu koneksi yang dilakukan.
Local_port	varchar(30)	YES	Port asal yang digunakan client untuk melakukan koneksi.
foreign_address	varchar (15)	YES	Alamat tujuan dari koneksi yang dilakukan client.
foreign_port	varchar (30)	YES	Port tujuan koneksi yang dilakukan client / port server.
time_in	datetime	YES	waktu saat koneksi dilakukan oleh client.
time_out	datetime	YES	Waktu saat client menghentikan koneksi.

#### 4. Deskripsi Perancangan Antarmuka

##### 4.1 Use Case : Login

The image shows a login form with the following elements:

- A label "User name" followed by a text input field.
- A label "Password" followed by a text input field.
- Two buttons at the bottom: "LOGIN" and "CANCEL".

##### Deskripsi

- Rancangan antarmuka ini diimplementasikan pada class *IPBlocked.PengelolaanUser.LoginUI*. Antarmuka ini digunakan pada use case Login.
- Text box atas, digunakan untuk menerima masukan *username*.
- Text box bawah, digunakan untuk menerima masukan *password*.
- Tombol Login, digunakan untuk aksi login *user*.
- Tombol "Cancel", digunakan untuk pembatalan aksi login dan menutup antarmuka LoginUI.

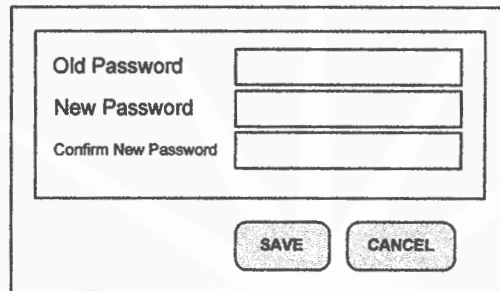
## Event

- Login.

Urutan aksi yang terjadi :

1. *User* memasukkan username pada textbox atas.
2. *User* memasukkan password pada textbox bawah.
3. *User* menekan tombol Login.
4. Sistem memeriksa dibasis data, apakah ada baris pada basis data dengan *username* yang sama dengan *username* yang diinputkan.

## 4.2 Use Case : Ubah Password



The image shows a user interface for changing a password. It consists of a rectangular box containing three text input fields. The first field is labeled 'Old Password', the second 'New Password', and the third 'Confirm New Password'. Below these fields are two buttons: 'SAVE' and 'CANCEL'.

### Deskripsi

- Rancangan antarmuka ini diimplementasikan pada class *IPBlocked.GantiPassword.GantiPasswordUI*. Antarmuka ini hanya dapat diakses oleh administrator sistem. Antarmuka ini digunakan pada use case ubah *password* untuk mengubah *password* user pengguna sistem/admin.
- User memasukkan *password* lama pada textbox atas.
- User memasukkan *password* baru pada textbox tengah.
- User memasukkan Konfirmasi *password* baru pada textbox bawah.
- User menekan tombol “*SAVE*”, maka sistem akan terlebih dulu mencocokkan *password* baru dengan konfirmasi password baru. Jika sama, maka sistem akan menyimpannya pada database.
- Tombol “*CANCEL*”, digunakan untuk membatalkan.

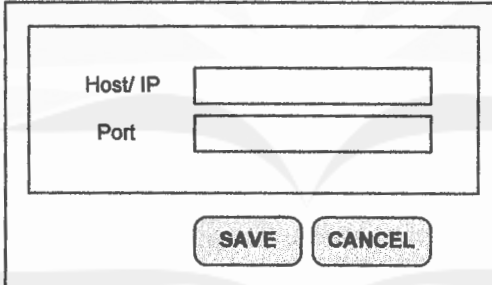
## Event

- Penggantian *Password*

Urutan aksi yang terjadi :

1. User memasukkan *password* lama.
2. User memasukkan *password* baru.
3. User memasukkan konfirmasi *password* baru.
4. User menekan tombol “*SAVE*” untuk menyimpan.
5. Sistem memeriksa kesamaan antara *password* baru dengan konfirmasi *password* baru, jika sama *password* baru akan disimpan, jika tidak akan muncul pernyataan bahwa *password* baru dan konfirmasinya belum sama.
6. User menekan tombol “*CANCEL*” untuk membatalkan proses.

### 4.3 Use Case : Setting Blocked



The image shows a graphical user interface for setting blocked rules. It consists of a rectangular frame containing two text input fields. The top field is labeled 'Host/ IP' and the bottom field is labeled 'Port'. Below these fields are two buttons: 'SAVE' and 'CANCEL'.

#### Deskripsi

- Rancangan antarmuka ini diimplementasikan pada class *IPBlocked.SettingBlocked.SettingBlockedUI*. Antarmuka ini hanya dapat diakses oleh administrator sistem. Antarmuka ini digunakan pada use case setting blocked, untuk mengelola *rule* /aturan yang digunakan sebagai filtering.
- *User* memasukkan Alamat *Host/IP* pada kolom text box atas.
- *User* memasukkan *Port* pada text box bawah.
- Tombol *Save*, digunakan untuk aksi penambahan rule / aturan filetering koneksi.



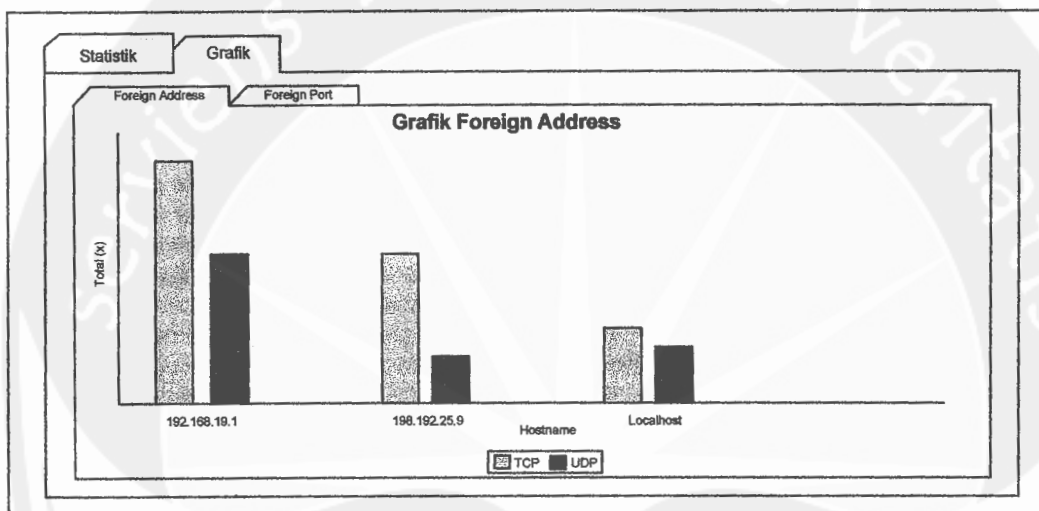
## Event

- Report Blocked

Urutan aksi yang terjadi :

1. User memilih tab “Statistik”.
2. Sistem menampilkan statistik yang dihasilkan dari blocked ip yang dilakukan.

## 4.5 Use Case : Report Grafik



### Deskripsi

- Rancangan antarmuka ini diimplementasikan pada class *IPBlocked.Monitoring.GrafikUI*. Antarmuka ini digunakan pada use case ReportGrafik.
- Tab Grafik digunakan User/Admin untuk melihat grafik yang dihasilkan dari data statistic yang dilakukan dalam proses blocked ip tersebut.

## Event

- Report Grafik

Urutan aksi yang terjadi :

1. User memilih tab “Grafik”, untuk melihat tampilan grafik yang dihasilkan berdasarkan data statistik yang ada.



## 4.6 Use Case : Tes Koneksi

The screenshot shows a window titled "Network Tested". Inside the window, there are two text input fields: "Hostname" and "Port". To the right of the "Port" field is a button labeled "Connect". Below these fields is a large, empty rectangular area, likely intended for displaying the results of the connection test.

### Deskripsi

- Rancangan antarmuka ini diimplementasikan pada class *IPBlocked.NetworkTested.ClientUI*. Atarmuka ini digunakan pada use case tes koneksi pada client.
- *Client* memasukkan alamat IP pada textbox hostname.
- *Client* memasukkan alamat Port pada textbox port.
- Tombol Connect digunakan untuk melakukan tes proses blok terhadap alamat ip dan port yang telah diinputkan, apakah berhasil atau tidak.

### Event

- Tes Koneksi.

Urutan aksi yang terjadi :

1. User meng-inputkan alamat IP dan alamat *port* yang akan dilakukan tes koneksi.
2. User menekan tombol *connect*.
3. Sistem akan segera melakukan tes apakah input yang diberikan tadi telah berhasil di-blok atau tidak