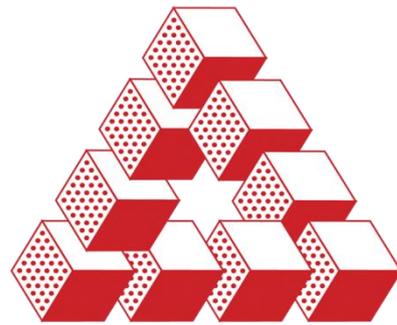
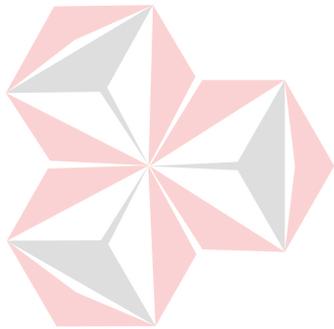


**RANCANG BANGUN APLIKASI MONITORING TRAFIK DAN  
PERFORMA SERVER PERMAINAN KOMPUTER.  
(STUDI KASUS : WARCRAFT III)**



**STIKOM  
SURABAYA**

UNIVERSITAS

**Dinamika**

Oleh:

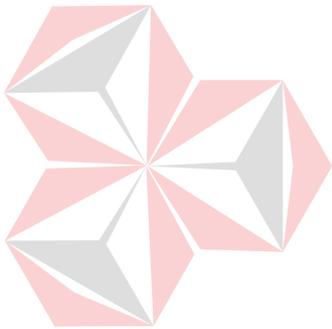
Nama : Rendy Praditya  
NIM : 06.41010.0241  
Program : S1 (Strata Satu)  
Jurusan : Sistem Informasi

**SEKOLAH TINGGI  
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER  
SURABAYA  
2011**

**RANCANG BANGUN APLIKASI MONITORING TRAFIK DAN  
PERFORMA SERVER PERMAINAN KOMPUTER.  
(STUDI KASUS : WARCRAFT III)**

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Komputer



Oleh :

UNIVERSITAS  
**Dinamika**

Nama : Rendy Praditya

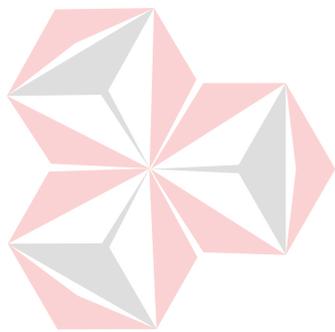
NIM : 06.41010.0241

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

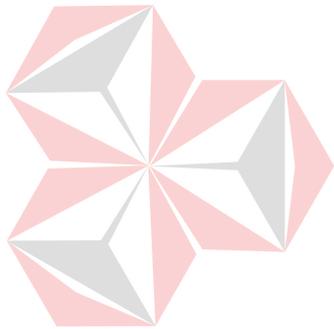
**SEKOLAH TINGGI  
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER  
SURABAYA**

**2011**



UNIVERSITAS  
**Dinamika**

*Gagal dalam sebuah pertempuran akan lebih ksatria, daripada gagal sebelum sempat menarik pedang. (Julius Caesar)*



Ku persembahkan kepada

Bapak dan Ibu saya tercinta

Saudara-saudaraku tercinta

Johan Silvanus Regar, S.Kom

Wira Syahputra, S.Kom

Adhie Septiadi, S.Kom

Rene Adi Kusumo Alie, S.Kom

Chrisandy Masta, S.Kom

Hevea Bori Perdana

Rekan – rekan SCS

Semua yang telah mendukungku selama ini

UNIVERSITAS  
Dinamika

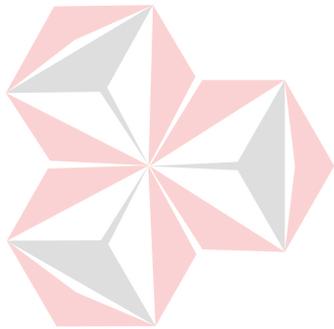
**RANCANG BANGUN APLIKASI MONITORING TRAFIK DAN  
PERFORMA SERVER PERMAINAN KOMPUTER.  
(STUDI KASUS : WARCRAFT III)**

Disusun Oleh :

**Nama : Rendy Praditya**

**Nim : 06.41010.0241**

Surabaya, April 2011



UNIVERSITAS  
Telah diperiksa, diuji dan disetujui :  
**Dinamika**

Wakil Ketua Bidang Akademik

Pembimbing

**Pantjawati Sudarmaningtyas, S.Kom**  
NIDN. 0712066801

**Anjik Sukmaaji, S.Kom.,M.Eng**  
NIDN 0731057301

## ABSTRAK

Warcraft III adalah permainan komputer *multiplayer* yang berbasiskan RTS (*Real Time Strategy*), yang menjadi salah satu cabang permainan lomba kompetisi dunia yaitu WCG (*World Championship Game*). Permainan Warcraft III biasa dimainkan secara *multiplayer*, untuk koneksi lokal atau LAN (*local area network*) permainan Warcraft III mampu menampung 10 orang pemain dalam satu *room*, begitu juga untuk koneksi internet, puluhan *room* permainan banyak sekali ditemukan jika pemain menggunakan koneksi internet, oleh karena itu diperlukan aplikasi yang mampu melakukan proses monitoring trafik yang digunakan oleh permainan Warcraft III.

Proses monitoring trafik dan performa permainan Warcraft III adalah proses *capture packet data* dan *memory usage* yang digunakan permainan. Selain itu aplikasi monitor trafik dan performa *server* Warcraft III juga berfungsi untuk mengestimasi *bandwidth* yang tepat untuk dimainkan di internet. Program monitor trafik dan performa *server* Warcraft III nantinya juga dapat mengestimasi *resource hardware* yang digunakan untuk *server* permainan.

Dalam pengembangan aplikasi penulis melakukan beberapa kali ujicoba yang menghasilkan *file database sniffer* yang berisi informasi *bandwidth* dan informasi performa CPU , *physical memory*, *private memory workset* dan *harddisk*. Informasi *bandwidth* dan performa kemudian diolah oleh aplikasi *server* sehingga menghasilkan laporan *bandwidth* dan performa *server*.

*Keyword : Warcraft III, Network Monitor, Sniffer, Performance Monitor, Bandwidth*

## KATA PENGANTAR

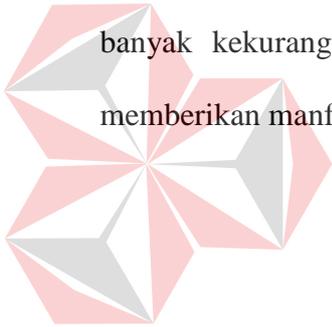
Segala puji syukur penulis panjatkan kehadirat Allah S.W.T, karena berkat Rahmat dan Hidayah-Nya tugas akhir yang merupakan persyaratan dalam menyelesaikan Program Studi Strata Satu di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM). Dalam penyelesaian tugas akhir yang berjudul “Rancang Bangun Aplikasi Monitoring Trafik dan Performa Server Permainan Komputer (Studi Kasus Warcraft III)” ini, tidak terlepas dari bantuan dari berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan rasa terima kasih dan penghargaan kepada :

1. Bapak, Ibu dan Kakak tercinta, yang senantiasa memberikan semangat dan doa kepada penulis selama pengerjaan tugas akhir ini.
2. Bapak Prof.Dr.Budi Jatmiko, M.Pd. selaku ketua STIKOM Surabaya.
3. Ibu Pantjawati Sudarmaningtyas, S.Kom. selaku Waka Bidang Akademik STIKOM Surabaya.
4. Bapak Erwin Sutomo, S.Kom selaku Kaprodi S1 Sistem Informasi.
5. Bapak Anjik Sukmaaji S.Kom,.M.Eng sebagai dosen pembimbing, atas segala arahan, waktu dan bimbingan yang diberikan kepada penulis, dalam penyelesaian tugas akhir ini.
6. Bapak Teguh Sutanto, M.Kom. atas segala arahan, waktu dan bimbingan yang diberikan kepada penulis khususnya desain sistem UML.
7. Bapak Romeo, S.T atas segala arahan, waktu dan bimbingan yang diberikan kepada penulis khususnya testing dan implementasi sistem.

8. Rekan – rekan di PT. Teknologi Informasi Indonesia yang telah membantu penulis atas idenya dan memberikan dukungan teknis dalam penyelesaian tugas akhir ini.
9. Teman – teman di SCS yang memberikan semangat dan motivasi selama pengerjaan Tugas Akhir ini.
10. Semua pihak yang namanya tidak bisa disebutkan satu persatu, atas bantuan langsung maupun bantuan tidak langsung dalam penyelesaian tugas akhir ini.

Semoga Allah SWT memberikan pahala yang besar kepada semua pihak yang telah banyak memberikan bantuan, ataupun nasehat-nasehat.

Akhirnya, penulis menyadari dalam pembuatan tugas akhir ini masih banyak kekurangan. Namun penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi semua pihak yang membutuhkannya.



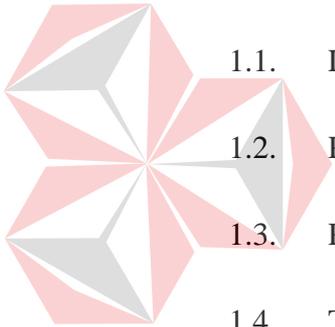
UNIVERSITAS  
**Dinamika**

Surabaya, Maret 2011

Penulis

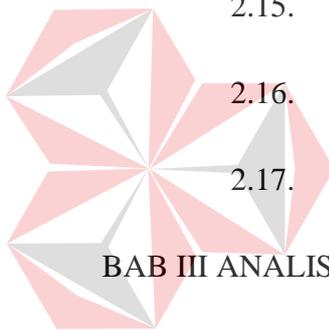
## DAFTAR ISI

|   |      |
|---|------|
| ABSTRAK .....   | vi   |
| KATA PENGANTAR .....                                  | vii  |
| DAFTAR ISI.....                                       | ix   |
| DAFTAR TABEL.....                                     | xii  |
| DAFTAR GAMBAR .....                                   | xiv  |
| DAFTAR LAMPIRAN.....                                  | xvii |
| BAB I PENDAHULUAN .....                               | 1    |
| 1.1. Latar Belakang Masalah .....                     | 1    |
| 1.2. Perumusan Masalah .....                          | 3    |
| 1.3. Pembatasan Masalah.....                          | 3    |
| 1.4. Tujuan .....                                     | 4    |
| 1.5. Sistematika Penulisan .....                      | 4    |
| BAB II LANDASAN TEORI .....                           | 6    |
| 2.1. MODEL TCP/IP .....                               | 6    |
| 2.2. Protokol TCP/IP.....                             | 7    |
| 2.3. IP (Internet Protocol) Header.....               | 11   |
| 2.4. TCP (Transmission Control Protocol) Header ..... | 12   |
| 2.5. UML (Unified Modelling Language) .....           | 15   |
| 2.6. Sniffer .....                                    | 17   |

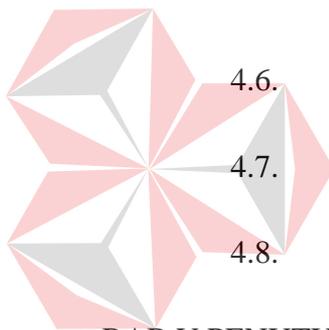


UNIVERSITAS  
Dinamika

|  |   |           |
|--|---|-----------|
| 2.7.   | Bandwidth.....  | 18        |
| 2.8.   | Throughput .....                                      | 19        |
| 2.9.   | Permainan Komputer .....                              | 20        |
| 2.10.  | Real Time Strategy (RTS) .....                        | 21        |
| 2.11.  | Performance Indicator .....                           | 22        |
| 2.12.  | Permodelan Visual.....                                | 24        |
| 2.13.  | Unified Modelling Language (UML) .....                | 25        |
| 2.14.  | Pengenalan diagram-diagram dalam UML.....             | 26        |
| 2.15.  | Network Monitor Untuk Mengawasi Trafik Jaringan ..... | 26        |
| 2.16.  | Monitor Network Trafik .....                          | 27        |
| 2.17.  | Sistem Monitor .....                                  | 27        |
| <b>BAB III ANALISA DAN PERANCANGAN SISTEM.....</b> |   | <b>29</b> |
| 3.1.   | Analisa Permasalahan .....                            | 29        |
| 3.2.   | Perancangan Sistem .....                              | 30        |
| 3.3.   | Blok Diagram Aplikasi Monitoring.....                 | 30        |
| 3.4.   | Flow Chart Monitoring Trafik .....                    | 31        |
| 3.5.   | Flow Chart Monitor Performa .....                     | 32        |
| 3.6.   | Flow Chart Sniffer .....                              | 33        |
| 3.7.   | Flow Chart Maintenance Data Master Komputer.....      | 34        |
| 3.8.   | Use case Model.....                                   | 34        |



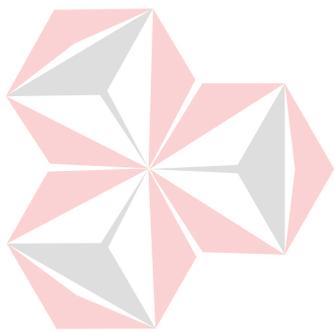
|                                       |  |     |
|---------------------------------------|--|-----|
| 3.9.                                  | Struktur Basis Data .....  | 56  |
| 3.10.                                 | Desain User Interface .....  | 59  |
| BAB IV IMPLEMENTASI DAN EVALUASI..... |  | 72  |
| 4.1.                                  | Implementasi.....  | 72  |
| 4.2.                                  | Kebutuhan Sistem.....  | 72  |
| 4.3.                                  | Instalasi Program dan Pengaturan Sistem.....                               | 73  |
| 4.4.                                  | Penjelasan Pemakaian Aplikasi. ....  | 74  |
| 4.5.                                  | Uji Coba Fungsionalitas Sistem Monitor Performa dan Trafik<br>Server ..... | 87  |
| 4.6.                                  | Evaluasi Hasil Uji Coba Sistem.....  | 88  |
| 4.7.                                  | Scenario Base Testing Untuk Performa Server .....                          | 88  |
| 4.8.                                  | Evaluasi Sistem.....   | 97  |
| BAB V PENUTUP.....                    |  | 105 |
| 5.1.                                  | Kesimpulan .....   | 105 |
| 5.2.                                  | Saran .....  | 105 |
| DAFTAR PUSTAKA .....                  |  | 107 |
| LAMPIRAN.....                         |  | 110 |



## DAFTAR TABEL

|   |     |
|---|-----|
| Tabel 2.1 Kondisi Pada Beberapa Kontrol Bit.....            | 13  |
| Tabel 3.1 Struktur Tabel Packet Data .....                  | 56  |
| Tabel 3.2 Struktur Tabel Respon .....                       | 57  |
| Tabel 3.3 Struktur Tabel Information .....                  | 57  |
| Tabel 3.4 Struktur Tabel Bandwidth.....                     | 58  |
| Tabel 3.5 Struktur Tabel Master Komputer .....              | 58  |
| Tabel 3.6 Struktur Tabel Performa.....                      | 59  |
| Tabel 4.1 Uji Coba Performa Server.....                     | 88  |
| Tabel 4.2 Data Performa .....                               | 90  |
| Tabel 4.3 Uji Coba Bandwidth Server.....                    | 93  |
| Tabel 4.4 Data Bandwidth.....                               | 95  |
| Tabel 4.5 Performa CPU Server.....                          | 97  |
| Tabel 4.6 Performa Physical Memory Server .....             | 97  |
| Tabel 4.7 Performa Warcraft III Private Memory Workset..... | 98  |
| Tabel 4.8 Performa Harddisk Read Server .....               | 98  |
| Tabel 4.9 Performa Harddisk Write Server.....               | 99  |
| Tabel 4.10 Performa CPU Max.....                            | 99  |
| Tabel 4.11 Performa CPU Min .....                           | 100 |
| Tabel 4.12 Kebutuhan Bandwidth Upload Server .....          | 100 |
| Tabel 4.13 Kebutuhan Bandwidth Download Server .....        | 101 |
| Tabel 4.14 Kebutuhan Bandwidth Average Server.....          | 101 |
| Tabel 4.15 Kebutuhan Bandwidth Peak Server .....            | 101 |

Tabel 4.16 Kebutuhan Bandwidth Broadcast Server ..... 102



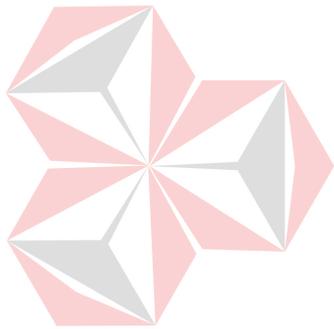
UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Layer TCP/IP .....   | 9  |
| Gambar 2.2 Pergerakan data dalam layer TCP/IP .....                   | 10 |
| Gambar 2.3 Bagan Internet Protocol.....                               | 11 |
| Gambar 2.4 Bagan TCP .....  | 13 |
| Gambar 2.5 Model 4+1 <i>view</i> .....                                | 16 |
| Gambar 2.6 Contoh notasi UML.....                                     | 25 |
| Gambar 3.1 Blok Diagram Aplikasi Monitoring .....                     | 30 |
| Gambar 3.2 Flow Chart Monitoring Trafik.....                          | 31 |
| Gambar 3.3 Flow Chart Monitoring Performa.....                        | 32 |
| Gambar 3.4 Flow Chart Sniffer.....                                    | 33 |
| Gambar 3.5 Flow Chart Maintenance Data Master Komputer .....          | 34 |
| Gambar 3.6 Use Case Diagram Aplikasi Monitoring .....                 | 35 |
| Gambar 3.7 Flow of Event Monitor Bandwidth .....                      | 37 |
| Gambar 3.8 Flow of Event Sniffing Packet Data.....                    | 38 |
| Gambar 3.9 Flow of Event Monitor Performa .....                       | 39 |
| Gambar 3.10 Flow of Event Menyimpan Data Master Komputer.....         | 40 |
| Gambar 3.11 Flow of Event Merubah Data Master Komputer.....           | 42 |
| Gambar 3.12 Flow of Event Melihat Laporan Bandwidth.....              | 43 |
| Gambar 3.13 Flow of Event Melihat Laporan Performa .....              | 44 |
| Gambar 3.14 Flow of Event Melihat Laporan Grafik Garis Bandwidth..... | 45 |
| Gambar 3.15 Sequence Menyimpan Data Master Komputer .....             | 46 |
| Gambar 3.16 Sequence Merubah Data Master Komputer .....               | 47 |

|  |    |
|--|----|
| Gambar 3.17 Sequence Sniffing packet Data .....                  | 48 |
| Gambar 3.18 Sequence Monitor Performa.....                       | 49 |
| Gambar 3.19 Sequence Monitor Bandwidth.....                      | 51 |
| Gambar 3.20 Sequence Menampilkan laporan Performa .....          | 52 |
| Gambar 3.21 Sequence Menampilkan Laporan Bandwidth .....         | 53 |
| Gambar 3.22 Class Diagram Monitor Bandwidth and Performa.....    | 54 |
| Gambar 3.23 Sequence Menampilkan Laporan Grafik Bandwidth.....   | 55 |
| Gambar 3.24 Desain Form Utama .....                              | 59 |
| Gambar 3.25 Desain Form Server Performance Monitoring.....       | 61 |
| Gambar 3.26 Desain Form Report performance .....                 | 62 |
| Gambar 3.27 Desain Form Data Monitor Performa.....               | 63 |
| Gambar 3.28 Desain Form Client and Server Bandwidth Usage .....  | 64 |
| Gambar 3.29 Desain Form Master Client dan Server .....           | 65 |
| Gambar 3.30 Desain Form Report Bandwidth.....                    | 66 |
| Gambar 3.31 Desain Data Monitor Trafik .....                     | 67 |
| Gambar 3.32 Desain Form Client dan Server Bandwidth Offline..... | 68 |
| Gambar 3.33 Desain Form Report Grafik Garis Bandwidth.....       | 69 |
| Gambar 3.34 Desain Form Sniffer .....                            | 70 |
| Gambar 3.35 Desain Form Laporan Tabel Perbandingan.....          | 71 |
| Gambar 4.1 Form Menu Utama .....                                 | 74 |
| Gambar 4.2 Form Proses Manajer. ....                             | 76 |
| Gambar 4.3 Form Laporan Performa .....                           | 77 |
| Gambar 4.4 Form Data Monitor Performa.....                       | 78 |
| Gambar 4.5 Form Client dan Server Bandwidth.....                 | 79 |

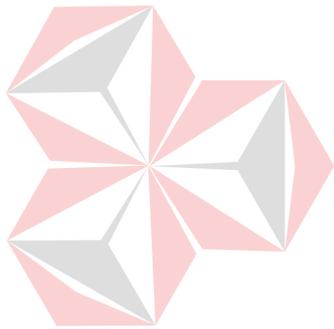
|   |    |
|---|----|
| Gambar 4.6 Form Master Client dan Server .....            | 80 |
| Gambar 4.7 Form Laporan Bandwidth .....                   | 81 |
| Gambar 4.8 Form Client dan Server Bandwidth Offline ..... | 82 |
| Gambar 4.9 Form List Data Monitor Trafik .....            | 83 |
| Gambar 4.10 Form Report Grafik Garis Bandwidth.....       | 84 |
| Gambar 4.11 Form Sniffer .....                            | 85 |
| Gambar 4.12 Form Laporan Tabel Perbandingan.....          | 86 |



UNIVERSITAS  
**Dinamika**

## DAFTAR LAMPIRAN

|                                   |     |
|-----------------------------------|-----|
| Lampiran 1. Listing Program ..... | 110 |
|-----------------------------------|-----|



UNIVERSITAS  
**Dinamika**

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

*Game* komputer atau disebut sebagai permainan *video* yang ada didalam sebuah *personal computer*, konsol permainan, *handphone*, *game watch* atau mesin permainan lainnya seperti *ding-dong* atau *PlayStation*. Evolusi *game* sangat cepat dari sistem grafis sederhana sampai yang sangat kompleks dan mutakhir. Evolusi yang cepat ini tidak hanya mengembangkan kualitas sistem gambar tetapi juga sistem permainannya. Salah satu evolusi yaitu permainan yang melibatkan banyak pemain atau *multiplayer game*. *Multiplayer game* dapat diartikan para pemain bermain bersamaan diwaktu yang sama disatu lokasi yang sama atau berbeda melalui jaringan lokal atau internet. Sejalan dengan pesatnya perkembangan *game*, membuat kebutuhan *hardware* meningkat. Semakin tinggi tingkat kompleksitas sistem permainan dan mutakhir membutuhkan *resource* yang besar, sehingga kebutuhan grafis, *processor* dan *memory* menjadi besar. Kebutuhan *hardware* tidak terbatas untuk permainan *offline* saja, jika permainan dilakukan *online* kebutuhan *bandwidth* menjadi masalah karena semakin tinggi kompleksitas *game* ditambah jumlah pemain yang banyak menjadi masalah tersendiri.

*Game Multiplayer* yang sedang populer saat ini adalah *Warcraft III* buatan developer terkenal yaitu *Blizzard*. *Warcraft III* adalah *game RTS (Real Time Strategy)* yang melibatkan 10 pemain dalam satu sesi permainan yang terbagi menjadi dua kelompok yaitu *Sentinel* dan *Scourge*. Permainan *Warcraft III* biasa

dimainkan melalui jaringan lokal maupun internet. Saat dimainkan dengan jaringan lokal permainan Warcraft III tidak mengalami kendala karena kapasitas jaringan yang dihasilkan sangat besar, berkisar antara 10 sampai 300 Mbps. Kendala akan muncul jika permainan Warcraft III dimainkan melalui internet dengan jumlah pemain yang banyak, sehingga muncul masalah *Lag* atau *Spike* yang sering pada jaringan internet yang tidak stabil. *Lag* yang terlalu sering bisa berakibat putusnya permainan ditengah-tengah. Masalah *Lag* atau *Spike* sangat merugikan pemain, selain mengurangi kenyamanan saat bermain juga aturan dari admin *server* yang memberikan sanksi untuk pemain yang meninggalkan permainan secara sengaja atau tidak sengaja yang umumnya dikarenakan meninggalkan permainan saat koneksi putus atau tidak stabil. Jadi tujuan mendasar untuk menjaga kualitas *bandwidth* atau QoS (*Quality of Service*) adalah mendapatkan kebutuhan *bandwidth* untuk setiap *client* dan kebutuhan untuk *server* yang terkoneksi dalam satu jaringan atau berbeda.

Masalah lainnya yaitu performa *server* dari permainan Warcraft III. Permainan Warcraft III menggunakan *resource* yang besar yang dipengaruhi oleh kompleksitas dan jumlah pemain, untuk itu pengukuran kebutuhan *hardware* untuk *server* menjadi sulit untuk ditentukan.

Dengan masalah diatas, maka dibuatlah sebuah program *client – server* untuk memonitoring dan mengestimasi besarnya *bandwidth* yang dibutuhkan oleh setiap *client* dan *server* Warcraft III sehingga *client* dan *server* dapat memperkirakan *bandwidth* yang tepat untuk dimainkan di internet. Program monitor trafik dan performa *server* Warcraft III nantinya juga dapat mengestimasi

kebutuhan *resource* performa *server* sehingga *hardware* yang dibutuhkan untuk *server* dapat memenuhi kebutuhan sesuai dengan jumlah pemain.

## 1.2. Perumusan Masalah

Perumusan masalah dalam Tugas Akhir ini adalah :

1. Bagaimana membuat aplikasi sistem monitoring permainan jaringan Warcraft III.
2. Bagaimana membuat aplikasi untuk mengetahui kebutuhan performa perangkat keras yang digunakan untuk komputer server permainan.

## 1.3. Pembatasan Masalah

Adapun batasan masalah yang digunakan sebagai berikut:

1. Aplikasi ini dibangun pada jaringan berbasis protocol TCP/IP.
2. Monitoring *hardware* dari proses manajer (*Windows Task Manager*).
3. Monitoring jaringan tidak memasukkan *latency* dan *delay* karena hal tersebut tidak hanya dipengaruhi oleh kualitas jaringan tetapi juga dipengaruhi oleh kualitas *hardware* masing-masing komputer *client*.
4. Monitoring performa server permainan Warcraft III tidak meliputi VGA dan perangkat keras lainnya, hanya *processor*, *harddisk* dan *physical memory*.
5. *Hardware* komputer yang dimonitor tidak meliputi perbandingan dan keunggulan merk dagang.
6. Map yang digunakan pada permainan Warcraft III adalah Dota versi 6.xx

#### 1.4. Tujuan

Tujuan dari Tugas Akhir ini adalah :

1. Membuat aplikasi monitoring jaringan permainan Warcraft III dengan model visual grafis dan tabel.
2. Membuat aplikasi untuk mengidentifikasi kebutuhan performa *server* sehingga mengetahui *resource hardware* yang dibutuhkan.

#### 1.5. Sistematika Penulisan

Sistematika penulisan tugas akhir yang berjudul “Rancang Bangun Aplikasi Monitoring Trafik dan Performa Server Permainan Komputer (Studi

Kasus : Warcraft III)” sebagai berikut:

##### **BAB I : PENDAHULUAN**

Bab ini menjelaskan gambaran umum penulisan yang berisi latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan, dan keterangan mengenai sistematika penulisan.

##### **BAB II : LANDASAN TEORI**

Bab ini akan menjelaskan tentang teori yang mendukung pokok pembahasan tugas akhir yang meliputi antara lain jaringan komputer, teori *game* komputer, *sniffer*, teori performa komputer, dan juga teori tentang desain sistem.

##### **BAB III : ANALISA DAN PERANCANGAN SISTEM**

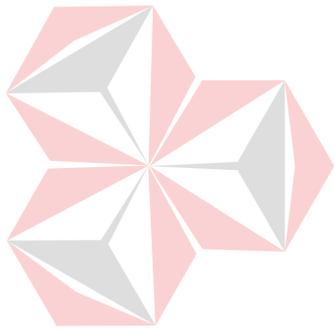
Bab ini menjelaskan mengenai analisis permasalahan, desain arsitektur, perancangan pengambilan data, diagram alir, *Block Diagram*, struktur *database*, *UML*, dan *Entity Relationship Diagram*

**BAB IV : IMPLEMENTASI DAN EVALUASI**

Bab ini menjelaskan tentang implementasi dari program, berisikan langkah-langkah implementasi dari perancangan program dan hasil implementasi dari program.

**BAB V : PENUTUP**

Bab ini berisi kesimpulan yang menjawab pernyataan dalam perumusan masalah dan beberapa saran yang bermanfaat dalam pengembangan program diwaktu yang akan datang.



UNIVERSITAS  
**Dinamika**

## BAB II

### LANDASAN TEORI

#### 2.1. MODEL TCP/IP

Model ini terdiri dari empat lapisan, berbeda dengan model *Open System Interconnection* (OSI) yang diprakarsai oleh panitia untuk membuat standarisasi, model TCP/IP ini berasal dari pekerjaan praktis para peneliti yang terlibat dalam membangun arsitektur jaringan. Model TCP/IP tidak memiliki Presentation dan Session Layer. Fungsi dari kedua lapisan ini dapat dilakukan sesuai kebutuhan oleh protokol TCP/IP yang berbeda. Mulai dari atas, lapisan-lapisan dari model TCP/IP yaitu:

1. *Application Layer*, sebagaimana pada model OSI pemakai berinteraksi dengan aplikasi jaringan pada lapisan ini. Data diterima dari pemakai oleh aplikasi jaringan serta dianggap sebagai komando dan sebagai data yang datang dari luar (peer node). Pada lapisan ini, aplikasi TCP/IP menggunakan model client/server.
2. *Transport Layer*, lapisan ini berfungsi mengatur alur data diantara dua node yang saling berhubungan. Protokol yang digunakan pada lapisan ini adalah *Transmission Control Protocol* (TCP) untuk penyampaian data.
3. *Network Layer*, data dikirimkan pada lapisan ini. *Internet Protocol* (IP) beroperasi pada lapisan ini untuk menentukan rute yang tidak bergantung kepada medium jaringan. Paket tersebut dilewatkan melalui internetwork.
4. *Data Link Layer*, pada lapisan ini data ditransmisikan melalui suatu jaringan

tunggal. Data yang berasal dari Network Layer, tiba pada jaringan lokal (kadang tidak meninggalkan jaringan sama sekali) dan ditransmisikan ke alamat tujuan.(Suryadi, 1997:10)

## 2.2. Protokol TCP/IP

Pada dasarnya komunikasi data merupakan proses mengirimkan data dari satu komputer ke komputer yang lain. Untuk dapat mengirimkan data, pada komputer harus ditambahkan alat khusus, yang dikenal sebagai network interface (interface jaringan). Jenis interface jaringan ini bermacam-macam, bergantung pada media fisik yang digunakan untuk mentransfer data tersebut.

Dalam proses pengiriman data ini terdapat beberapa masalah yang harus dipecahkan. Pertama, data harus dapat dikirimkan ke komputer yang tepat, sesuai tujuannya. Hal ini akan menjadi rumit jika komputer tujuan transfer data ini tidak berada pada jaringan lokal, melainkan di tempat yang jauh. Jika lokasi komputer yang saling berkomunikasi "jauh" (secara jaringan) maka terdapat kemungkinan data rusak atau hilang. Karenanya, perlu ada mekanisme yang mencegah rusaknya data.

Hal lain yang perlu diperhatikan ialah, pada komputer tujuan transfer data mungkin terdapat lebih dari satu aplikasi yang menunggu datangnya data. Data yang dikirim harus sampai ke aplikasi yang tepat, pada komputer yang tepat, tanpa kesalahan.

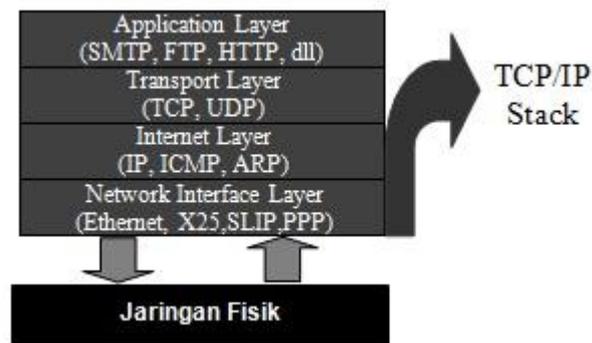
Cara alamiah, untuk menghadapi setiap masalah yang rumit ialah memecahkan masalah tersebut menjadi bagian yang lebih kecil. Dalam memecahkan masalah transfer data di atas, para ahli jaringan komputer pun

melakukan hal yang sama. Untuk setiap problem komunikasi data, diciptakan solusi khusus berupa aturanaturan untuk menangani problem tersebut. Untuk menangani semua masalah komunikasi data, keseluruhan aturan ini harus bekerja sama satu dengan lainnya. Sekumpulan aturan untuk mengatur proses pengiriman data ini disebut sebagai *protokol komunikasi data*. Protokol ini diimplementasikan dalam bentuk program komputer (software) yang terdapat pada komputer dan, peralatan komunikasi data lainnya.

TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada *Wide Area Network* (WAN). TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Berkat prinsip ini tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, sepanjang ia masih bisa saling mengirim dan menerima data.

Berkat penggunaan prinsip ini, TCP/IP menjadi protokol komunikasi data yang fleksibel. Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan interface jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Agar TCP/IP dapat berjalan di atas interface jaringan tertentu, hanya perlu dilakukan perubahan path protokol yang berhubungan dengan interface jaringan saja.

Sekumpulan protokol TCP/IP ini dimodelkan dengan empat layer TCP/IP, sebagaimana terlihat pada gambar di bawah ini.



Gambar 2.1 Layer TCP/IP

TCP/IP terdiri atas empat lapis kumpulan protokol yang tertingkat.

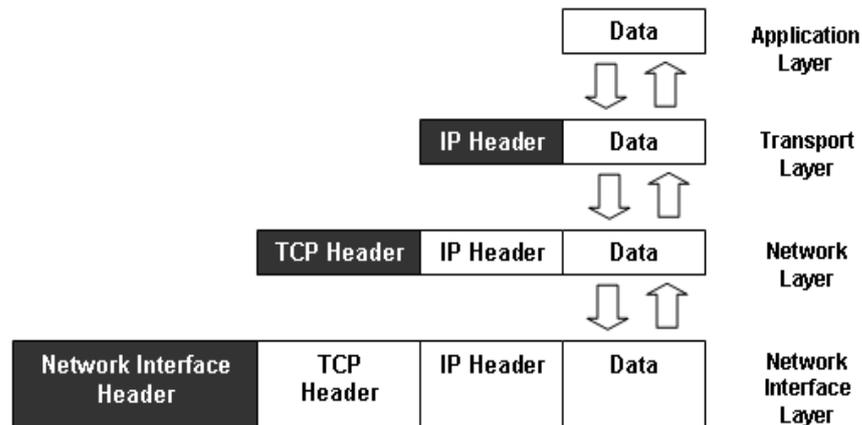
Keempat lapis/layer tersebut adalah:

1. Network Interface Layer
2. Internet Layer
3. Transport Layer
4. Application Layer

Dalam TCP/IP, terjadi penyampaian data dan protokol yang berada di satu layer ke protokol yang berada di layer yang lain. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data

Jika suatu protokol menerima data dari protokol lain di layer atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu, data ini diteruskan lagi ke protokol path layer di bawahnya.

Hal yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada layer di bawahnya. Jika data ini dianggap valid, protokol akan melepas informasi tambahan tersebut, untuk kemudian meneruskan data itu ke protokol lain yang berada pada layer di atasnya.



Gambar 2.2 Pergerakan data dalam layer TCP/IP

Lapisan/Layer terbawah, yaitu *Network Interface layer*, bertanggung jawab mengirim dan menerima data dari media fisik. Media fisiknya dapat berupa kabe!, serat optik, atau gelombang radio. Karena tugasnya ini, protokol pada layer ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan lain yang sejenis.

Lapisan/Layer protokol berikutnya ialah *Internet Layer*. Protokol yang berada pada layer ini bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada layer ini terdapat tiga macam protokol, yaitu IP, ARP, dan ICMP.

IP (Internet Protocol) berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan untuk menemukan alamat hardware dari komputer yang terletak pada network yang sama. Sedangkan ICMP (Internet Control Message Protocol) ialah protokol yang digunakan untuk mengirimkan pesan dan melaporkan kegagalan pengiriman data.

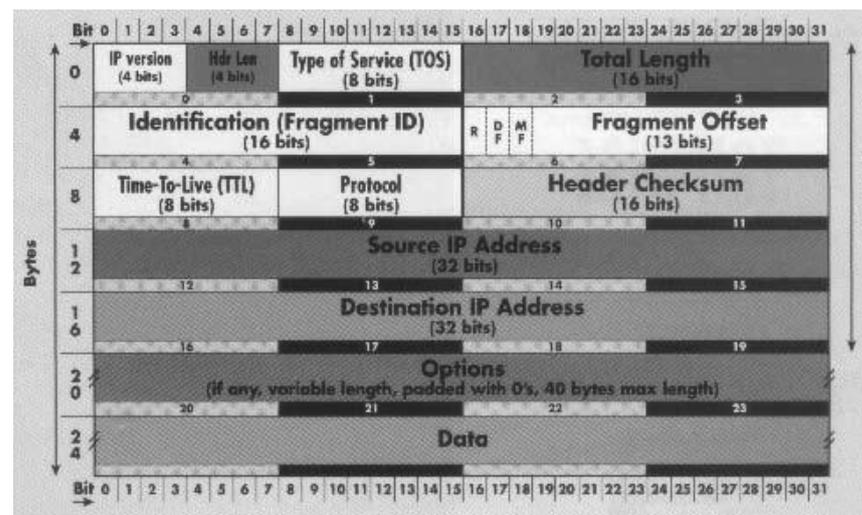
Layer berikutnya, yaitu *Transport Layer*, berisi protokol yang bertanggung jawab untuk mengadakan komunikasi antara dua komputer. Kedua protokol terse-

but ialah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*). Layer teratas, ialah *Application Layer*. Pada layer inilah terletak semua aplikasi yang menggunakan protokol TCP/IP.

### 2.3. IP (Internet Protocol) Header

Header Internet Protocol (IP) panjangnya 20 *byte*. Dalam 20 byte protokol *header* tersebut adalah beberapa informasi penting yang menjadi jantung proses routing dan pola penyampaian paket dari satu node ke node lain di jaringan Internet. Informasi penting yang digunakan untuk proses routing adalah:

1. *Source IP Address* (32 bit), alamat IP komputer pengirim data
2. *Destination IP Address* (32 bit), alamat IP komputer tujuan
3. *Time-To-Live* (8 bit), jumlah maksimum *router* yang dapat dilewati (maksimum 255 *router*)
4. Protokol (8 bit), memberitahukan protokol apa yang dibawa di atas IP. Contohnya adalah TCP dan User Datagram Protocol (UDP)



Gambar 2.3 Bagan Internet Protocol

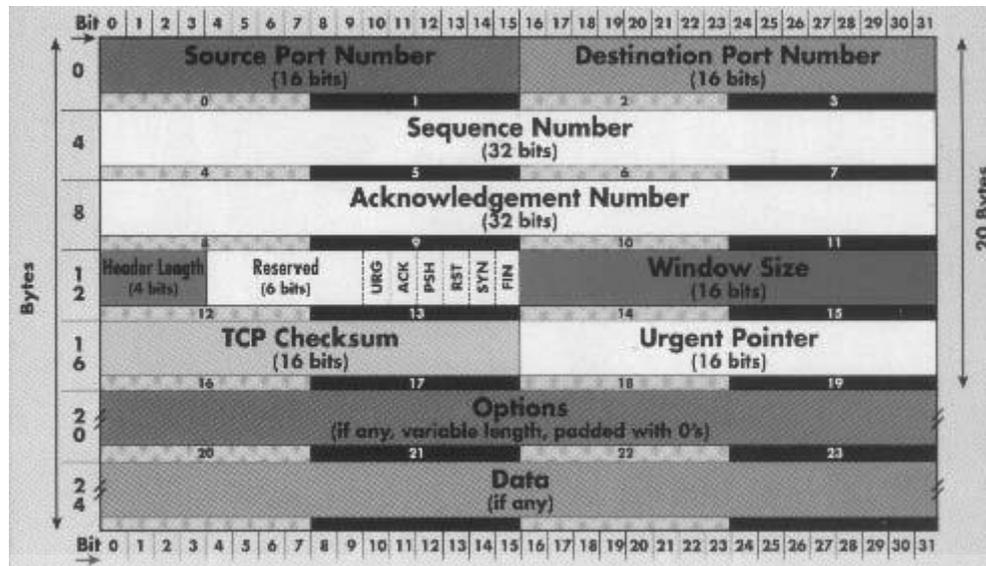
Untuk operasi tidak terlalu normal, misalnya kita ingin mengirimkan suara telepon melalui jaringan Internet secara *reliable* maka kita perlu menseset bit pada *Type of Service* (ToS) yang jumlahnya 8 bit. Dalam bahasa sederhana, misalnya bit ToS yang ada dalam IP yang kita terima adalah 00111100 mempunyai arti bahwa paket tersebut harus diprioritaskan dalam antrian di router, agar memperoleh *delay* seminimal mungkin, dengan *throughput* semaksimal mungkin dan reliabilitas yang tinggi.

Untuk paket yang normal isi *Type of Service* adalah 00000000. IP juga mengontrol apakah data yang kita kirim dapat di potong menjadi potongan yang lebih kecil, istilahnya difragmentasi. Kontrol apakah paket dapat difragmentasi, dan jika difragmentasi paket tersebut merupakan fragmentasi yang ke berapa akan dikontrol oleh Flag di protokol IP. Adapun isi tiga bit pertama dari Flag yang kemudian diikuti oleh informasi *offset* Fragmentasi dapat dilihat pada tabel 1. (Purbo, 2003:29-31)

#### **2.4. TCP (Transmission Control Protocol) Header**

*Transmission Control Protocol* (TCP) beroperasi di atas IP. Untuk operasi normal kita hanya perlu memperhatikan *Source Port* (16 bit), *Destination Port* (16 bit), *Sequence Number* (32 bit) yang merupakan nomor urut paket dan *Acknowledgement Number* (32 bit) merupakan nomor paket yang sudah di *acknowledge* atau sampai dengan selamat ke tujuan. Perhatikan bahwa *Acknowledgement Number* dan *Sequence Number* merupakan kunci utama dalam menjamin reliabilitas pengiriman data menggunakan protokol TCP. Jika ada paket

yang belum diterima atau belum sampai akan dapat dilihat dengan mudah melalui nomor tersebut.



Gambar 2.4 Bagan TCP

Nomor *port* (16 bit) adalah kode *port* untuk menentukan aplikasi apa yang sedang kita gunakan. Berbeda dengan IP yang tidak mengenal kondisi sambungan (*stateless*), pada TCP dikenal *state* (kondisi) sambungan. Yang di maksud kondisi sambungan (*state*) disini, misalnya klien sedang berusaha menghubungi server, server telah bersedia menerima hubungan komunikasi dengan klien, server dan klien bertransaksi data, server memutuskan hubungan, klien memutuskan hubungan komunikasi. Kondisi (*state*) tersebut di beberapa kontrol bit seperti pada tabel di bawah ini:

Tabel 2.1 Kondisi Pada Beberapa Kontrol Bit

| Nama        | Keterangan                                      |
|-------------|---|
| URG (1 bit) | paket yang sifatnya <i>urgent</i> / penting     |
| ACK (1 bit) | <i>acknowledge</i> , paket diterima dengan baik |
| PSH (1 bit) | <i>push</i> , memaksa                           |
| RST (1 bit) | <i>reset</i> , mereset hubungan                 |
| SYN (1 bit) | sinkronisasi, untuk membuka hubungan awal       |
| FIN (1 bit) | <i>final</i> , memutuskan hubungan              |

Beberapa contoh skenario dalam pemakaian bit kontrol adalah sebagai berikut,

1. Klien mengirimkan SYN ke Server, menandakan klien meminta hubungan dengan server
2. Server menjawab dengan mengirimkan ACK SYN, artinya Server siap menerima hubungan
3. Server dapat juga menjawab dengan mengirimkan PSH RST FIN, artinya server memutuskan hubungan dengan klien dan tidak bersedia menerima hubungan

4. Jika hubungan telah terjalin, biasanya hanya digunakan bit kontrol ACK atau PSH ACK untuk memberikan *acknowledgement* bahwa paket nomor tertentu telah diterima dengan baik

TCP *Checksum* (16 bit) digunakan untuk kode cek apakah paket yang dikirim masih utuh sampai di tujuan atau ada kerusakan. Jika masih utuh maka paket ACK akan dikirimkan oleh penerima ke pengirim. *Window Size* (16 bit) merupakan usaha untuk mengefisienkan penggunaan jaringan dengan cara mengirimkan beberapa paket sekaligus tanpa menunggu ACK terlebih dulu. Untuk kondisi yang tidak terlalu *reliable* biasanya TCP akan mengirimkan sebuah paket ke tujuan dan menunggu ACK dari tujuan sebelum mengirimkan paket selanjutnya. Konsekuensinya akan ada jeda (*delay*) yang lumayan (bisa beberapa ratus milidetik) antara satu paket dengan paket lainnya. Dengan konsep *sliding window* yang diset maksimum pada window size (16 bit), kita dapat menset *window size* beberapa kali panjang Maximum Transmission Unit (MTU).

Misalnya MTU 1500 byte, kita dapat menset *window size* 6000 byte. Artinya awalah jika pada saat pengiriman data TCP melihat bahwa jaringan cukup *reliable*, artinya tidak ada paket yang hilang / rusak di jalan dan ACK selalu dikirimkan oleh penerima dengan tepat waktu maka TCP akan berusaha mengirimkan beberapa (2-4) paket sekaligus tergantung panjang *window size* tanpa menunggu ACK dari penerima terlebih dulu.

Tentunya proses menaikkan jumlah paket yang dikirim sekaligus dilakukan secara bertahap, jadi berawal dari satu (1) paket jika ternyata baik maka akan dicoba dengan dua (2) paket, jika sambungan masih baik di coba lagi dengan tiga (3) paket dst. Jika terjadi kerusakan pada paket, artinya ACK tidak diperoleh pada waktunya, maka TCP langsung menurunkan lagi paket menjadi satu (1) paket lagi, dan secara bertahap dinaikkan lagi jika sambungan di rasakan *reliable*. Konsep naik turunnya jumlah paket yang dikirimkan secara otomatis ini dinamakan *sliding window*. Tentunya ada beberapa mekanisme lain yang tidak ada pada *header* protokol TCP tapi juga penting untuk menjamin reliabilitas pengiriman data yaitu, konsep TCP *back off* yang menentukan kapan pengiriman ulang sebuah paket yang rusak harus dilakukan. Berbeda dengan Internet Protocol (IP) yang sama sekali tidak menjamin reliabilitas pengiriman data. TCP berusaha secara maksimal agar proses pengiriman data andal dan efisien. (Purbo, 2003:29-31)

## 2.5. UML (Unified Modelling Language)

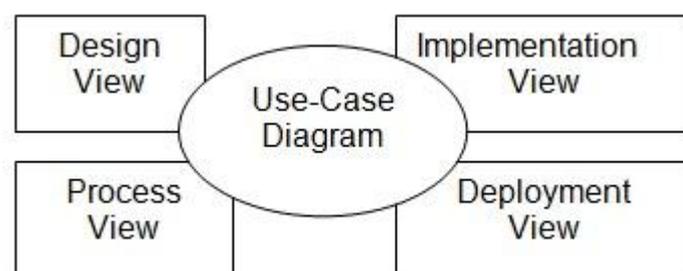
UML (*Unified Modelling Language*) adalah sebuah alat Bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini disebabkan

karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Usulan UML diberikan ke OMG (*Object Manajement Group*-konsorsium standarisasi teknologi objek) agar UML dijadikan bahasa dan notasi pemodelan dilakukan pada 1997. OMG menerima UML, UML telah menjadi standar *de-facto* karena pencipta-penciptanya sangat populer. Banyak pengembangan perangkat lunak yang mengadopsi UML. OMG adalah konsorsium yang beranggotakan lebih dari 850 perusahaan untuk mendefinisikan standar-standar teknologi objek termasuk CORBA (*Common Object Request Broker Arcitechture*)

UML adalah bahasa untuk memvisualisasi, menspesifikasikan, dan mendokumentasi artifak-artifak sistem perangkat lunak. UML merupakan sistem notasi yang membantu pemodelan sistem menggunakan konsep berorientasi objek.

UML dibangun atas model 4+1 *view*. Model ini didasarkan pada fakta struktur sebuah sistem dideskripsikan dalam 5 *view*, dimana salah satunya adalah *Use Case View*.



Gambar 2.5 Model 4+1 *view*

Dalam proyek pengembangan sistem apapun, fokus utama dalam analisis dan perancangan adalah model. Hal ini berlaku umum tidak hanya untuk perangkat lunak. Dengan model kita dapat merepresentasikan sesuatu karena :

1. Model mudah dan cepat untuk dibuat
2. Model bisa digunakan sebagai simulasi untuk mempelajari lebih detail tentang sesuatu
3. Model bisa dikembangkan sejalan dengan pemahaman kita tentang sesuatu.
4. Model bisa mewakili sesuatu yang nyata maupun tidak nyata.

Konsep-konsep yang diterapkan di UML adalah satu model berisikan informasi mengenai sistem (atau domain), model-model berisi elemen-elemen model seperti kelas, simpul-simpul, paket-paket, dan sebagainya. Satu diagram menunjukkan satu pandangan tertentu dari model (Dharwiyanti, 2003:2).

## 2.6. Sniffer

Sniffer merupakan salah satu teknik yang digunakan untuk mengambil dan menganalisa data di suatu jaringan, sehingga sering digunakan untuk memeriksa dan melakukan tes *network security*. Tersedia banyak *tools* sniffer, sebagai contoh salah satunya yaitu network monitor dari Microsoft atau produk *intrusion detection* yang ada saat ini atau berupa sniffer lainnya yang tersedia di *Internet* berupa *freeware*. Produk sniffer itu bekerja dalam *promiscuous mode* yaitu di mana dia akan mengambil dan menganalisa semua paket yang lewat melalui suatu *network interface* secara *real time*. (Harry, 2005)

## 2.7. Bandwidth

Seperti telah kita tahu, *bandwidth* paling banyak digunakan sebagai ukuran kecepatan aliran data. Tetapi apakah itu bandwidth sebenarnya? Bandwidth adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam suatu waktu tertentu. Bandwidth dapat dipakaikan untuk mengukur baik aliran data analog mau pun aliran data digital. Sekarang telah menjadi umum jika kata bandwidth lebih banyak dipakaikan untuk mengukur aliran data digital. Satuan yang dipakai untuk bandwidth adalah bits per second atau sering disingkat sebagai bps. Seperti kita tahu bahwa bit atau binary digit adalah basis angka yang terdiri dari angka 0 dan 1. Satuan ini menggambarkan seberapa banyak bit (angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat yang lain dalam setiap detiknya melalui suatu media.

Bandwidth adalah konsep pengukuran yang sangat penting dalam jaringan, tetapi konsep ini memiliki kekurangan atau batasan, tidak peduli bagaimana cara Anda mengirimkan informasi mau pun media apa yang dipakai dalam penghantaran informasi. Hal ini karena adanya hukum fisika mau pun batasan teknologi. Ini akan menyebabkan batasan terhadap panjang media yang dipakai, kecepatan maksimal yang dapat dipakai, mau pun perlakuan khusus terhadap media yang dipakai.

Sedangkan batasan terhadap perlakuan atau cara pengiriman data misalnya adalah dengan pengiriman secara paralel (synchronous), serial (asynchronous), perlakuan terhadap media yang spesifik seperti media yang tidak boleh ditekuk (serat optis), pengirim dan penerima harus berhadapan langsung (line of sight), kompresi data yang dikirim, dll. (Dewo, 2003:1)

## 2.8. Throughput

Ternyata konsep bandwidth tidak cukup untuk menjelaskan kecepatan jaringan dan apa yang terjadi di jaringan. Untuk itulah konsep *Throughput* muncul. Throughput adalah bandwidth aktual yang terukur pada suatu ukuran waktu tertentu dalam suatu hari menggunakan rute internet yang spesifik ketika sedang mendownload suatu file. Bagaimana cara mengukur bandwidth? Dan bagaimana hubungannya dengan throughput? Seperti telah diulas di atas, bandwidth adalah jumlah bit yang dapat dikirimkan dalam satu detik. Rumus dari bandwidth adalah total bandwidth dibagi waktu perdetik

Sedangkan throughput walau pun memiliki satuan dan rumus yang sama dengan bandwidth, tetapi throughput lebih pada menggambarkan bandwidth yang sebenarnya (aktual) pada suatu waktu tertentu dan pada kondisi dan jaringan internet tertentu yang digunakan untuk men-download suatu file dengan ukuran tertentu. Rumus perbandingan throughput dengan bandwidth untuk waktu download terbaik adalah ukuran file dibagi dengan bandwidth, sedangkan waktu download tipikal adalah ukuran file dibagi throughput.

Dengan hanya mempergunakan bandwidth sebagai patokan, seharusnya file yang akan didownloadnya yang berukuran 64 kb seharusnya bisa didownload dalam waktu sekedip mata atau satu detik, tetapi setelah diukur ternyata memerlukan waktu 4 detik. Jadi jika ukuran file yang didownload adalah 64 kb, sedangkan waktu downloadnya adalah 4 detik, maka bandwidth yang sebenarnya atau bisa kita sebut sebagai throughput adalah  $64 \text{ kb} / 4 \text{ detik} = 16 \text{ kbps}$ . (Dewo, 2003:2)

## 2.9. Permainan Komputer

Permainan Komputer dimainkan lewat bantuan alat komputer. Terdapat 5 alat yang dapat dikategorikan sebagai komputer, yaitu :

1. *Expensive dedicated machine*, mesin yang dioperasikan dengan koin untuk memainkannya.
2. *Inexpensive dedicated machine*, disebut juga dengan *hand held machine*. Alat *game watch* termasuk dalam katagori ini.
3. *Multiprogram home*, mesin seperti Atari, Nintendo termasuk dalam kelompok komputer ini.
4. *Personal computer*
5. *Mainframe computer*

*Computer game* berbeda dengan jenis game yang lain karena tidak ada pergerakan secara fisik atau interaksi langsung dengan object kecuali lewat perantaraan komputer. *Software* yang dibuat harus dapat menangkap reaksi yang cepat dari interaksi yang dihasilkan dengan pemain. Karena itu *software* untuk *computer games* harus bersifat *real time*. Kompleksitas *game* adalah bergantung dari kemampuan merepresentasikan aturan dan lingkungan *game* dalam program yang dibuat.

Diantara sekian banyak definisi *game*, maka definisi yang umum untuk *computer game* adalah :

*A computer game is a software program in which one or more players make decisions through the control of game objects and resources, in pursuit of a goal.*

Game komputer adalah sebuah program *software* dimana satu atau lebih pemain berusaha untuk membuat keputusan lewat kontrol terhadap object dan *resource* guna memenuhi satu tujuan tertentu.(Prayudi : 2008)

### **2.10. Real Time Strategy (RTS)**

RTS adalah permainan yang meliputi strategi dengan unsur-unsur bangunan dasar dan manajemen sumber daya. Memang benar bahwa sebagian besar permainan dalam sesuai genre dalam kategori tradisional tetapi ada permainan pasti telah diterima sebagai bagian genre yang hanya memiliki satu atau tidak persyaratan itu. Pertanyaannya adalah di mana garis mendapatkan ditarik. Apakah ingin dimasukkan sub-genre seperti permainan taktis real-time atau apakah tetap dengan definisi tradisional.

kecenderungannya adalah bersandar kepada Blizzard Rob Pardo yang menjelaskan sebuah RTS sebagai "Sebuah permainan strategis dimana modus utama bermain adalah dalam pengaturan real-time." Tentu saja bahwa definisi membuka pintu terlalu banyak dan akan mencakup permainan seperti Firaun dan RollerCoaster Tyoon. Jadi untuk kepentingan artikel ini, akan menyertakan kata "militer" dalam definisi. Jadi definisi tentang permainan strategi real-time, untuk yang lebih baik atau lebih buruk, adalah "sebuah permainan strategi militer dimana modus utama bermain adalah dalam pengaturan real-time."

Anehnya, ini berarti bahwa permainan seperti Roma: Perang Total sebenarnya dikecualikan dari definisi ini sebagai argumen bahwa mereka hebat secara teknis strategi permainan dengan komponen real-time. Sistem turn-based yang sangat kuat bisa dipandang sebagai banyak mendominasi pengalaman

pemain tunggal. Namun, mereka seperti pada Perang Empire, Rise of Nations, dan Battle for Middle-earth masih RTSs untuk penekanan mereka pada real-time pertempuran dengan turn-based sebagai komponen tambahan. Tentu saja, jika mencari daerah abu-abu lebih, melihat ke judul seperti di seri Europa Universalis.

Dalam definisi umum adalah beberapa sub-genre. Seperti disebutkan sebelumnya, game RTS tradisional memiliki komponen bangunan dasar, pengelolaan sumber daya, dan pohon teknologi. Game seperti Command & Conquer, Starcraft, dan Age of Empires masuk ke dalam kategori ini. Real-time taktis (RTT) permainan di sisi lain bahkan tidak selalu membutuhkan komponen sumber daya dan umumnya tantangan pemain akan menang dengan jumlah tentara set untuk setiap tingkat. Judul seperti Mitos atau Blitzkrieg membuat rumah mereka di sini dan kami akan menyelidiki secara lebih mendalam yang Agak terlambat dalam artikel.(Adams, 2006)

### **2.11. Performance Indicator**

Menurut Renu Rajani (2005:164), Untuk menjamin stabilitas server website dan memberikan ukuran yang baik untuk pengujian kinerja, perlu untuk mendefinisikan satu set indikator. indikator ini akan memberikan perilaku kinerja server website. Satu dapat kategori indikator dari klien dan sisi server. Meskipun indikator ini memberikan informasi dasar tentang kinerja, mereka tidak memberikan gambaran yang sebenarnya dari lingkungan nyata. pemilihan parameter kinerja murni tergantung pada jenis situs web dan persyaratan. Namun, beberapa parameter kinerja yang umum untuk websiter semua. Sebuah indikator kinerja sedikit yang defined di bawah ini:

### 2.11.1 Processor

Ini merupakan sumber penting, yang mempengaruhi kinerja aplikasi. Penting untuk mengukur jumlah waktu yang dihabiskan dalam memproses pekerjaan oleh satu atau banyak CPU. Jika indikator tersebut secara konsisten di atas 90% pada satu atau lebih prosesor saja menunjukkan bahwa tes ini terlalu kuat untuk hardware. indikator ini berlaku hanya ketika sistem memiliki beberapa prosesor.

### 2.11.2 Disk Fisik

Sejak *disk* adalah perangkat lambat dengan kapasitas lebih untuk menyimpan, parameter seperti *latency* dan waktu berlalu memberikan informasi lebih lanjut tentang meningkatkan kinerja. Kinerja sistem juga tergantung pada panjang antrian *disk* yang menunjukkan jumlah permintaan yang beredar di dalam *disk*. Diterima antrian panjang menunjukkan *disk* atau masalah memori.

### 2.11.3 Memory

Memori dalam sistem komputer merupakan bagian integral dari sistem perangkat keras. kecepatan memori lebih sampai I / O selama proses eksekusi tetapi beban pada parameter biaya. Semua data dan instruksi yang akan dieksekusi harus tersedia pada memori terdiri dari halaman. Jika halaman lain telah tersedia, eksekusi lebih cepat. Melihat jumlah kesalahan halaman memberikan informasi yang membantu menentukan persyaratan memori. Perlu mengingat berapa banyak halaman yang pindah ke dan dari disk untuk memenuhi persyaratan memori virtual.

#### 2.11.4 Lalu Lintas Jaringan

Hal ini tidak mungkin untuk secara langsung menganalisa lalu lintas jaringan di internet yang tergantung pada bandwidth, jenis koneksi jaringan dan biaya lainnya. Namun, dimungkinkan untuk mengetahui waktu yang dibutuhkan untuk mencapai jumlah byte klien dari server. Masalah bandwidth juga akan mempengaruhi kinerja, sebagai jenis sambungan bervariasi dari klien ke klien. Indikator kinerja menyediakan dasar untuk pengujian, dan dalam setiap kategori jenis counter dapat diatur. Nilai yang diberikan oleh counter greatly membantu dalam menganalisis kinerja. Untuk mengatur penghitung kinerja, lingkungan yang tepat harus established untuk pengujian.

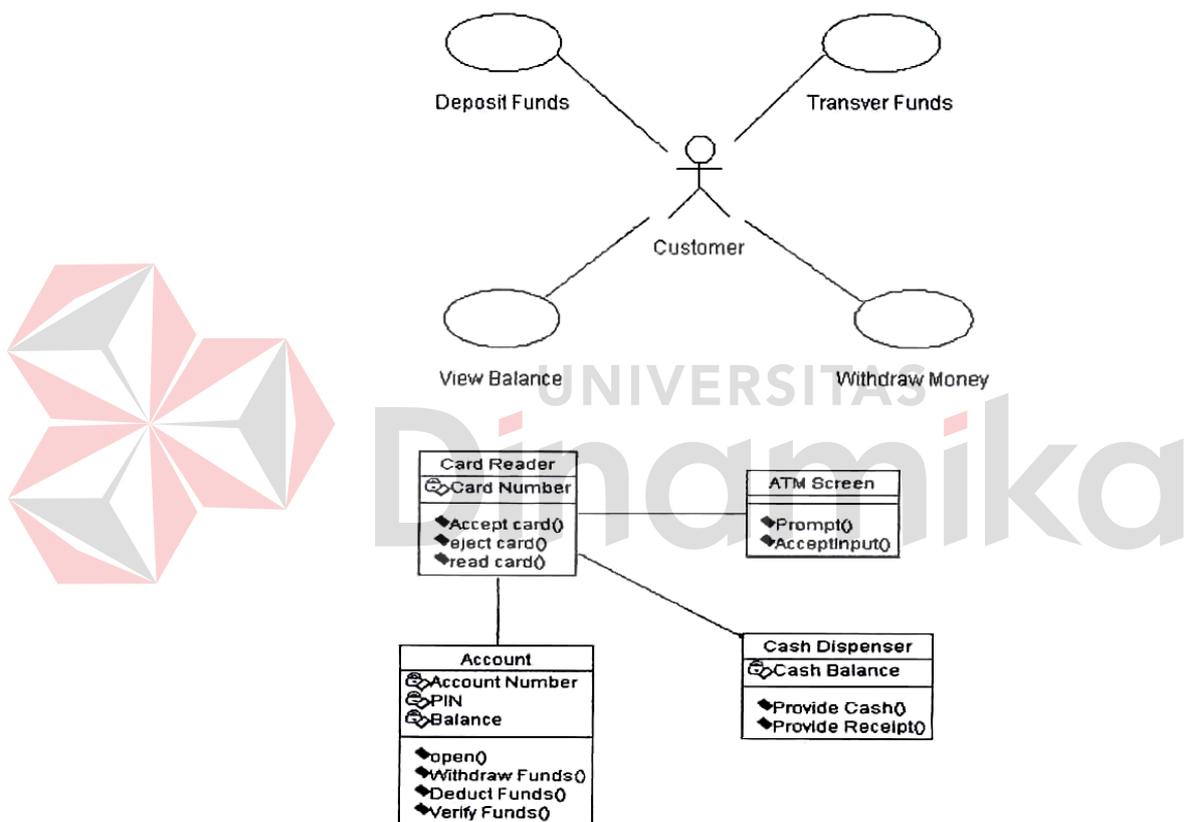
#### 2.12. Permodelan Visual

Menurut Sholiq dalam buku Analisa & Perancangan Berorientasi Objek, model adalah abstraksi yang menggambarkan hal-hal signifikan dari persoalan yang kompleks atau rancangan dengan membuang hal-hal yang tidak diperlukan., sehingga. membuat persoalan lebih mudah dipahami. Abstraksi adalah kemampuan manusia yang memungkinkan berurusan dengan hal-hal yang kompleks Misalkan ahli teknik seniman dan pengarang membuat model untuk masalah-masalah yang dihadapi sebelum benar-benar membangunnya.

Untuk membangun sistem yang kompleks, pengembang harus menampilkan beberapa pandangan yang berbeda dan sistem, membangun model menggunakan notasi-notasi yang tepat, melakukan verifikasi bahwa model yang dibuatnya memenuhi persyaratan-persyaratan sistem, dan menambahkan detail secara berangsur-angsur untuk mentransformasikan model menjadi implementasi.

### 2.13. Unified Modelling Language (UML)

Menurut Sholiq, *Unified Modelling Language* (UML) adalah salah satu jenis pemodelan berorientasi objek yang menyediakan notasi yang cukup tangguh untuk membangun sistem dari tahap analisa sampai ke tahap perancangan. Selain UML terdapat pula pemodelan berorientasi objek yang lain, yaitu Notasi Booch dan *Object Management Technology* (OMT).



Gambar 2.6 Contoh notasi UML

#### 2.14. Pengenalan diagram-diagram dalam UML

UML: membantu memahami sistem dengan menggunakan beberapa diagram visual yang menunjukkan bermacam aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain:

1. Diagram use case bisnis (business use case diagram)
2. Diagram use case (use case diagram)
3. Diagram aktivitas (activity diagram)
4. Diagram sekuensial (sequence diagram )
5. Diagram kolaborasi (collaboration diagram)
6. Diagram kelas (class diagram)
7. Diagram keadaan (statechart diagram)
8. Diagram komponen (component diagram)
9. Diagram penyebaran (deployment diagram)

#### 2.15. Network Monitor Untuk Mengawasi Trafik Jaringan

Menurut Don Jones (2004:82), monitor jaringan (NetMon.exe) untuk mengawasi trafik jaringan memungkinkan untuk menganalisis secara mendalam, lalu lintas level terbawah jaringan dan memungkinkan untuk mendeteksi dan menganalisa masalah pada jaringan lokal dan koneksi WAN. Monitor Jaringan menangkap dan menampilkan paket jaringan yang kontroler domain Win2k menerima dari pengguna dan server lainnya untuk menyediakan real-time monitoring lalu lintas dan analisis. Dapat juga menampilkan lalu lintas di mode setelah *capture* untuk membantu menganalisis setelah data tersedia (fakta). Dalam mode *real-time*, monitor jaringan memungkinkan untuk memonitor dan menguji

lalu lintas jaringan untuk satu kondisi tertentu. Jika kondisi tertentu, akan menampilkan suatu kejadian dan meminta respon user untuk tindakan yang tepat. Dalam analisis *post-capture*, lalu lintas jaringan disimpan dalam *file* dan bisa diurai oleh protokol untuk memilih jenis jaringan tertentu.

## 2.16. Monitor Network Trafik

Menurut Mark Foust, James Chellis, Matthew Sheltz, Suzan Sage London (2003:683), monitor Jaringan adalah contoh dari jenis program yang disebut "network analyzer" (atau *sniffer*, dinamai *toolset General Network Sniffer*). Analisis Jaringan *capture* trafik paket data dari jaringan dan kemudian *decode* hanya sebagai protokol *stack*. Karena analisis jaringan tidak tergantung pada protokol *stack*, dapat menggunakan alat analisis untuk memonitor lalu lintas untuk jenis protokol yang tidak diinstal. Misalnya, menggunakan *network monitor* untuk menangkap dan memecahkan kode paket AppleTalk sementara tips masalah konektivitas Mac, bahkan tanpa AppleTalk pada *workstation*. Monitor Jaringan memiliki dua komponen :

1. Aplikasi yang Anda instal pada Windows
2. Driver, yang Anda instal pada setiap Windows

## 2.17. Sistem Monitor

Menurut Glen E. Clarke (2009:707), ketika memantau performa sistem, beberapa hal menentukan mengapa sistem berjalan lambat. Ada empat area potensi kemacetan pada sebuah sistem :

1. Prosesor mungkin terlalu banyak bekerja jika terlalu banyak proses yang berjalan pada saat yang sama, atau menjalankan aplikasi prosesor-intensif.

Jika menentukan bahwa prosesor sedang digunakan secara berlebihan (biasanya di atas 85 persen pemanfaatan), mungkin perlu upgrade prosesor, tambahkan prosesor lain, atau mengaktifkan processor Hyperthreading (jika prosesor Anda mendukung Hyperthreading).

2. Memori mungkin memperhatikan bahwa sistem semakin lesu karena tidak memiliki cukup memori. Jika tidak memiliki cukup memori, sistem akan melakukan banyak paging-yaitu, bertukar informasi dari memori ke disk disk dan kemudian ke memori, ini menempatkan banyak beban kerja pada sistem. Untuk mengurangi paging, selalu menambah RAM lebih banyak.
3. Harddisk adalah bidang lain hambatan potensial. Karena harddisk melayani semua permintaan file di server, pastikan bahwa memiliki disk disk cepat atau ganda untuk layanan permintaan.
4. Jaringan Kartu jaringan merupakan hambatan potensial, terutama pada server jaringan, karena jaringan semua menjawab permintaan untuk sumber daya jaringan. Pastikan memiliki kartu jaringan tercepat mungkin.

## BAB III

### ANALISA DAN PERANCANGAN SISTEM

#### 3.1. Analisa Permasalahan

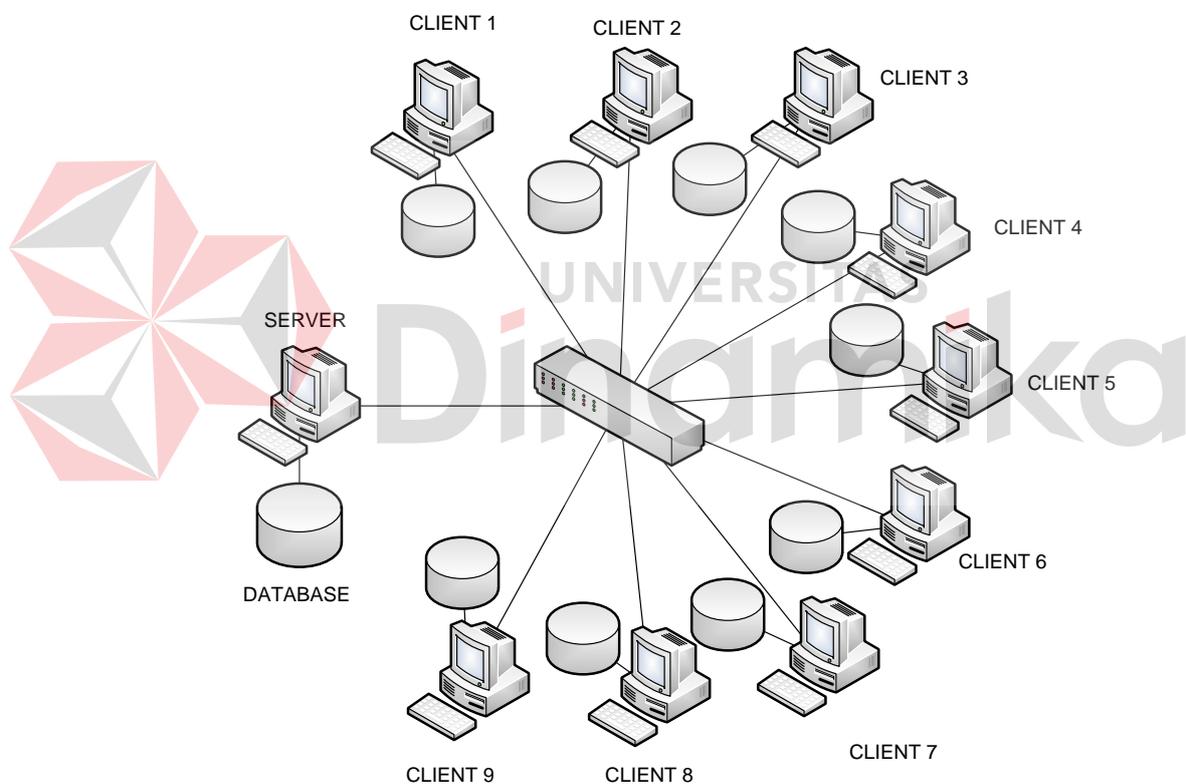
Permasalahan utama adalah jika permainan Warcraft III dimainkan melalui internet dengan jumlah pemain yang banyak, sehingga muncul masalah *Lag* atau *spike* yang sering terjadi pada jaringan internet. *Latency* yang besar bisa berakibat putusnya permainan di tengah-tengah permainan. Masalah *lag* atau *delay* ini sangat merugikan pemain dikarenakan aturan dari admin *server* yang memberikan sanksi untuk pemain yang meninggalkan permainan secara sengaja atau tidak sengaja yang umumnya dikarenakan meninggalkan permainan saat koneksi putus. Secara garis besar pemain tidak dapat menentukan koneksi internet yang cukup untuk memainkan permainan ini. Kebutuhan untuk *resource hardware server* juga penting karena *usage memory* dan *processor* mempengaruhi performa dari *server*. Masalah utama performa *server* dari permainan Warcraft III yaitu permainan ini menggunakan *resource* yang besar, sehingga pemain kesulitan untuk mengetahui kebutuhan *resource hardware* yang digunakan untuk *server*.

Dengan masalah di atas, maka dibuatlah sebuah program *client – server* untuk memonitoring dan menghitung besarnya *bandwidth* yang dibutuhkan oleh permainan Warcraft III sehingga pemain dapat mengkalkulasikan dan memperkirakan *bandwidth* yang dibutuhkan untuk dimainkan di *internet*. Program ini nantinya juga dapat memperkirakan performa *hardware* yang digunakan untuk *server* permainan.

### 3.2. Perancangan Sistem

Setelah dilakukan analisa terhadap sistem, langkah berikutnya perancangan sistem. Dimana dalam perancangan sistem ini dapat memberikan gambaran tentang sistem yang dibuat. Seluruh diagram UML yang digunakan untuk menggambarkan perancangan sistem, dibuat dengan menggunakan *tool* UModel.

### 3.3. Blok Diagram Aplikasi Monitoring

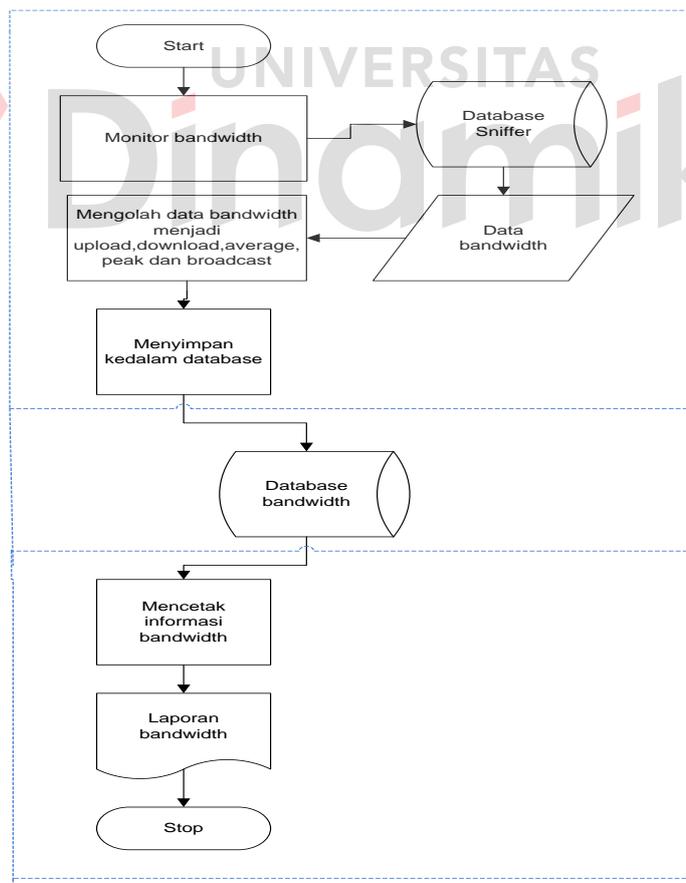
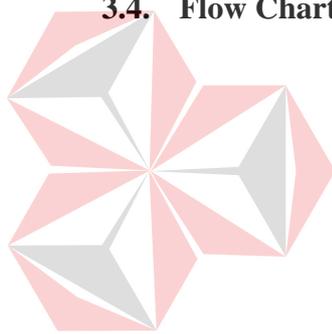


Gambar 3.1 Blok Diagram Aplikasi Monitoring

Gambar 3.1, desain umum sistem menggambarkan rancang bangun aplikasi monitoring ini. Terdapat 9 *client* dan 1 *server* yang terhubung dalam 1 *network* melalui *switch hub* atau *wireless access point*. *Server* membuat *room* permainan yang diikuti 9 *client* kemudian permainan dimulai. Saat permainan

dimulai *client* menjalankan aplikasi *sniffer* untuk menangkap paket *network* yang lewat, kemudian *database client dishare* untuk dikoleksi oleh aplikasi *server*. Bersamaan dengan proses *sniffer* pada *client*, *server* menjalankan aplikasi *server* untuk memonitor performa komputer yang terdiri dari 4 parameter yaitu CPU (*Central Processing Unit*), *physical memory*, Warcraft III *memory working set* dan *harddisk*, Kemudian data hasil monitoring juga disimpan kedalam *database bandwidth*. *Server* juga menjalankan aplikasi *sniffer* untuk melakukan *capture paket* yang lewat jaringan *server*, data monitor trafik kemudian disimpan kedalam *database bandwidth*.

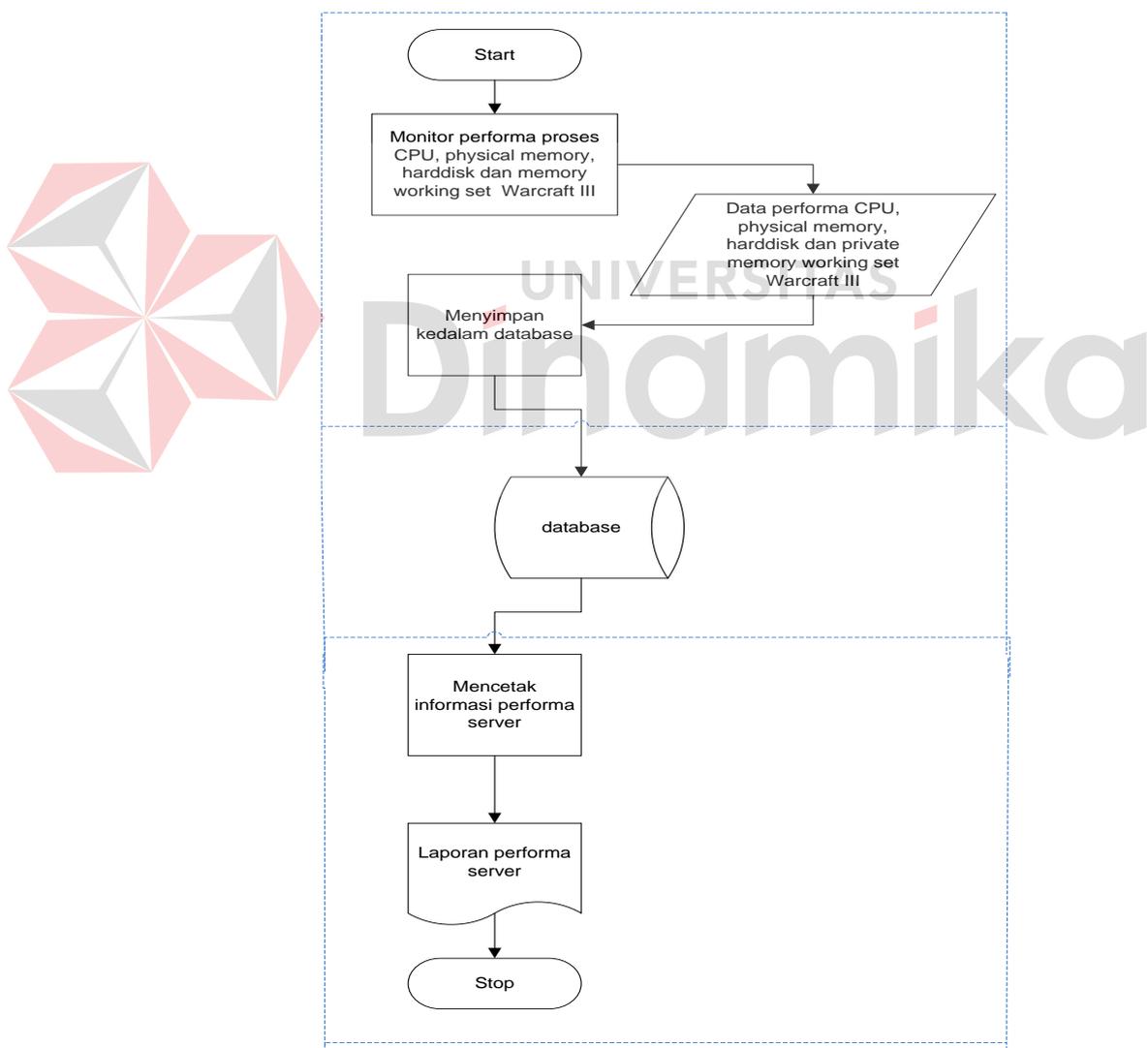
### 3.4. Flow Chart Monitoring Trafik



Gambar 3.2 Flow Chart Monitoring Trafik

Gambar 3.2 Desain umum monitoring trafik menggambarkan alur proses monitoring trafik yang bermula dari mengambil data di *database sniffer* milik *client* dan *server* kemudian diproses menjadi *upload*, *download*, *average*, *peak* dan *broadcast* yang selanjutnya disimpan kedalam *database bandwidth*. Setelah data tersimpan di dalam *database bandwidth* seterusnya akan diproses menjadi laporan *bandwidth*.

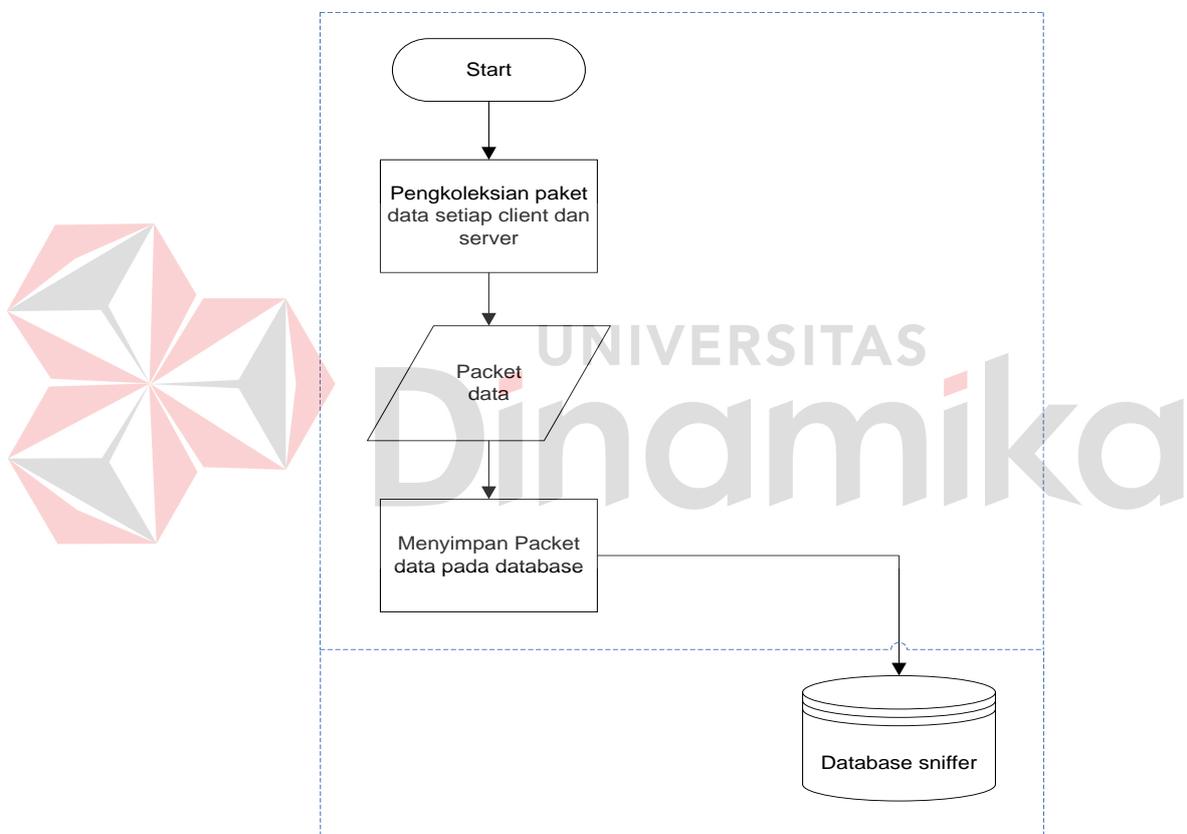
### 3.5. Flow Chart Monitor Performa



Gambar 3.3 Flow Chart Monitoring Performa

Gambar 3.3 Desain umum monitoring performa menggambarkan alur proses monitoring performa yang bermula dari melakukan *real time capturing* dengan perhitungan *running average* performa *server* saat permainan dijalankan, kemudian setelah *capturing* selesai data disimpan kedalam *database* performa yang selanjutnya akan diteruskan kedalam laporan performa.

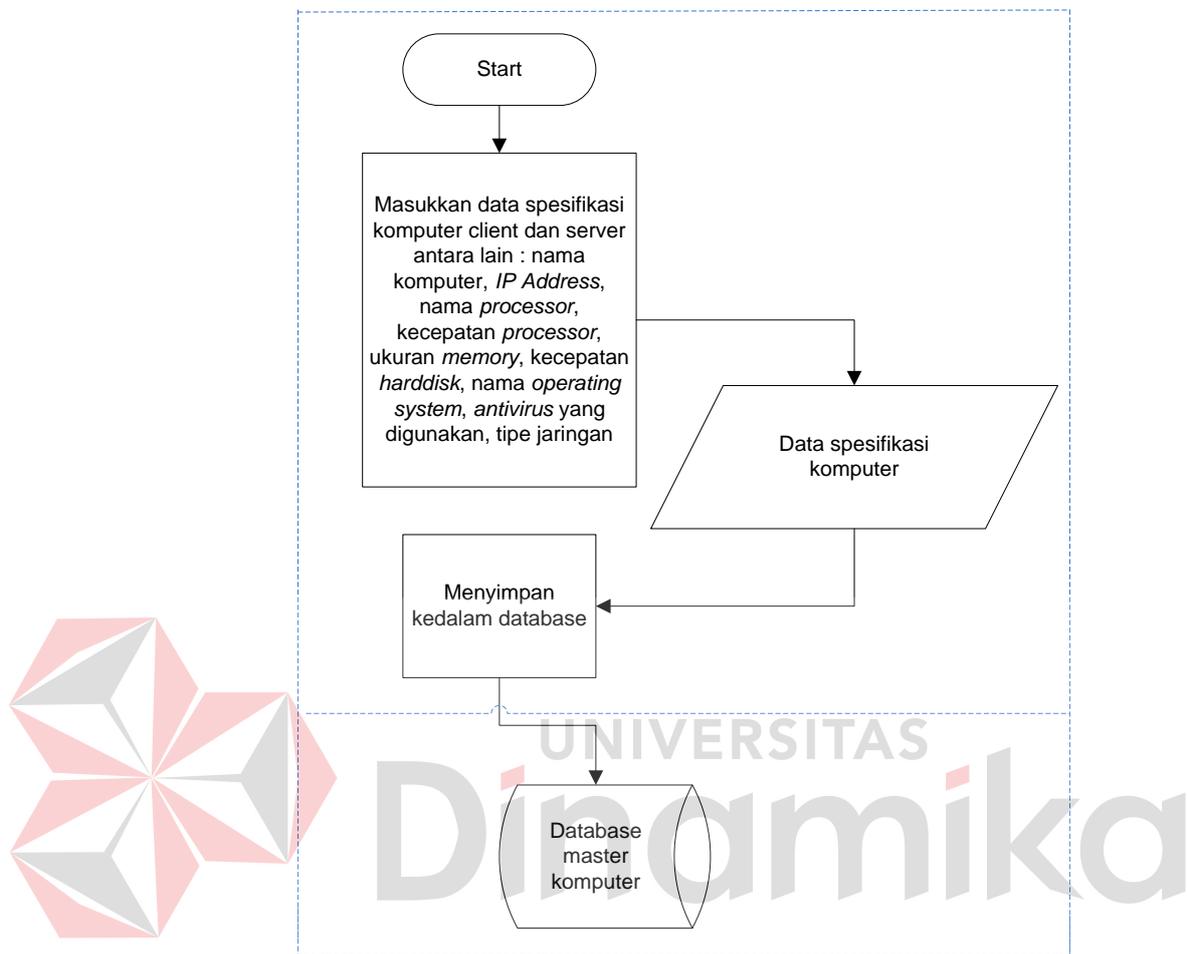
### 3.6. Flow Chart Sniffer



Gambar 3.4 Flow Chart Sniffer

Gambar 3.4 Desain umum *sniffing packet data* yang bermula dari melakukan *sniffing packet data* disetiap komputer kemudian data hasil *sniffing* disimpan kedalam *database sniffer* yang selanjutnya akan diteruskan kedalam proses pengolahan *bandwidth*.

### 3.7. Flow Chart Maintenance Data Master Komputer



Gambar 3.5 Flow Chart Maintenance Data Master Komputer

Gambar 3.5 Desain umum *sniffing packet data* yang bermula dari melakukan *sniffing packet data* di setiap komputer kemudian data hasil *sniffing* disimpan kedalam *database sniffer* yang selanjutnya akan diteruskan kedalam proses pengolahan *bandwidth*.

### 3.8. Use case Model

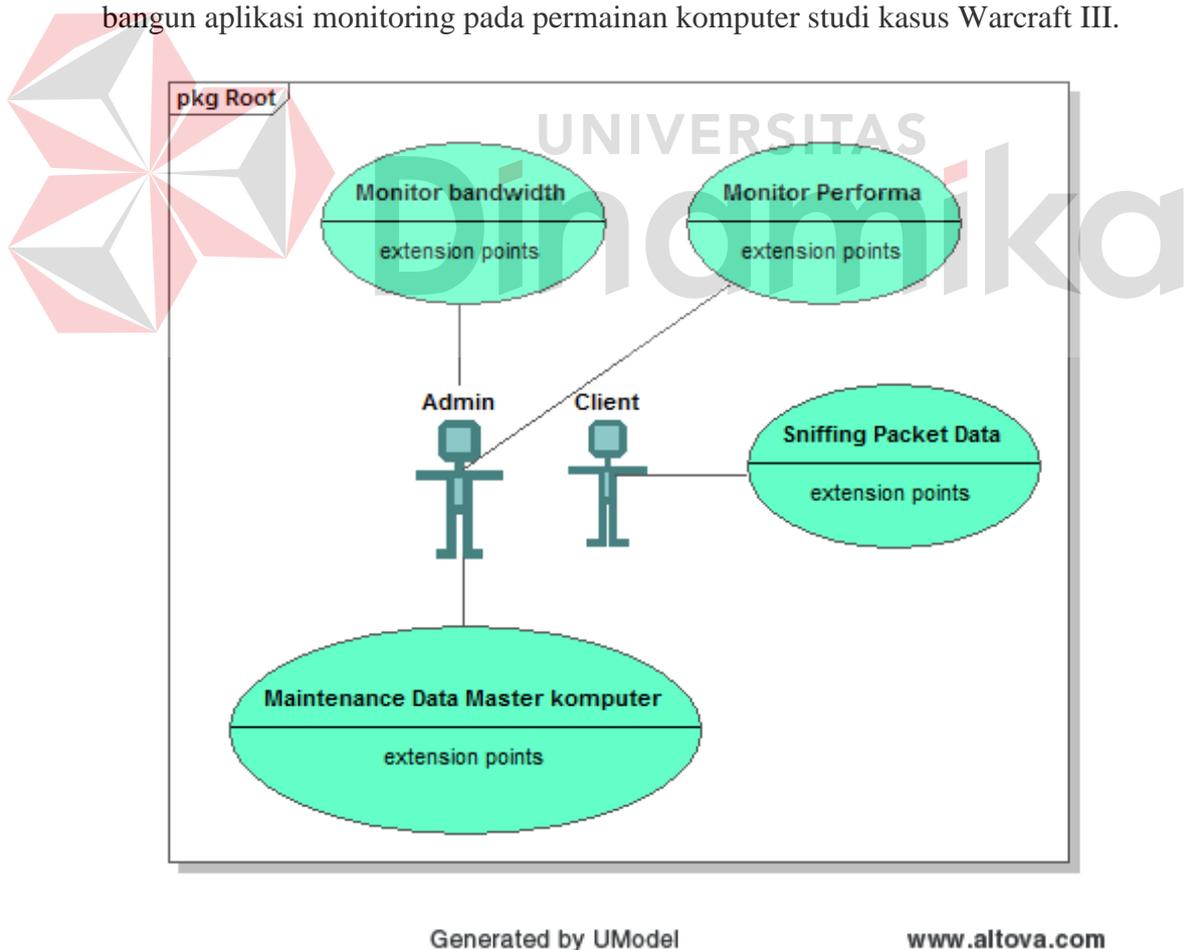
*Use case Model* merupakan model yang menggambarkan kebutuhan perangkat lunak dalam bentuk satu atau lebih *Use case*. *Use case model* dibuat

untuk mengidentifikasi suatu sistem yang penting secara arsitektural dari perangkat lunak yang akan dibuat. *Use case model* juga digunakan sebagai masukan penting selama proses analisa, desain dan testing.

### 3.8.1 Use case Diagram

*Use case Diagram* merupakan diagram yang menggambarkan secara grafik perilaku (*behavior*) *software* aplikasi serta hubungannya dengan aktor. Peranan terpenting dari suatu model *Use case* adalah untuk mengkomunikasikan suatu sistem dan perilakunya kepada pemakai akhir.

Berikut adalah gambar *Use case* diagram aplikasi monitoring pada rancang bangun aplikasi monitoring pada permainan komputer studi kasus Warcraft III.



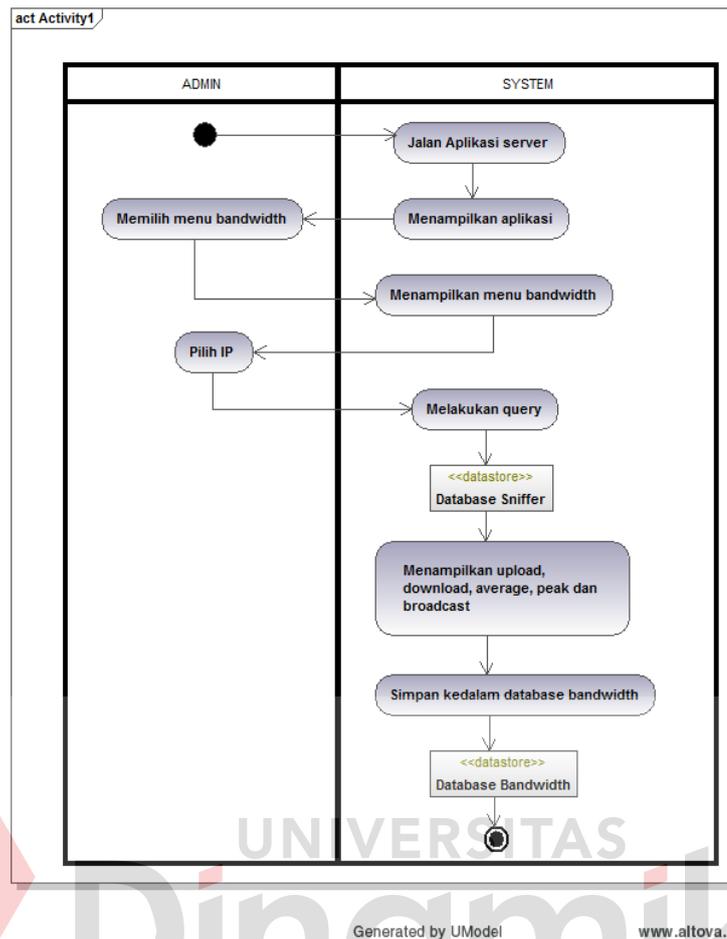
Gambar 3.6 Use Case Diagram Aplikasi Monitoring

Dari gambar 3.6 memperlihatkan sistem aplikasi yang akan diterapkan untuk sistem aplikasi monitoring trafik dan performa *server* pada permainan komputer (studi kasus warcraft III). Aktor yang terdapat pada sistem monitoring adalah *admin* dan *client*. Terdapat 4 *usecase* yaitu monitor *bandwidth*, monitor performa dan *maintenace* data master komputer yang dilakukan oleh *admin*, kemudian *sniffing packet data* yang dilakukan oleh *client* dan *server*.

Dalam *Use case Diagram* aplikasi monitoring trafik dan performa *server* terdapat beberapa *case* yaitu :

1. Monitor Bandwidth

*Use case* monitor *bandwidth* digunakan *admin* untuk memonitor trafik yang melewati jaringan antara *server* dan *client*. *Bandwidth* yang dimonitor meliputi *upload*, *download*, *average*, *peak* dan *broadcast*. *Upload bandwidth* untuk mengetahui *bandwidth* yang keluar dari setiap komputer, *download bandwidth* untuk mengetahui *bandwidth* yang masuk dari setiap komputer, *average bandwidth* untuk mengetahui *bandwidth* rata – rata yang keluar ataupun yang masuk dari setiap komputer, *peak bandwidth* untuk mengetahui *bandwidth* terbesar yang masuk ataupun yang keluar dari setiap komputer dan *broadcast bandwidth* untuk mengetahui *bandwidth* yang digunakan untuk *listener* dari setiap komputer. Data *bandwidth* yang diproses didapat melalui proses *sniffing* yang dilakukan *client* dan *server* saat permainan dilakukan. Setelah *sniffing* selesai dilakukan, *server* melakukan pengumpulan data. Berikut adalah *flow of event* untuk *Use case Start capture* :

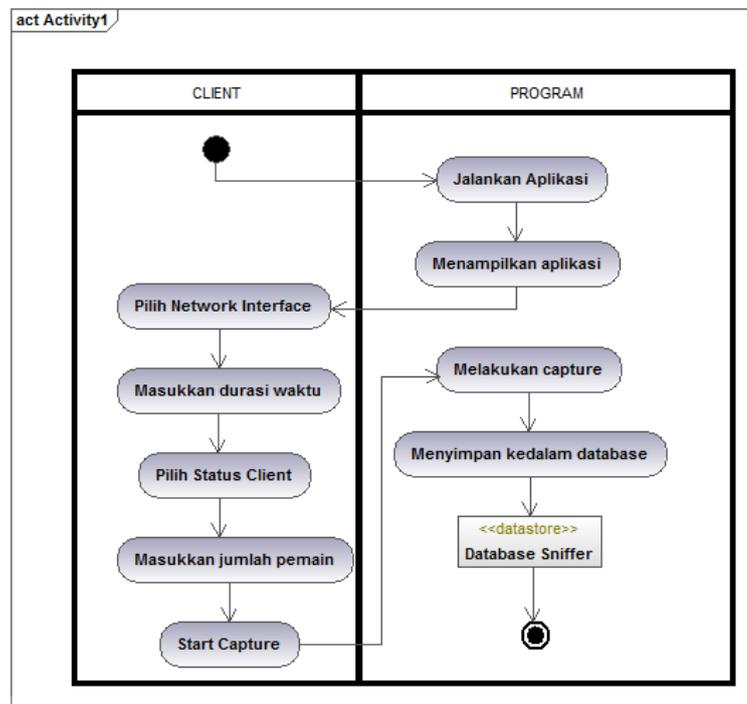


Gambar 3.7 Flow of Event Monitor Bandwidth

Proses monitor diawali dengan *admin* yang menjalankan aplikasi, setelah aplikasi dijalankan *admin* memilih *IP Address client*. Sistem akan melakukan proses *query* kedalam *database sniffer* dan nilainya akan dikembalikan menjadi data *bandwidth upload, download, average, peak dan broadcast*. Setelah data *sniffer* diproses menjadi data *bandwidth* seterusnya akan disimpan kedalam *database bandwidth*.

## 2. Use case Sniffing Packet Data

*Use case sniffing packet data* digunakan untuk melakukan *capturing* semua *packet data* yang melewati jaringan *client*. Berikut adalah gambar *flow of event sniffing packet data* :



Generated by UModel

www.altova.com

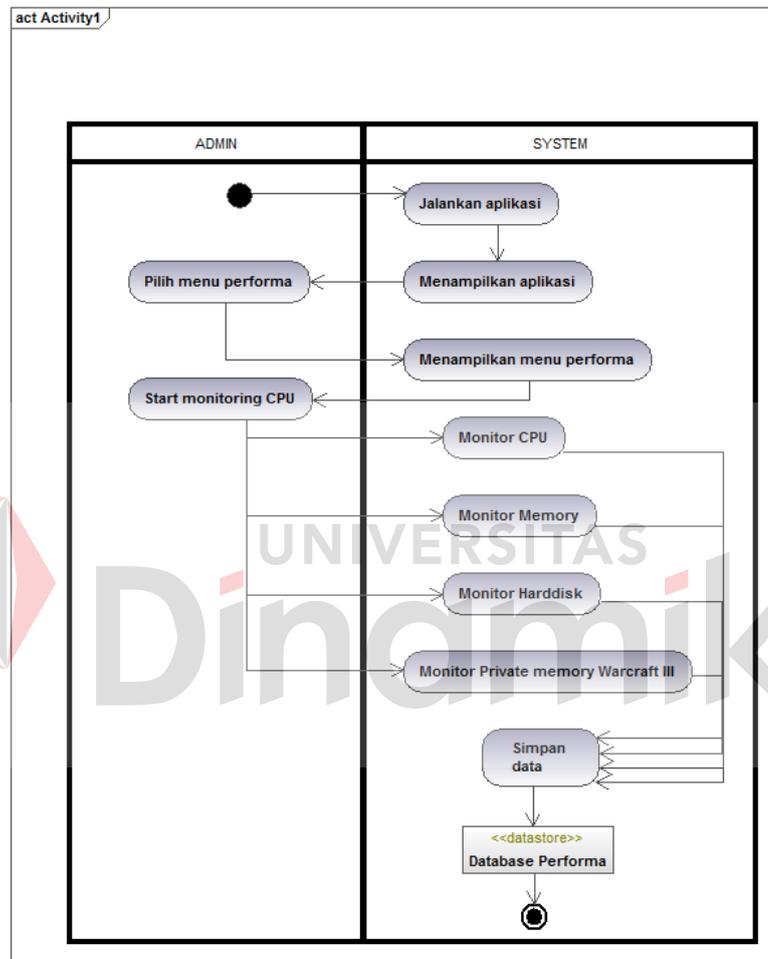
Gambar 3.8 Flow of Event Sniffing Packet Data

Proses *sniffing packet* data dimulai setelah *client* menjalankan aplikasi kemudian memilih *interface network* yang digunakan, pada umumnya *wireless*. Kemudian *client* memasukkan durasi waktu untuk lamanya bermain, setelah memasukkan durasi barulah *client* mengeksekusi *capturing* dan program memulai *sniffing*.

### 3. Use case Monitor Performa

*Use case* monitor performa digunakan *admin* untuk memonitor performa *server* yang terdiri dari CPU (*Central Processing Unit*), *physical memory*, *harddisk* dan Warcraft III *private memory working set*. Monitor CPU untuk melihat CPU *usage* saat permainan berlangsung. Monitor CPU yang dilakukan untuk melihat perubahan kenaikan dan penurunan CPU. Monitor *physical memory* untuk melihat besarnya *resource memory* saat permainan dijalankan. Monitor

*harddisk* untuk melihat performa dari *read* dan *write* tempat dimana permainan disimpan dan monitor Warcraft III *private memory working set* untuk melihat besarnya *private memory* yang digunakan khusus untuk permainan warcraft III. Berikut adalah gambar *flow of event* monitor performa:



Generated by UModel

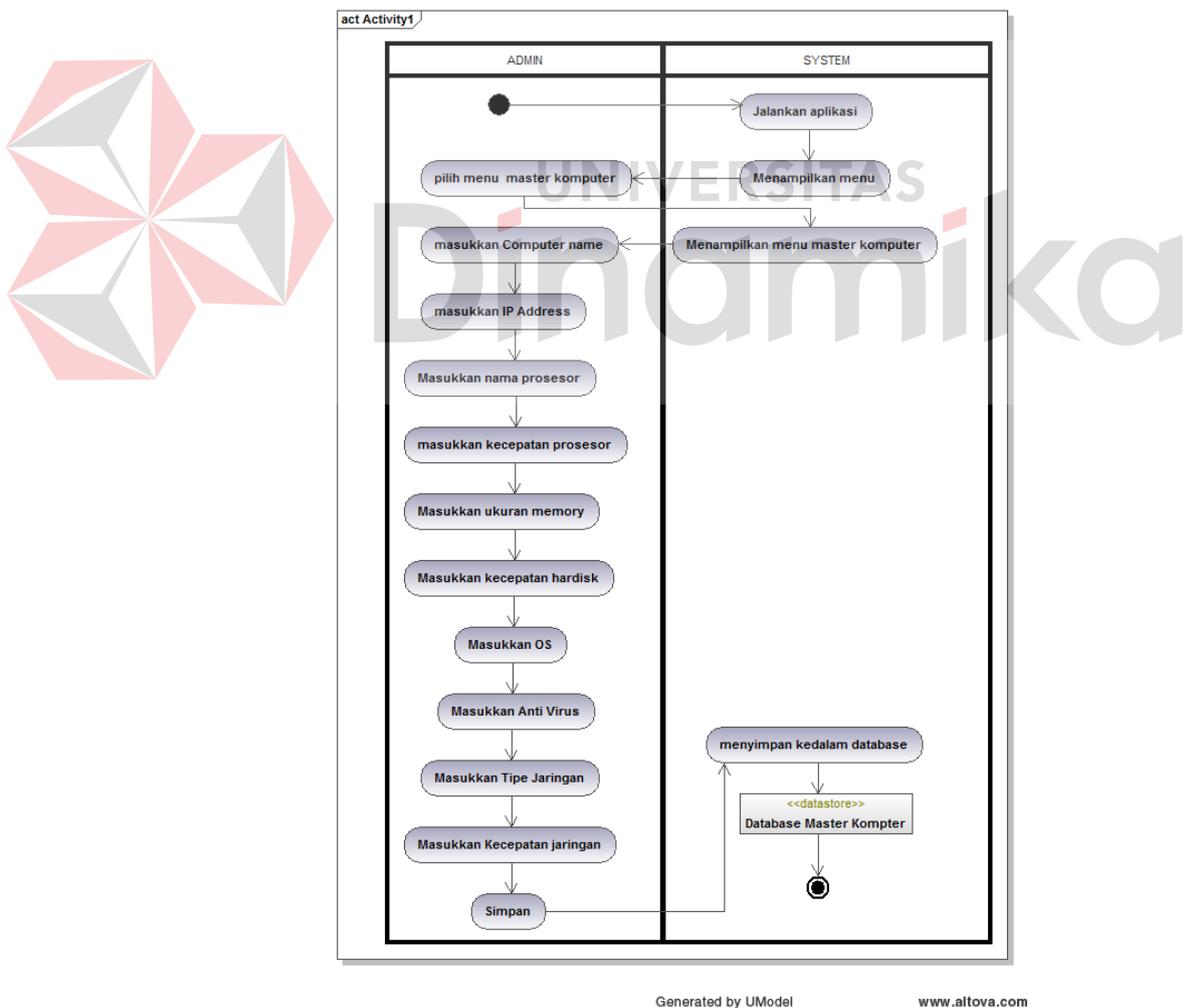
www.altova.com

Gambar 3.9 Flow of Event Monitor Performa

Proses monitor performa dimulai saat permainan dimulai kemudian *admin* melakukan eksekusi monitor performa. Monitor performa secara bersamaan memonitor CPU, *physical memory*, *harddisk* dan Warcraft III *private memory working set*. Setelah permainan berakhir barulah *admin* menyimpan data hasil monitor kedalam *database performance*.

#### 4. Use case Menyimpan Data Master Komputer

*Use case* menyimpan data master komputer digunakan untuk menyimpan data spesifikasi komputer *client* termasuk *server*. Data spesifikasi yang disimpan meliputi nama komputer, *IP Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, nama *operating system*, *antivirus* yang digunakan, tipe jaringan, apakah *wireless* atau kabel dan kecepatan jaringan, apakah *fast ethernet* 10/100 Mbps atau *gigabyte ethernet* 1000 Mbps, standar *wireless* G 54 Mbps, N 108 Mbps, N 150 Mbps, dan N 300 Mbps. Berikut adalah gambar *flow of event* menyimpan data master komputer:

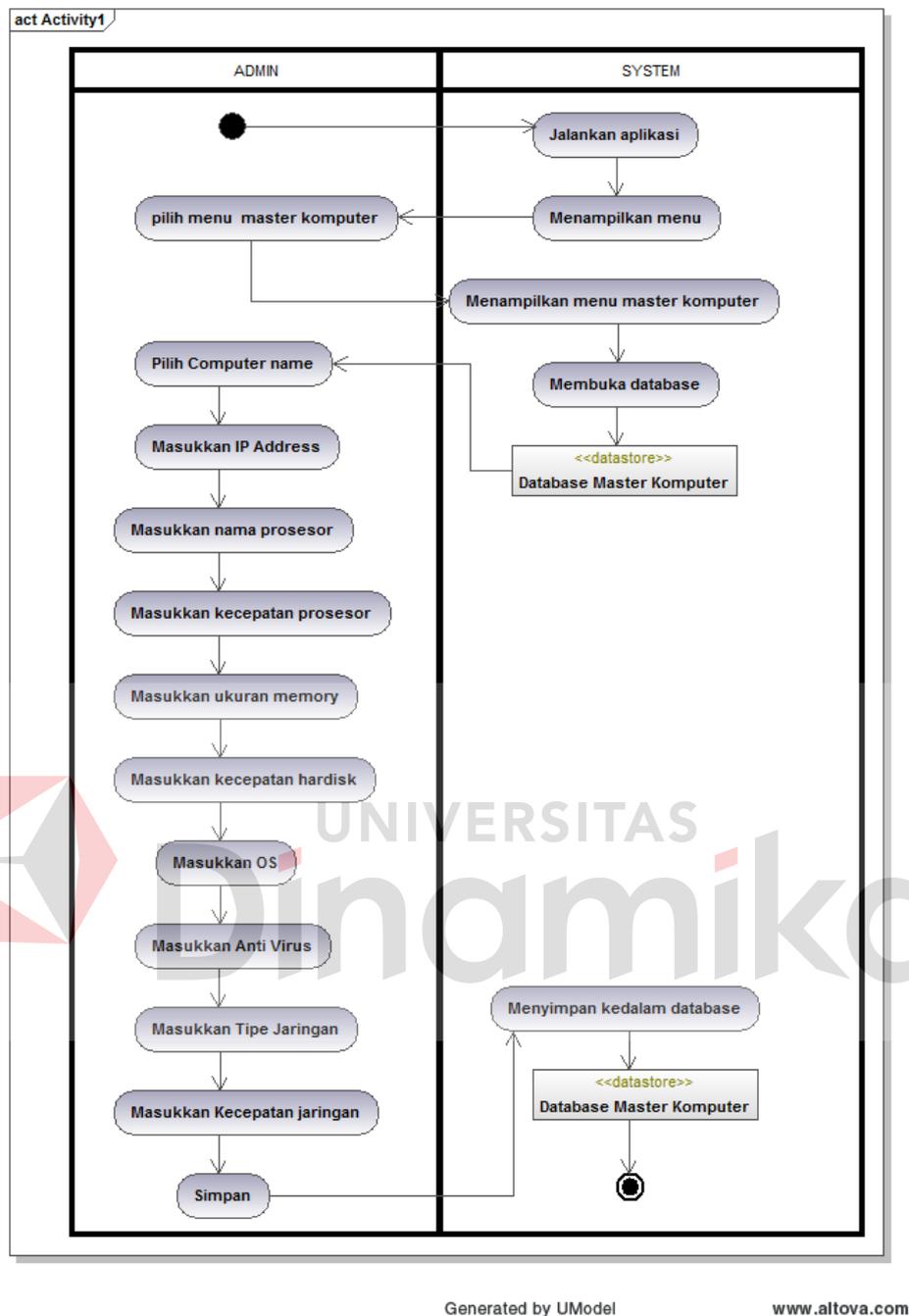


Gambar 3.10 Flow of Event Menyimpan Data Master Komputer

Proses menyimpan data master komputer dimulai dengan *admin* memasukkan spesifikasi *hardware komputer* yang terdiri dari nama komputer, *IP Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, nama *operating system*, *antivirus* yang digunakan, tipe jaringan, apakah *wireless* atau kabel dan kecepatan jaringan, apakah *fast ethernet* 10/100 Mbps atau *gigabyte ethernet* 1000 Mbps, standar *wireless* G 54 Mbps, N 108 Mbps, N 150 Mbps, dan N 300 Mbps. Setelah semua inputan dimasukkan kemudian *admin* mengeksekusi proses simpan data kedalam *database* master komputer.

#### 5. Use case Merubah Data Master Komputer

*Use case* merubah data master komputer digunakan untuk menyimpan perubahan data spesifikasi komputer *client* termasuk *server*. Data spesifikasi yang disimpan meliputi *IP Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, nama *operating system*, *antivirus* yang digunakan, tipe jaringan, apakah *wireless* atau kabel dan kecepatan jaringan, apakah *fast ethernet* 10/100 Mbps atau *gigabyte ethernet* 1000 Mbps, standar *wireless* G 54 Mbps, N 108 Mbps, N 150 Mbps, dan N 300 Mbps. Berikut adalah gambar *flow of event* merubah data master komputer:



Gambar 3.11 Flow of Event Merubah Data Master Komputer

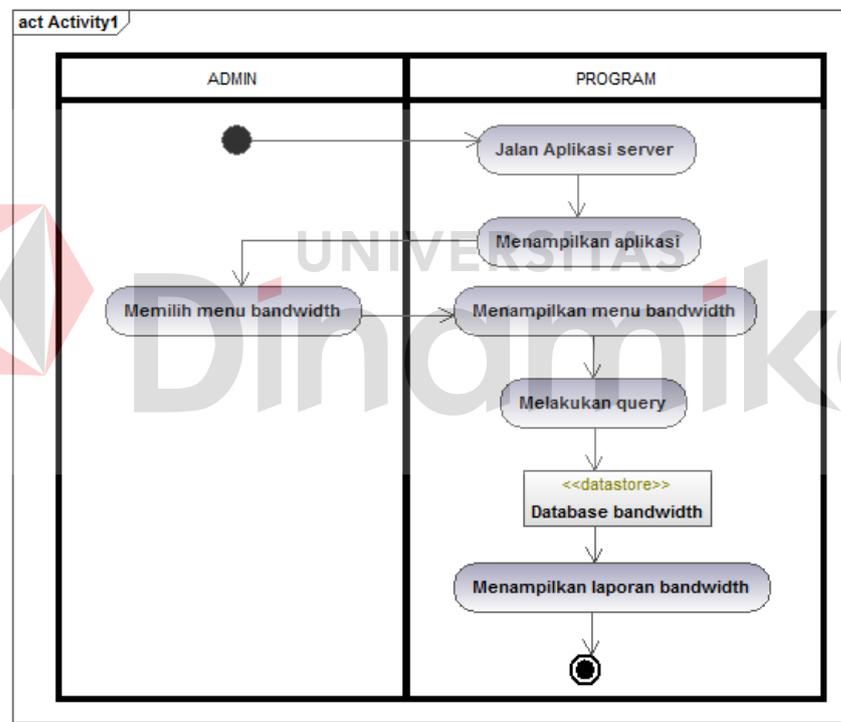
Proses merubah data master komputer dimulai dengan *admin* memilih nama komputer yang akan dirubah kemudian mengganti spesifikasi *hardware* yang terdiri dari *IP Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, nama *operating system*, *antivirus* yang digunakan, tipe jaringan, apakah *wireless* atau kabel dan kecepatan jaringan, apakah *fast*

*ethernet* 10/100 Mbps atau *gigabyte ethernet* 1000 Mbps, standar *wireless* G 54 Mbps, N 108 Mbps, N 150 Mbps, dan N 300 Mbps. Setelah semua dimasukkan kemudian *admin* mengeksekusi proses simpan kedalam *database* master komputer.

#### 6. Use case Melihat Laporan Bandwidth

*Use case* melihat laporan *bandwidth* digunakan untuk melihat laporan hasil proses monitor *bandwidth* yang disajikan dalam bentuk grafik batang.

Berikut adalah gambar *flow of event* melihat laporan *bandwidth* :



Generated by UModel

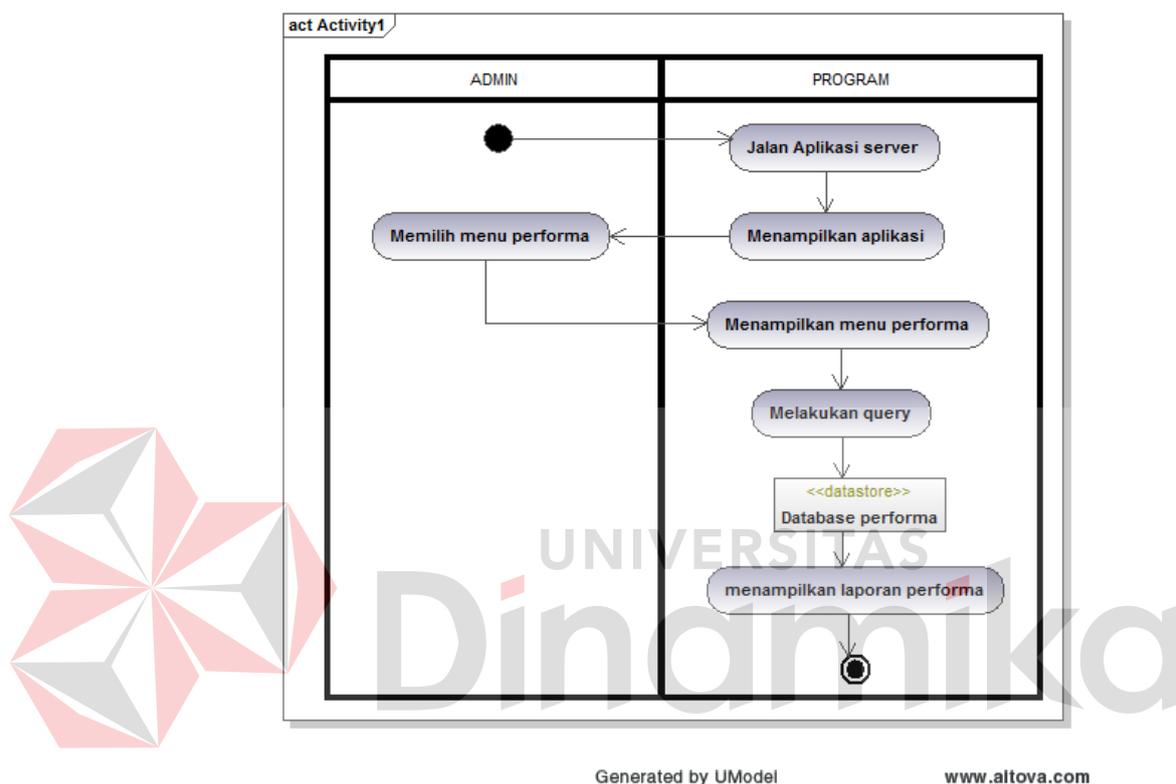
www.altova.com

Gambar 3.12 Flow of Event Melihat Laporan Bandwidth

Proses menampilkan laporan *bandwidth* dimulai dengan menjalankan aplikasi, kemudian *admin* mengeksekusi perintah untuk memunculkan laporan grafik batang *bandwidth*. Sumber data laporan grafik batang *bandwidth* berasal dari *query database bandwidth*.

## 7. Use case Melihat Laporan performa

*Use case* melihat laporan performa digunakan untuk melihat laporan hasil proses monitor performa yang disajikan dalam bentuk grafik batang. Berikut adalah gambar *flow of event* melihat laporan performa:



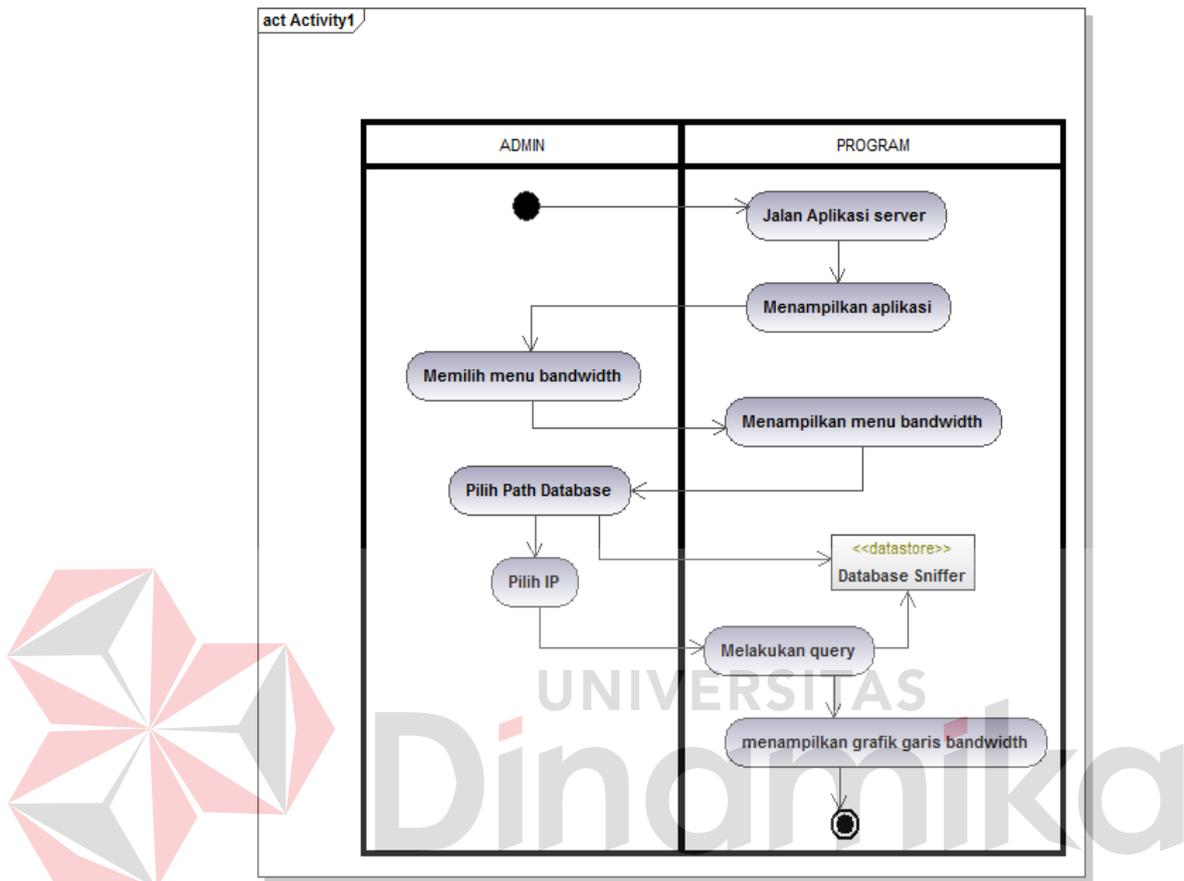
Gambar 3.13 Flow of Event Melihat Laporan Performa

Proses menampilkan laporan performa dimulai dengan menjalankan aplikasi kemudian *admin* mengeksekusi perintah untuk memunculkan laporan performa. Sumber data laporan performa berasal dari *query database* performa.

## 8. Use case Melihat Laporan grafik Garis Bandwidth

*Use case* melihat laporan grafik garis *bandwidth* melihat laporan hasil proses monitor *bandwidth* yang disajikan dalam bentuk grafik garis. Proses penyajian berbeda dengan laporan grafik batang, dalam laporan ini proses penyajian harus diawali dengan memilih *database sniffer* secara individu

kemudian diproses dan disajikan dalam bentuk grafik garis. Berikut adalah gambar *flow of event* melihat ;aporan grafik garis *bandwidth*:



Generated by UModel

www.altova.com

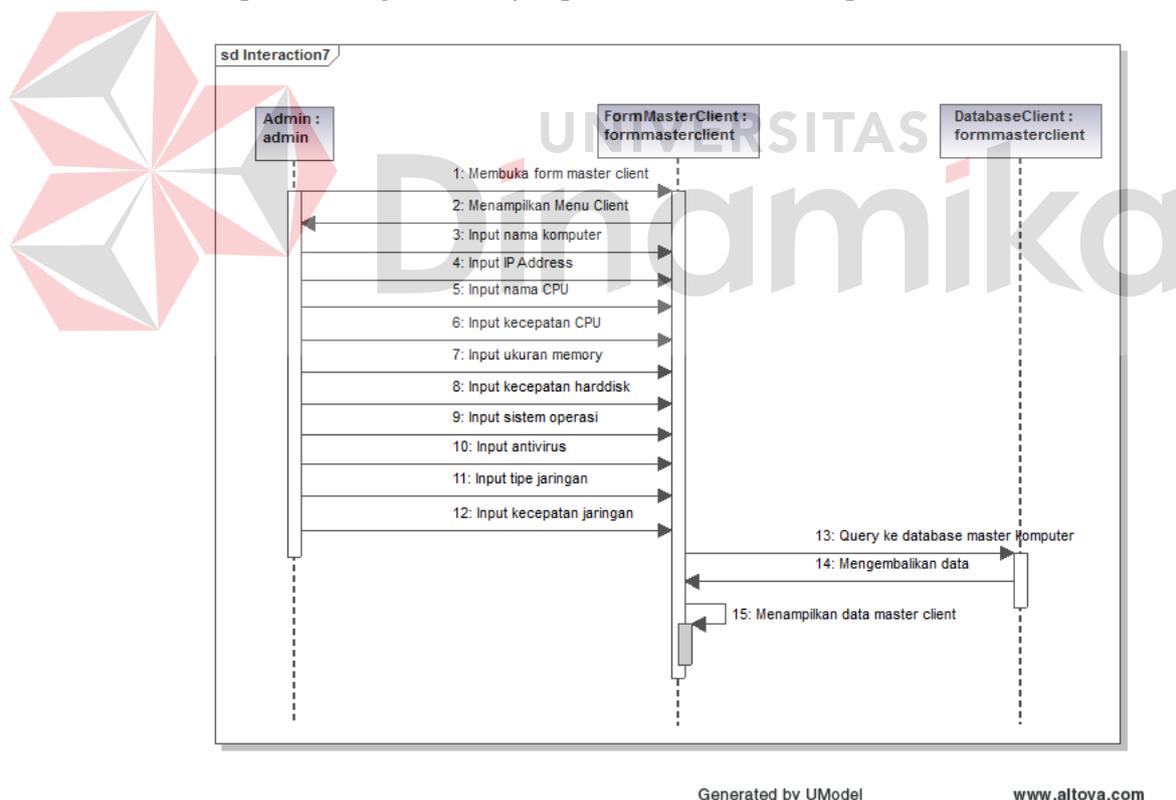
Gambar 3.14 Flow of Event Melihat Laporan Grafik Garis Bandwidth

Proses menampilkan laporan dimulai dengan menjalankan aplikasi kemudian *admin* mengeksekusi perintah untuk memunculkan laporan grafik garis *bandwidth*. Untuk memunculkan laporan grafik garis ini, *admin* harus memilih *database sniffer* yang sudah dikumpulkan sebelumnya, kemudian *admin* memilih *IP Address* dari komputer tempat dimana *sniffer* dilakukan. Setelah itu *admin* melakukan eksekusi proses dan grafik garis dari *database sniffer* tersajikan.

### 3.8.2 Sequence Diagram

*Sequence diagram* merupakan sebuah *diagram* yang menggambarkan interaksi antar objek di dalam sebuah sistem. Interaksi tersebut berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari dimensi *horizontal* (objek-objek) dan dimensi *vertical* (waktu). *Diagram* ini juga menggambarkan urutan *even* yang terjadi. Dan lebih detil dalam menggambarkan aliran data, termasuk data atau *behavior* yang dikirimkan atau diterima. Berikut adalah *sequence diagram* yang terdapat dalam aplikasi monitoring trafik dan performa *server* permainan komputer warcraft III.

#### 1. Sequence Diagram Menyimpan Data Master Komputer

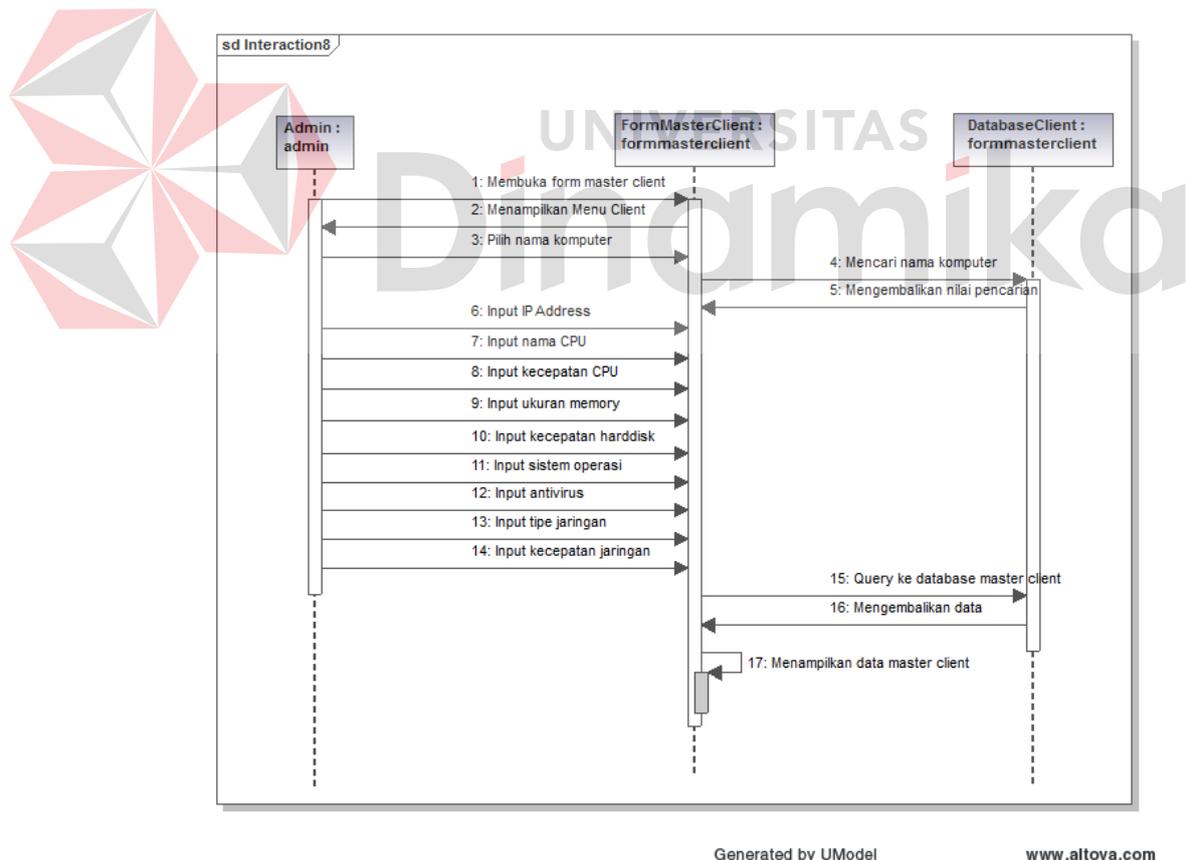


Gambar 3.15 Sequence Menyimpan Data Master Komputer

Penjelasan untuk gambar 3.15, proses menyimpan data spesifikasi komputer *client* dan *server* dalam program yang bermula dari *admin* membuka

*form* aplikasi kemudian dilanjutkan dengan *form master client*. Setelah *form* tampil *admin* memasukkan data teknis *client* yang terdiri dari nama komputer, *IP Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, nama *operating system*, *antivirus* yang digunakan, tipe jaringan, apakah *wireless* atau kabel dan kecepatan jaringan, apakah 10/100 Mbps atau 1000 Mbps, 54 Mbps sampai 300 Mbps. Kemudian *admin* melakukan eksekusi penyimpanan diikuti dengan sistem menyimpan kedalam *database* master komputer. Setelah data tersimpan didalam *database* kemudian sistem melakukan panggilan kembali data untuk ditampilkan di *GridView*.

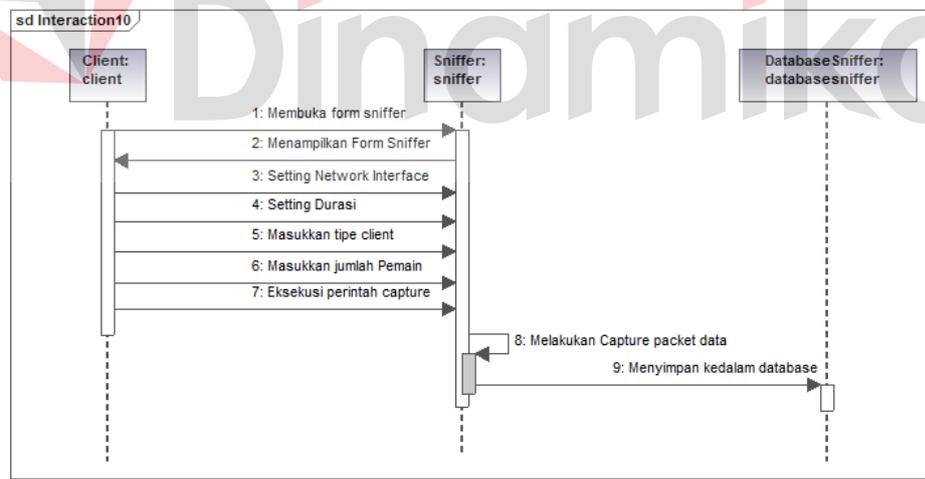
## 2. Sequence Diagram Merubah Data Master Komputer



Gambar 3.16 Sequence Merubah Data Master Komputer

Penjelasan untuk gambar 3.16, proses merubah data spesifikasi komputer *client* dan *server* dalam program yang bermula dari *admin* membuka *form* aplikasi kemudian dilanjutkan dengan *form* master *client*. Setelah *form* tampil *admin* memasukkan data teknis *client* yang terdiri dari *IP Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, nama *operating system*, *antivirus* yang digunakan, tipe jaringan, apakah *wireless* atau kabel dan kecepatan jaringan, apakah 10/100 Mbps atau 1000 Mbps, 54 Mbps sampai 300 Mbps. Kemudian *admin* melakukan eksekusi penyimpanan diikuti dengan sistem menyimpan kedalam *database* master komputer. Setelah data tersimpan didalam *database* kemudian sistem melakukan panggilan kembali data untuk ditampilkan didalam *Gridview*.

### 3. Sequence Diagram Sniffing Packet Data



Generated by UModel

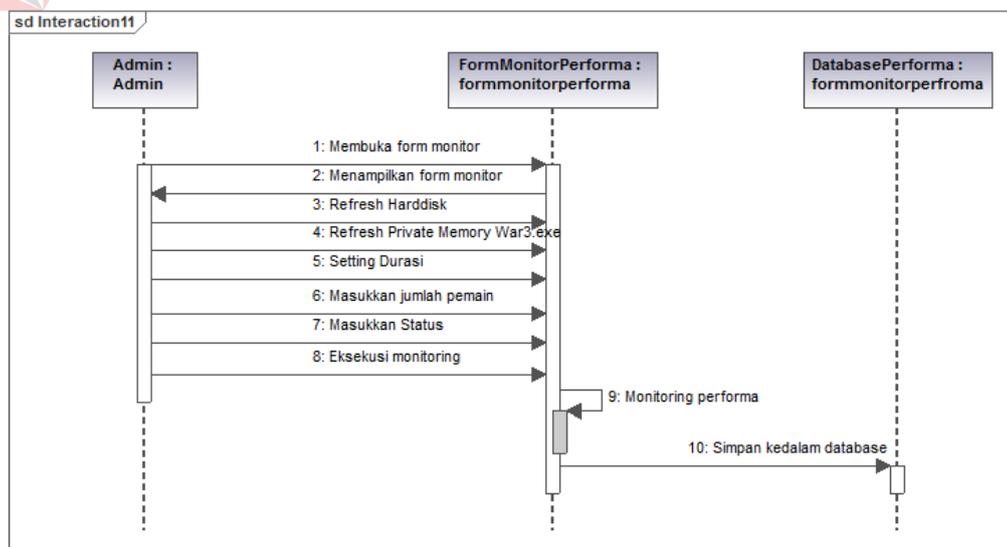
www.altova.com

Gambar 3.17 Sequence Sniffing packet Data

Penjelasan untuk gambar 3.17, proses *sniffing* pada *client* dan *server* bermula dari menampilkan aplikasi *sniffer* disetiap komputer *client* dan *server*. Setelah aplikasi muncul, *client* dan *server* memilih *network interface*, *network*

*interface* tergantung dari *interface* yang digunakan oleh *client* dan *server* saat permainan apakah menggunakan *wireless* atau *kabel*. Setelah melakukan pemilihan *network interface client* dan *server* memasukkan durasi permainan, durasi permainan secara *default* 9999 karena nilai tersebut akan dikurangi dengan nilai lamanya bermain sehingga didapat lamanya *sniffer* melakukan *sniffing* saat permainan mulai sampai selesai. Langkah berikutnya *client* dan *server* memasukkan tipe komputer dirinya sendiri, apakah *client* atau *server*. Setelah memilih tipe komputer *client* dan *server* memasukkan jumlah pemain yang bergabung atau mengikuti permainan dalam *room* yang sudah dibuat oleh *server*. Langkah terakhir yang dilakukan *client* dan *server* adalah melakukan eksekusi *sniffing* dan diikuti oleh sistem yang melakukan *sniffing packet data* dalam *network*. Setelah permainan selesai, proses *sniffing* berhenti dan sistem menyimpan semua *packet data* yang *tersniffing* kedalam *database sniffer*.

#### 4. Sequence Diagram Monitor Performa



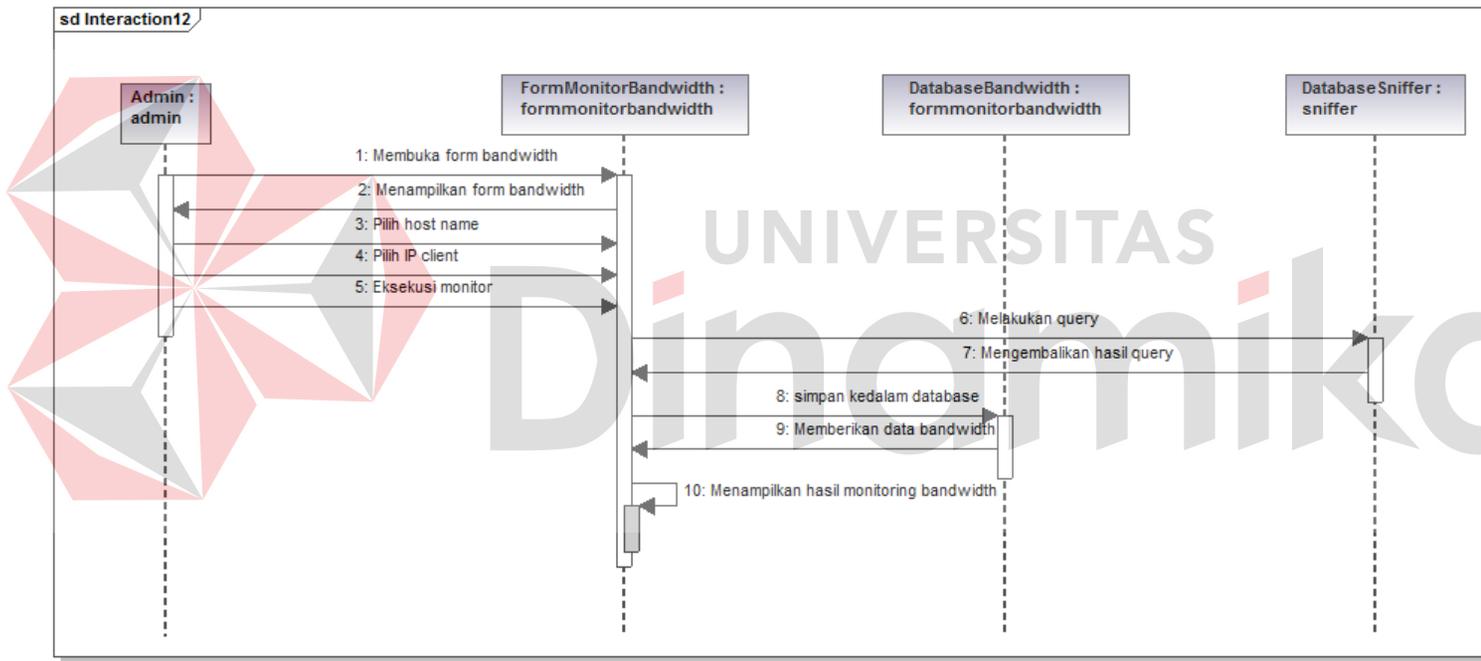
Generated by UModel

www.altova.com

Gambar 3.18 Sequence Monitor Performa

Penjelasan untuk gambar 3.18, proses monitor performa dimulai dengan *admin* membuka aplikasi dan menjalankan *form* monitor performa. Setelah *form* monitor performa terbuka *admin* melakukan *refresh harddisk*, *refresh harddisk* dilakukan untuk mengambil nilai *value* sistem yang akan ditampilkan dalam blok monitor *harddisk*. Untuk dapat memonitor *private memory working set* Warcraft III *admin* harus memilih aplikasi yang akan dimonitor, dalam hal ini Warcraft III. Setelah melakukan setting nilai mesin, *admin* memasukkan durasi lamanya monitor performa, jumlah pemain adalah banyaknya pemain yang mengikuti permainan dan Status komputer adalah tipe komputer yang melakukan monitoring apakah *client* atau *server*. Setelah semua inputan selesai *admin* melakukan eksekusi perintah untuk monitor performa selama permainan berlangsung, secara otomatis sistem akan melakukan monitor performa mesin. Setelah permainan usai, proses monitoring dihentikan dan nilai yang didapat disimpan kedalam *database* performa. Performa yang dimonitor adalah CPU (*Central Processing Unit*), *physical memory*, *private memory working set* Warcraft III dan *harddisk*.

5. Sequence Diagram Monitor Bandwidth



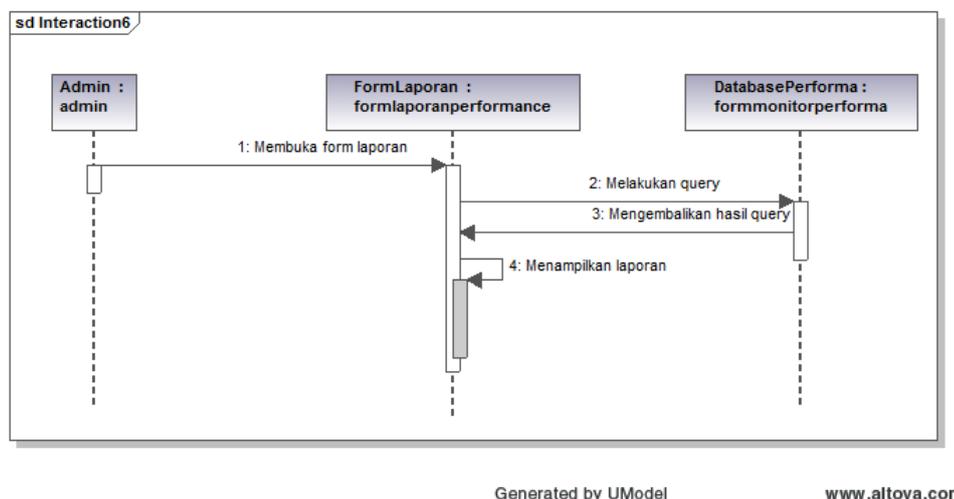
Generated by UModel

www.altova.com

Gambar 3.19 Sequence Monitor Bandwidth

Penjelasan untuk gambar 3.19, proses monitor *bandwidth* dimulai dengan *admin* membuka aplikasi dan menjalankan *form* monitor *bandwidth*. Setelah *form* monitor *bandwidth* terbuka *admin* memilih komputer *name* dan *IP Address client*, kemudian melakukan eksekusi untuk memonitor *bandwidth*. Monitor *bandwidth* dilakukan oleh sistem dengan memasukkan *query* ke *database sniffer* untuk mendapatkan nilai – nilai *bandwidth* yaitu *upload*, *download*, *average*, *peak* dan *broadcast*. *Upload* adalah besarnya *bandwidth* yang keluar dari setiap komputer, *download* adalah besarnya *bandwidth* yang masuk kedalam komputer setiap komputer, *average* adalah besarnya rata – rata *bandwidth upload* dan *download* dari setiap komputer, *peak* adalah *bandwidth* terbesar *upload* atau *download* dari setiap komputer dan *broadcast* adalah *bandwidth listener* yang digunakan *client* dan *server* untuk melakukan komunikasi antar *client* dan *server*. Setelah sistem melakukan *query bandwidth* kedalam *database sniffer client*, kemudian nilainya dikembalikan ke *server* monitor untuk disimpan kedalam *database bandwidth* yang ada di *server*.

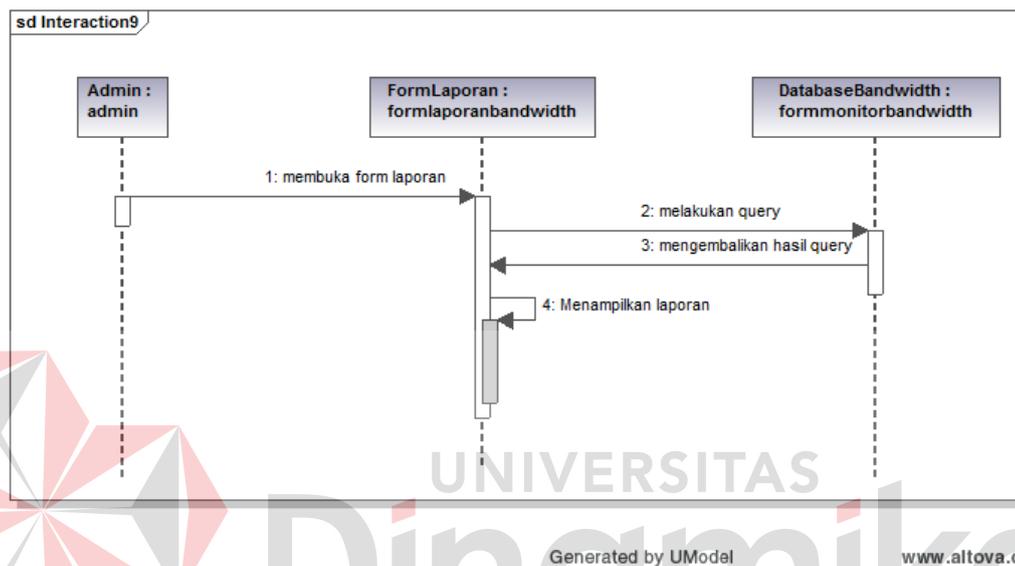
## 6. Sequence Diagram Menampilkan Laporan Performa



Gambar 3.20 Sequence Menampilkan laporan Performa

Penjelasan untuk gambar 3.20, proses menampilkan laporan performa dimulai dengan membuka *form* laporan. *Form* laporan performa menyajikan laporan grafik batang, sumber data berasal dari *database* performa yang kemudian ditampilkan berdasarkan beberapa kriteria.

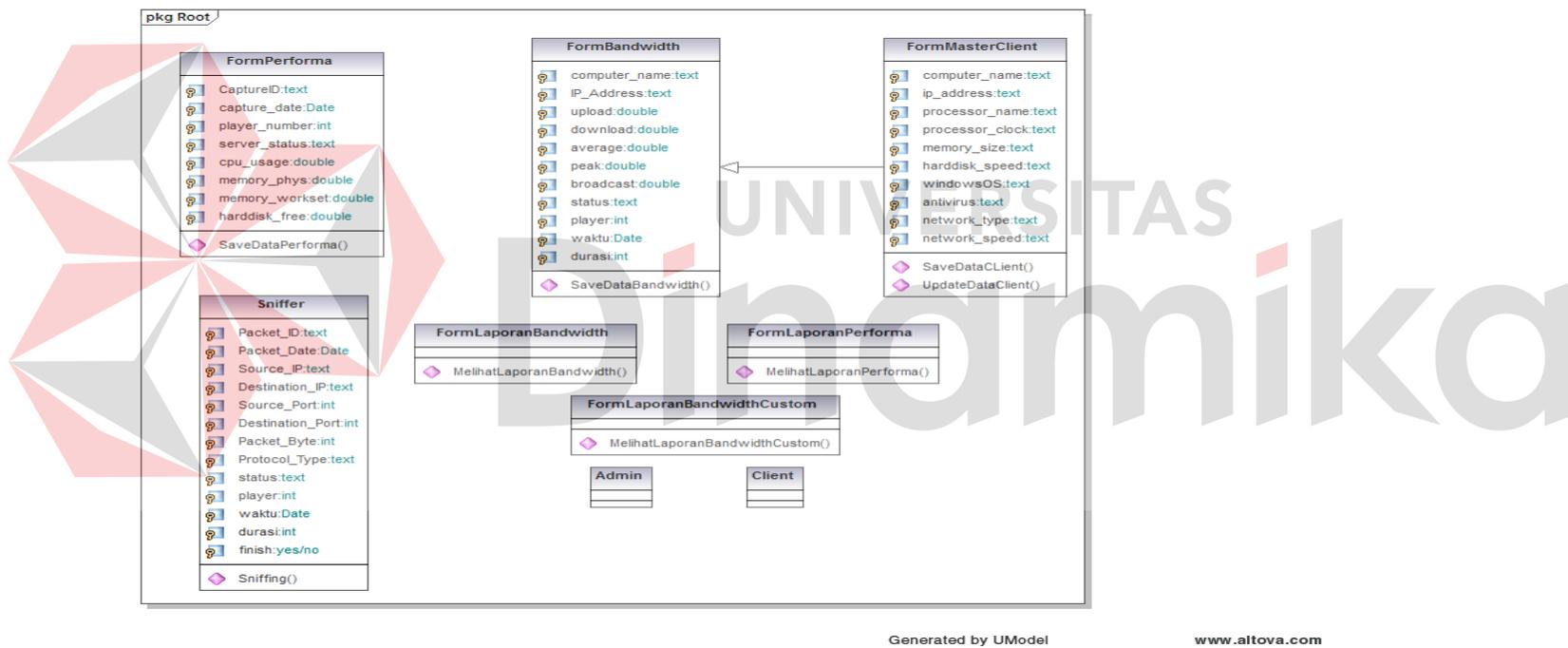
#### 7. Sequence Diagram Menampilkan Laporan Bandwidth



Gambar 3.21 Sequence Menampilkan Laporan Bandwidth

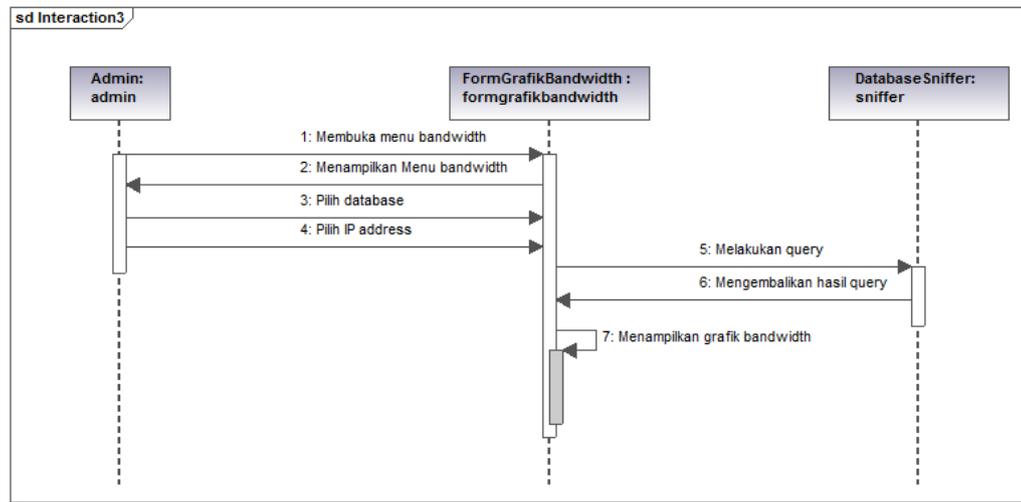
Penjelasan untuk gambar 3.21, proses menampilkan laporan *bandwidth* dimulai dengan membuka *form* laporan. *Form* laporan *bandwidth* menyajikan laporan grafik batang, sumber data berasal dari *database bandwidth* yang kemudian ditampilkan berdasarkan beberapa kriteria.

## 1. Class Diagram Monitor Bandwidth dan Performa



Gambar 3.22 Class Diagram Monitor Bandwidth and Performa

## 8. Sequence Diagram Menampilkan laporan Grafik Bandwidth



Generated by UModel

www.altova.com

Gambar 3.23 Sequence Menampilkan Laporan Grafik Bandwidth

Penjelasan untuk gambar 3.23, proses menampilkan laporan grafik garis *bandwidth*. Bermula dengan *admin* membuka *form* laporan grafik garis *bandwidth*, kemudian *form* laporan akan tampil. Sebelum grafik garis dapat disajikan, *admin* harus memilih *database sniffer* dari *client* untuk diproses menjadi laporan grafik garis. Kemudian *admin* memilih *IP Address* dari *client* yang memiliki *database sniffer* tersebut. Berbeda dengan laporan grafik batang *bandwidth*, laporan grafik *bandwidth* melihat naik turunnya *bandwidth* selama permainan, dengan begitu dapat dilihat pola *bandwidth* selama jalannya permainan. Laporan grafik garis *bandwidth* dibuat untuk melengkapi analisa dari pola *bandwidth* setiap *client*.

### 3.8.3 Class Diagram

Penjelasan untuk gambar 3.22, *Class Diagram* digunakan untuk menampilkan kelas-kelas dan paket-paket yang terdapat pada sistem. *Class*

*Diagram* menggambarkan objek-objek yang ada didalam sistem serta menggambarkan hubungan antar objek didalam sistem. Pada *class diagram* monitor *bandwidth* dan performa *server* Warcraft III terdapat 4 *class diagram* utama dan 3 *class diagram* laporan. 4 *class diagram* utama digunakan untuk melakukan monitor *bandwidth* dan performa *server*, sedangkan 3 *class diagram* laporan untuk menampilkan laporan dari data hasil monitor.

### 3.9. Struktur Basis Data

1. Nama Tabel : Packet Data
- Primary Key : Packet\_id
- Foreign Key : -
- Fungsi : Untuk menyimpan *packet data* hasil *capture* dengan *sniffer*.

Tabel 3.1 Struktur Tabel Packet Data

| No | Kolom            | Type Data | Panjang | Batasan     |
|----|------------------|-----------|---------|-------------|
| 1  | Packet_id        | Text      | 50      | Primary Key |
| 2  | Packet_Date      | Date      | 50      |             |
| 3  | Source_Ip        | Text      | 50      |             |
| 4  | Destination_IP   | Text      | 50      |             |
| 5  | Source_Port      | Number    |         |             |
| 6  | Destination_Port | Number    |         |             |
| 7  | Packet_Byte      | Number    |         |             |
| 8  | Protocol_Type    | Text      | 50      |             |

2. Nama Tabel : Respon
- Primary Key : Finish
- Foreign Key :

Fungsi : Menyimpan data dengan *value true* atau *false* sebagai tanda apakah proses *capture* sudah selesai.

Tabel 3.2 Struktur Tabel Respon

| No | Kolom  | Tipe Data | Panjang | Batasan     |
|----|--------|-----------|---------|-------------|
| 1  | Finish | Yes/No    |         | Primary key |

3. Nama Tabel : Information

Primary Key : -

Foreign Key : -

Fungsi : Untuk menyimpan *packet data* hasil *capture* dengan *sniffer*.

Tabel 3.3 Struktur Tabel Information

| No | Kolom  | Tipe Data | Panjang | Batasan |
|----|--------|-----------|---------|---------|
| 1  | Status | Text      | 50      |         |
| 2  | Player | Number    |         |         |
| 3  | Durasi | Text      | 50      |         |
| 4  | Waktu  | Number    |         |         |

4. Nama Tabel : Bandwidth

Primary Key : computer\_name

Foreign Key : -

Fungsi : Menyimpan data hasil *query bandwidth client* termasuk *server*.

Tabel 3.4 Struktur Tabel Bandwidth

| No | Kolom         | Tipe Data | Panjang | Batasan     |
|----|---------------|-----------|---------|-------------|
| 1  | computer_name | Text      | 50      | Primary key |
| 2  | IP_Address    | Text      | 50      |             |
| 3  | Upload        | Number    |         |             |
| No | Kolom         | Tipe Data | Panjang | Batasan     |
| 4  | Download      | Number    |         |             |
| 5  | average       | Number    |         |             |
| 6  | peak          | Number    |         |             |
| 7  | Broadcast     | Number    |         |             |
| 8  | Status        | Text      | 50      |             |
| 9  | Player        | Number    |         |             |
| 10 | waktu         | Date      |         |             |
| 11 | durasi        | Number    |         |             |

3. Nama Tabel : Master Komputer

Primary Key : computer\_name

Foreign Key :-

Fungsi : Menyimpan data spesifikasi komputer.

Tabel 3.5 Struktur Tabel Master Komputer

| No | Kolom           | Tipe Data | Panjang | Batasan     |
|----|-----------------|-----------|---------|-------------|
| 1  | computer_name   | Text      | 50      | Primary key |
| 2  | ip_address      | Text      | 50      |             |
| 3  | processor_name  | Text      | 50      |             |
| 4  | processor_clock | Text      | 50      |             |
| 5  | Memory_size     | Text      | 50      |             |
| 6  | Harddisk_speed  | Text      | 50      |             |
| 7  | windowsOS       | Text      | 50      |             |
| 8  | antivirus       | Text      | 50      |             |
| 9  | network_type    | Text      | 50      |             |
| 10 | network_speed   | Text      | 50      |             |

4. Nama Tabel : Performa

Primary Key : capture\_id

Foreign Key :-

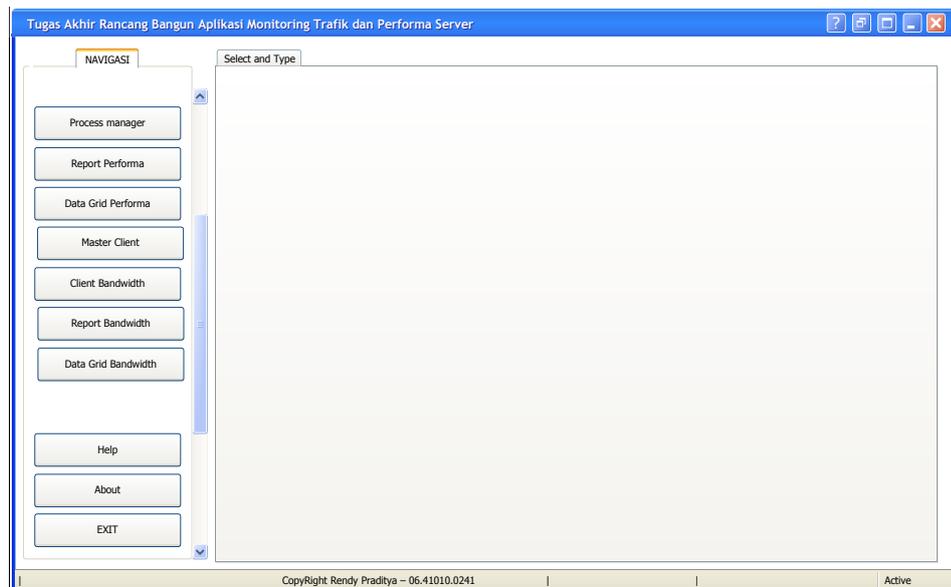
Fungsi : Menyimpan data hasil monitor performa saat permainan berjalan.

Tabel 3.6 Struktur Tabel Performa

| No | Kolom          | Type Data | Panjang | Batasan     |
|----|----------------|-----------|---------|-------------|
| 1  | capture_id     | Text      | 50      | Primary Key |
| 2  | capture_date   | Text      | 50      |             |
| 3  | player_number  | Number    |         |             |
| 4  | server_status  | Text      | 50      |             |
| 5  | CPU_usage      | Number    |         |             |
| 6  | CPUmax         | Number    |         |             |
| 7  | CPUmin         | Number    |         |             |
| 8  | Memory_phys    | Number    |         |             |
| 9  | Memory_workset | Number    |         |             |
| 10 | Harddisk_free  | Number    |         |             |
| 11 | HDD_read       | Number    |         |             |
| 12 | HDD_write      | Number    |         |             |

### 3.10. Desain User Interface

#### 3.10.1 Form Utama



Gambar 3.24 Desain Form Utama

Penjelasan gambar 3.24, pada *form* menu utama terdapat dua menu utama yaitu performa dan *bandwidth*.

Sub menu performa yaitu :

1 Process Manager

*Process Manager* adalah *form* yang menangani monitoring performa *server*.

2 Report Process Manager

*Report Process Manager* adalah *form* laporan untuk performa

3 Performance GridView

*Performance GridView* adalah *form* yang menampilkan semua data yang ada didalam *database* performa.

Sub menu *Bandwidth* yaitu :

4 Client Bandwidth

*Client Bandwidth* adalah *form* yang menangani *bandwidth* monitoring (proses pengolahan *bandwidth*).

5 Client dan Server Master

*Client* dan *server master* adalah *form* *maintenance* data master *client* dan *server*, fungsi *insert* dan fungsi *update* data

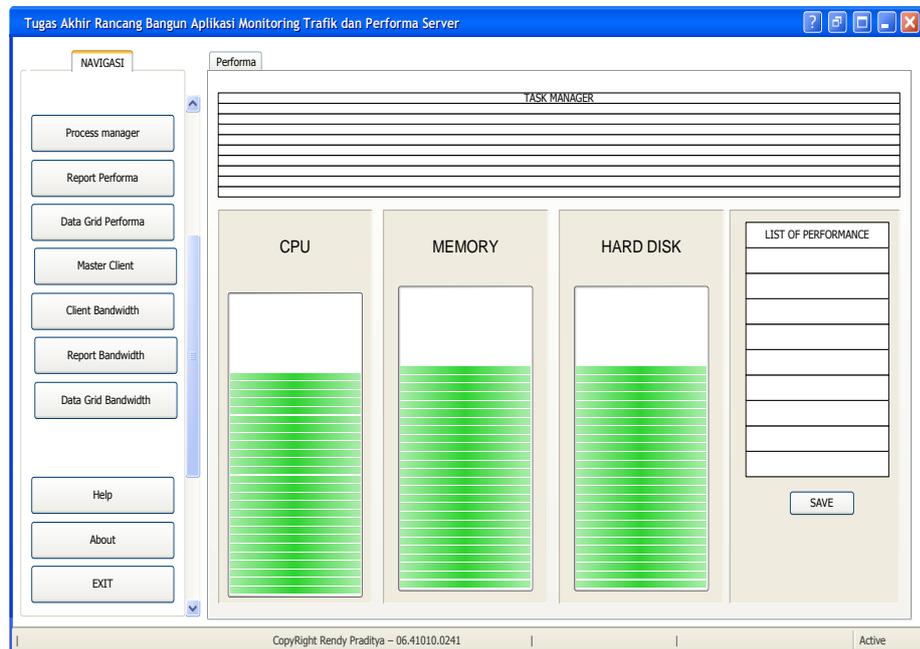
6 Report Bandwidth

*Report bandwidth* adalah *form* laporan untuk *bandwidth*

7 Bandwidth GridView

*Bandwidth GridView* adalah *form* yang menampilkan semua data yang ada didalam *database* *bandwidth*

### 3.10.2 Form Performance Monitor



Gambar 3.25 Desain Form Server Performance Monitoring

Penjelasan gambar 3.25, *form server performance* monitoring digunakan untuk melakukan monitoring performa *server*. Terdapat 4 *progress bar* yang menunjukkan aktifitas *CPU*, *memory*, *harddisk* dan *private memory* Warcraft III.

Awal proses *capture* performa dengan menekan tombol *start* saat permainan dijalankan. Setelah waktu durasi habis maka secara otomatis *capture* performa berhenti. Hasil dari *running average* akan tampil setelah *capture* selesai kemudian tombol *save* untuk menyimpan hasilnya. Tombol *report* digunakan untuk menampilkan *form report* performa.

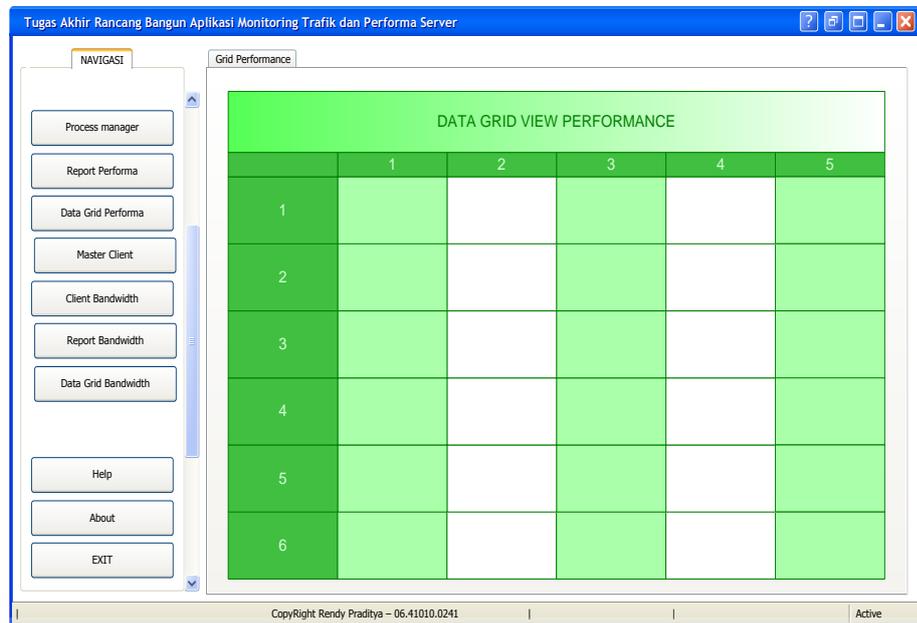
### 3.10.3 Form Report Performance



Gambar 3.26 Desain Form Report performance

Penjelasan gambar 3.26, *form report performance* digunakan untuk melihat hasil *capture* performa yang telah tersimpan didalam *database*. Penyajian laporan performa disajikan dengan beberapa grafik batang dengan kriteria tertentu agar mudah dalam melakukan perbandingan hasil *capture*. Kriteria yang disajikan statis, tidak dapat menerima inputan dari *user*. Beberapa kriteria yang wajib untuk disajikan adalah CPU, *physical memory*, *harddisk* dan *private Warcraft III memory working set*.

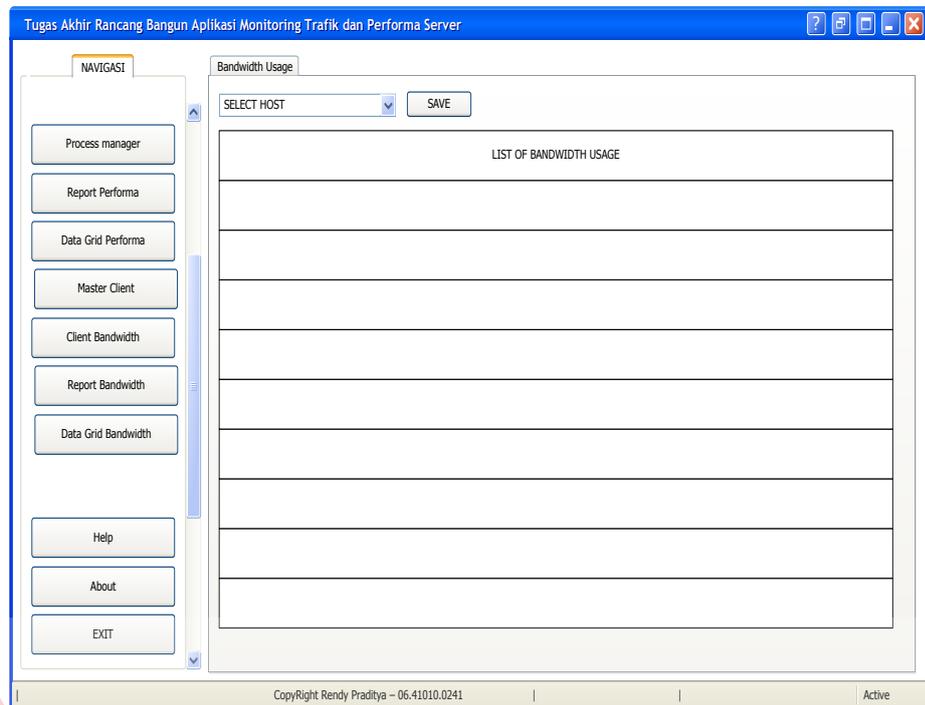
### 3.10.4 Form Data Monitor Performa



Gambar 3.27 Desain Form Data Monitor Performa

Penjelasan gambar 3.27, *form* data monitor performa berisi semua data yang tersimpan dalam *database* performa yang disajikan secara berurutan. Fungsinya hampir sama dengan laporan grafik tetapi *form* data monitor performa memberikan data secara detail. *Form* data monitor performa tidak memberikan fungsi menyimpan atau mengubah data yang ada didalam *database* performa, karena *form* data monitor performa hanya bersifat menampilkan data saja. Fungsi *sorting* data dapat dilakukan di tabel data, untuk mempermudah melihat data berdasarkan kolom. Fungsi lainnya yaitu fungsi pengelompokan data berdasarkan kolom, fungsi pengelompokan data untuk mempermudah melihat data yang dikelompokkan.

### 3.10.5 Form Client and Server Bandwidth Usage



Gambar 3.28 Desain Form Client and Server Bandwidth Usage

Penjelasan gambar 3.28, *form client* dan *server bandwidth usage* berfungsi untuk mengambil data *bandwidth* dari *client* dan *server*. *Form client* dan *server bandwidth usage* digunakan untuk mengambil data – data *bandwidth* dari *client* maupun *server* kemudian diproses untuk menghasilkan *upload*, *download*, *average*, *peak* dan *broadcast*. Cara kerja dari proses pengambilan data dengan cara :

1. Setiap *client* terhubung dengan jaringan lokal, termasuk *server*.
2. Setiap *client* melakukan *sharing folder sniffer* yang diletakkan di partisi C:\ agar *server* dapat melakukan *query*.
3. Setiap *client* dan *server* harus menonaktifkan *firewall* baik yang bawaan dari Microsoft Windows atau dari *antivirus*.
4. Setiap *client* harus terdaftar pada *database master client*.

Jika 4 syarat tadi sudah terpenuhi, langkah selanjutnya adalah memilih nama *client* dari *combobox client* kemudian secara otomatis proses *query* dan perhitungan berjalan. Setelah proses *query* dan perhitungan selesai kemudian tombol *save* yang digunakan untuk menyimpan kedalam database *bandwidth* untuk disajikan kedalam laporan.

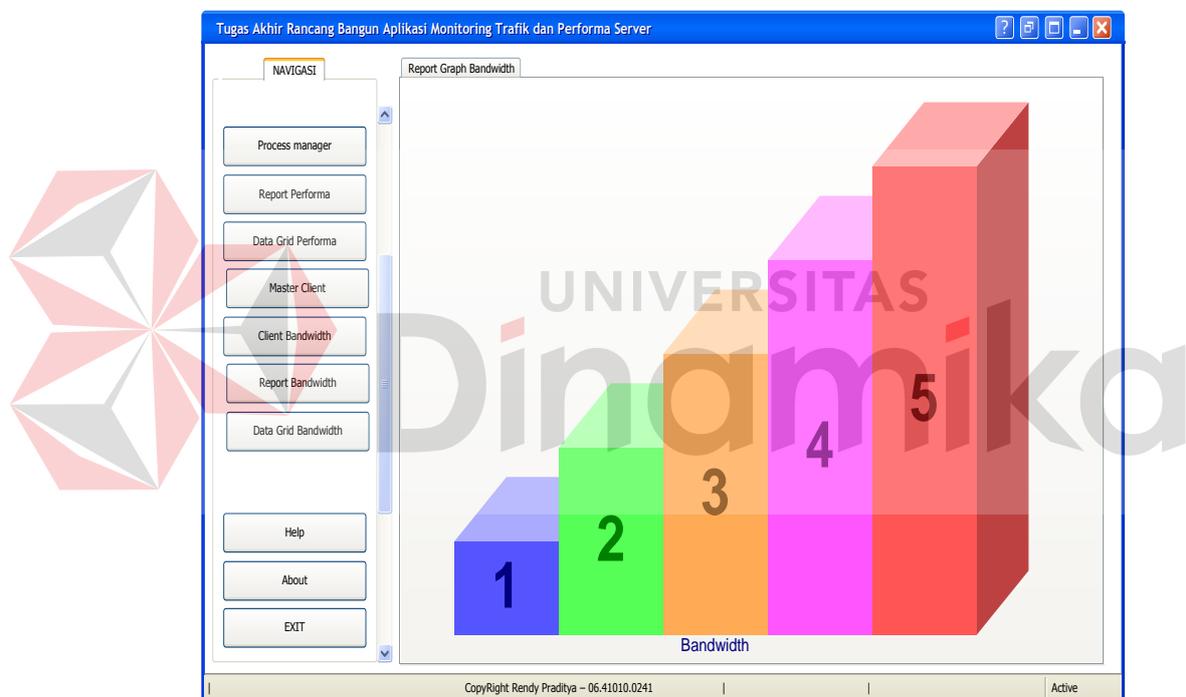
### 3.10.6 Form Master Client

Gambar 3.29 Desain Form Master Client dan Server

Penjelasan gambar 3.29, *form master client* dan *server* digunakan untuk menyimpan spesifikasi *Client*. Cara kerja dari *form master client* dan *server* cukup dengan mengisi semua data teknis *hardware* dan *software* dari setiap mesin yaitu nama komputer, alamat *IP Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, sistem operasi, *antivirus*, tipe jaringan dan kecepatan jaringan, kemudian menekan tombol simpan, semua data teknis tadi

akan disimpan kedalam *database*. Untuk melakukan perubahan data teknis yang ada didalam *database*, cukup dengan memilih *client* atau *server* yang ada di daftar kemudian secara otomatis data akan muncul kedalam kotak inputan, untuk menyimpan perubahan cukup dengan menekan tombol edit pada *form*. Untuk fungsi hapus hampir sama dengan fungsi ubah, pilih *client* atau *server* yang ada dalam daftar kemudian menekan tombol hapus.

### 3.10.7 Form Report Bandwidth



Gambar 3.30 Desain Form Report Bandwidth

Penjelasan gambar 3.30, *form* laporan *bandwidth* berguna untuk menampilkan laporan grafik batang *bandwidth* yang isinya hasil *query* dari *database bandwidth*. Laporan grafik batang *bandwidth* menampilkan beberapa grafik batang untuk mempermudah perbandingan dari beberapa kriteria tertentu.

Laporan grafik batang *bandwidth* menampilkan keseluruhan data yang ada didalam *database bandwidth*

### 3.10.8 Form Data Monitor Trafik



Gambar 3.31 Desain Data Monitor Trafik

Penjelasan untuk gambar 3.31, *form* data monitor trafik digunakan untuk menampilkan seluruh data *bandwidth* yang tersimpan didalam *database bandwidth*. Fungsi dari data *list bandwidth* hampir sama dengan laporan grafik tetapi menampilkan data secara detil. Data yang ditampilkan yaitu nama komputer, *IP Address*, *upload*, *download*, *average*, *peak*, *broadcast*, status, jumlah *player*, waktu dan durasi dari *database bandwidth*. Data yang ada didalam *list* tabel tidak dapat dirubah, hanya untuk ditampilkan saja. Fungsi *sorting* data dapat dilakukan di tabel data, untuk mempermudah melihat data berdasarkan kolom. Fungsi lainnya yaitu fungsi pengelompokan data berdasarkan kolom,

fungsi pengelompokan *bandwidth* untuk mempermudah melihat data yang dikelompokkan.

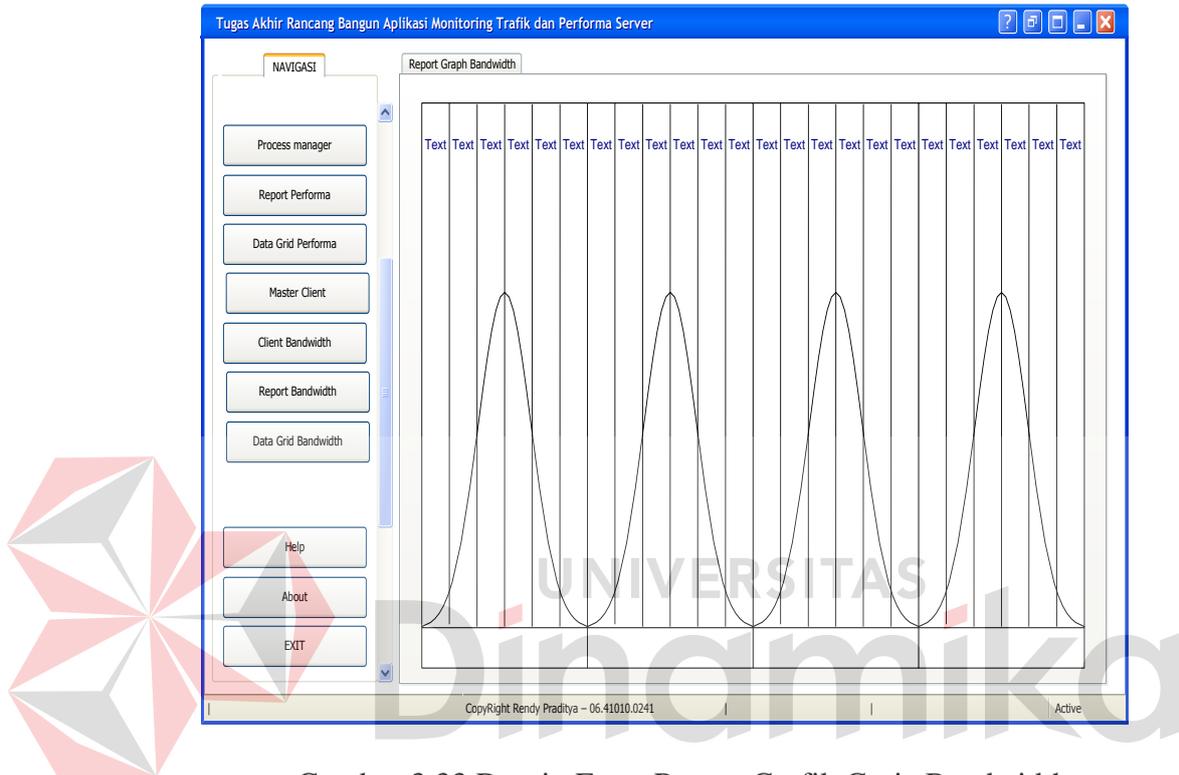
### 3.10.9 Form Client dan Server Bandwidth Offline

Gambar 3.32 Desain Form Client dan Server Bandwidth Offline

Penjelasan gambar 3.32, *form client* dan *server bandwidth offline* digunakan untuk mengambil data – data *bandwidth* dari *client* maupun *server* kemudian diproses untuk menghasilkan *upload*, *download*, *average*, *peak* dan *broadcast* secara *offline*. Cara kerja dari pengambilan data *bandwidth* dengan mengambil semua *database sniffer* yang ada di *client* atau *server*. Berbeda dengan *form bandwidth client* dan *server* secara *on-line*, pengambilan data di *form client* dan *server bandwidth offline* tidak membutuhkan komputer *client* lagi karena *database* harus berada didalam *server* dengan cara dikopi. *Form client* dan *server*

*bandwidth offline* menjadi alternatif jika pengumpulan data secara *on-line* menjadi sulit karena keberadaan komputer *client* tidak selalu *ready* atau berada ditempat.

### 3.10.10 Form Report Grafik Garis Bandwidth



Gambar 3.33 Desain Form Report Grafik Garis Bandwidth

Penjelasan gambar 3.33, *form report* grafik garis *bandwidth* digunakan menampilkan report grafik garis *bandwidth usage*. *Form report* grafik garis *bandwidth* berbeda dengan *form* grafik batang *bandwidth*, karena *form report* grafik garis *bandwidth* digunakan untuk menganalisa *bandwidth* dari *database sniffer* secara individual. Kemudian *form report* grafik garis *bandwidth* digunakan untuk melihat naik turunnya *bandwidth* sehingga dapat melihat pola gerakan *bandwidth*.

### 3.10.11 Form Sniffer

| SOURCE IP | DEST IP | SOURCE PORT | DESTINATION PORT | PACKET BYTE | PROTOCOL |
|-----------|---------|-------------|------------------|-------------|----------|
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |
|           |         |             |                  |             |          |

Gambar 3.34 Desain Form Sniffer

Penjelasan gambar 3.34, *form sniffer* digunakan untuk melakukan *sniffing* paket data di setiap komputer, *client* maupun *server*. *Sniffing* paket data berguna mengkoleksi semua paket data yang melewati trafik jaringan, termasuk paket data yang bukan dari permainan Warcraft III. *Sniffing* dilakukan saat permainan berlangsung, dari awal permainan sampai berakhir. Setelah *sniffing* selesai data *sniffing* otomatis tersimpan kedalam *database sniffer* yang ada di *folder* program *sniffer*.

### 3.10.12 Form Laporan Tabel Perbandingan

Gambar 3.35 Desain Form Laporan Tabel Perbandingan

Penjelasan gambar 3.35, *form* laporan tabel perbandingan digunakan untuk melakukan perbandingan performa dan trafik *server* dengan jumlah pemain yang disajikan dengan menampilkan 5 tabel jumlah pemain mulai dari dua pemain sampai sepuluh pemain. Data yang ditampilkan bersumber dari *database* performa dan trafik server. Fungsi dari laporan tabel perbandingan hampir sama dengan laporan data monitor performa dan trafik server, tetapi menampilkan data dengan memisahkan dan membandingkan data performa dan trafik server dengan jumlah pemain. Data yang dibandingkan untuk performa server yaitu *cpu usage*, *cpu max*, *cpu min*, *physical memory*, *private memory working set*, *harddisk read* dan *harddisk write*, dan untuk trafik yaitu *upload*, *download*, *average*, *peak* dan *broadcast*. Data yang ada didalam tabel tidak dapat dirubah, hanya untuk ditampilkan dan dibandingkan saja.

## BAB IV

### IMPLEMENTASI DAN EVALUASI

#### 4.1. Implementasi

Implementasi program adalah implementasi dari analisis dan desain sistem yang dibuat sebelumnya. Sehingga diharapkan dengan adanya implementasi dapat dipahami jalannya aplikasi Rancang Bangun Monitoring Trafik dan Performa *Server* Permainan Komputer (Studi Kasus Warcraft III). Sebelumnya user harus mempersiapkan kebutuhan-kebutuhan dari program yang akan diimplementasikan baik dari segi perangkat keras maupun perangkat lunak komputer. Perangkat lunak yang dibangun dikembangkan dengan menggunakan Microsoft Visual Basic 2008, sedangkan sistem operasi menggunakan Microsoft Windows 7 Ultimate 32 Bits.

#### 4.2. Kebutuhan Sistem

Aplikasi monitoring trafik dan performa *server* Warcraft III memerlukan perangkat lunak (*software*) dan perangkat keras (*hardware*), agar dapat berjalan sesuai dengan yang diharapkan.

##### 4.2.1 Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang digunakan yaitu :

1. Sistem operasi menggunakan Microsoft Windows 7 Ultimate.
2. *Database* untuk pengolahan data menggunakan Microsoft Access 2010.

3. Bahasa pemrograman yang digunakan adalah Microsoft Visual Basic .NET 2008 Professional.
4. .Net Framework Minimal Versi 3.0.
5. Untuk *report* menggunakan Crystal Reports for Visual Studio .NET 2008.
6. Komponen menggunakan Developer Express.Net v9.3 for Visual Studio .NET 2008.
7. Untuk perancangan sistem menggunakan Altova UML 2010.
8. Untuk perancangan desain input/output menggunakan Microsoft Office Visio 2007.
9. Untuk dokumentasi menggunakan Microsoft Office Word 2010.

#### **4.2.2 Kebutuhan Perangkat Keras**

Perangkat keras yang digunakan yaitu :

1. *Processor* Intel Dual Core, AMD64 3000+, atau di atasnya.
2. *Memory* 1024 Mb atau lebih.
3. *Harddisk* 5 Gb atau lebih.
4. Monitor dengan resolusi minimal 1024 x 768.
5. *VGA Card* 128 MB, *Mouse*, dan *keyboard*.

#### **4.3. Instalasi Program dan Pengaturan Sistem**

Aplikasi monitoring trafik dan performa *server* Warcraft III memerlukan perangkat lunak yang telah terinstal, adapun tahapan-tahapan instalasi dan pengaturan sistem, yaitu :

1. Microsoft Windows 7
2. WinPcap

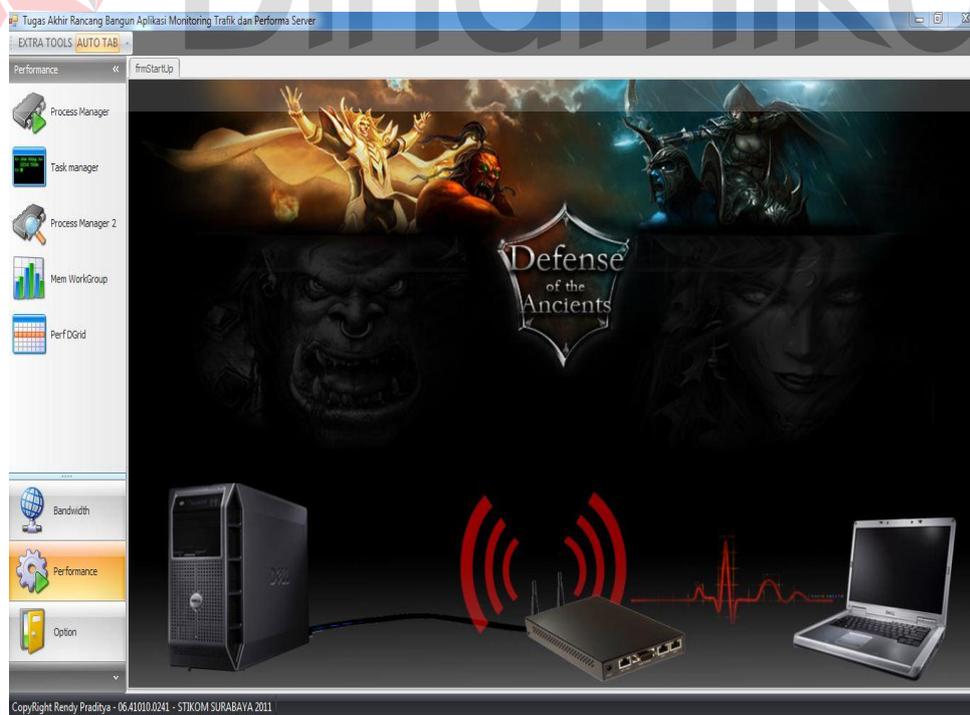
### 3. Microsoft Access 2010.

Proses monitoring *bandwidth* cukup dengan melakukan *copy paste folder sniffer* ke dalam partisi C:\ termasuk *server*. Untuk monitoring performa, *server* harus melakukan *copy paste* aplikasi monitoring performa dan *bandwidth* ke sembarang tempat, aplikasi *client* sebaiknya diletakkan pada partisi C:\ karena path *database* pada program *sniffer client* ditujukan ke C:\.

#### 4.4. Penjelasan Pemakaian Aplikasi.

Sistem monitoring trafik dan performa *server* Warcraft III dapat dijalankan setelah dilakukan tahapan-tahapan instalasi program seperti di atas. Aplikasi monitor trafik dan performa *server* Warcraft III terdiri dari dua bagian, *sniffer* serta monitoring performa dan trafik.

##### 4.4.1 Menu Utama



Gambar 4.1 Form Menu Utama

Penjelasan gambar 4.1, *form* menu utama digunakan untuk melakukan monitoring performa *server* dan *bandwidth client server*. Aplikasi monitor trafik dan performa *server* terdiri dari 3 menu, yang pertama menu performa, yang isinya yaitu:

1. Proses manajer
2. Laporan proses manajer
3. List data proses manajer

Proses manajer berguna untuk melakukan monitoring CPU, *physical memory*, *harddisk* dan Warcraft III *private memory*. Laporan proses manajer berguna untuk melihat hasil proses monitoring performa yang tersimpan didalam *database*. List data proses manajer berguna untuk melihat semua data yang tersimpan didalam *database* secara detail.

Menu kedua yaitu *Bandwidth* yang terdiri dari 4 sub menu, yaitu :

1. Client bandwidth
2. Client master
3. Laporan grafik batang bandwidth
4. List data bandwidth
5. Client bandwidth offline
6. Laporan grafik garis bandwidth

*Client bandwidth* berguna untuk mengumpulkan query data *bandwidth* dari *client* yang terdiri dari nama komputer, IP Address, Upload, Download, Average, peak, broadcast, Status, Jumlah *Player*, Tanggal permainan, Durasi permainan. *Client master* berguna untuk menyimpan data spesifikasi *hardware* dan *software client*. Laporan *bandwidth* berguna untuk melihat hasil proses

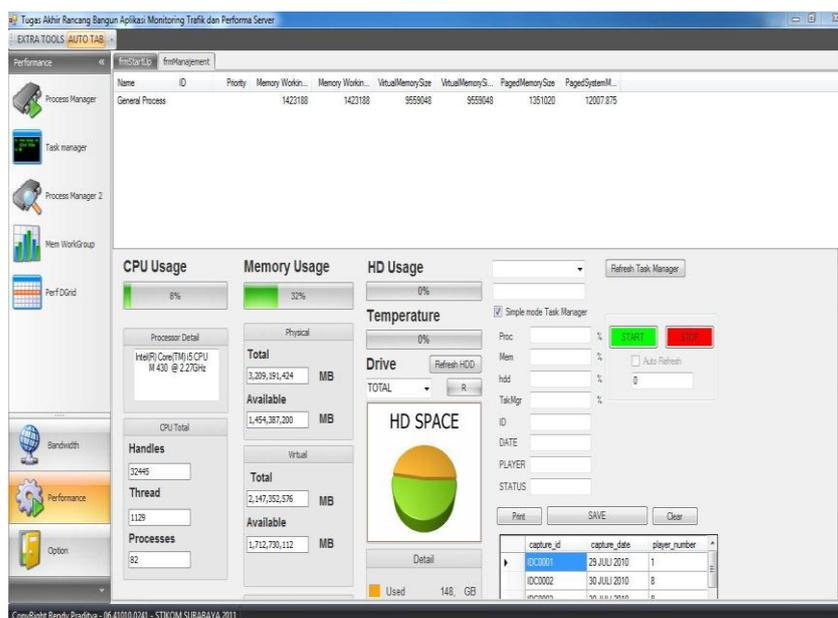
pengumpulan data *bandwidth* dari *client*. List data *bandwidth* berguna untuk melihat keseluruhan data *client* yang tersimpan didalam *database*. *Client bandwidth offline* fungsinya hampir sama dengan *client bandwidth* tetapi dilakukan di komputer *server*. Laporan grafik garis fungsinya hampir sama dengan laporan grafik batang *bandwidth*, tetapi *database sniffer* diproses secara individu.

Menu ketiga yaitu *Option* yang terdiri dari 3 sub menu, yaitu :

1. Exit
2. Help
3. About

Tombol *exit* digunakan untuk keluar dari aplikasi. Tombol *Help* digunakan untuk melihat manual dari program. Tombol *About* untuk melihat informasi *developer* tugas akhir ini.

#### 4.4.2 Proses Manajer

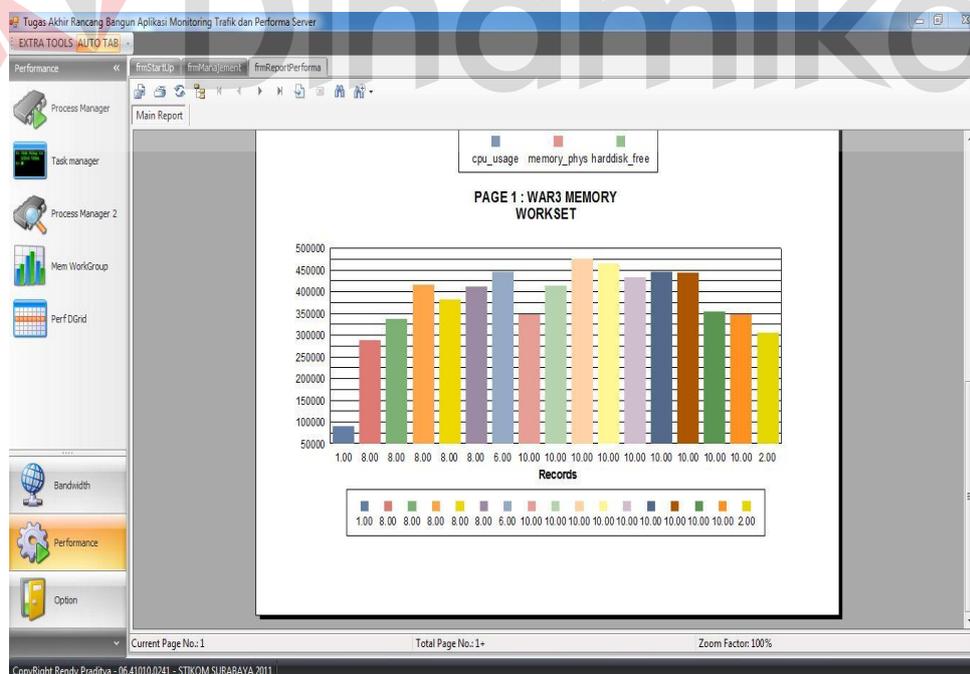


Gambar 4.2 Form Proses Manajer.

Penjelasan gambar 4.2, *form* proses manajer digunakan untuk melakukan proses monitoring performa *server* yang terdiri dari 4 bagian, yaitu *CPU*, *physical memory*, *harddisk* dan *private memory* Warcraft III. Sebelum proses monitoring dilakukan, terlebih dahulu mengisi aplikasi yang akan dimonitor (dalam kasus ini War3.exe), kapasitas *harddisk*, durasi *capture* performa, tanggal permainan, jumlah pemain, dan status komputer yang menjalankan aplikasi ini. Setelah mengisi semua data tadi barulah melakukan proses *capture* performa dengan menekan tombol *start* saat permainan dijalankan. Setelah waktu durasi habis maka secara otomatis *capture* performa berhenti. Hasil dari *running average* akan tampil setelah *capture* selesai kemudian tombol *save* untuk menyimpan hasilnya.

Tombol *report* digunakan untuk menampilkan *form report* performa.

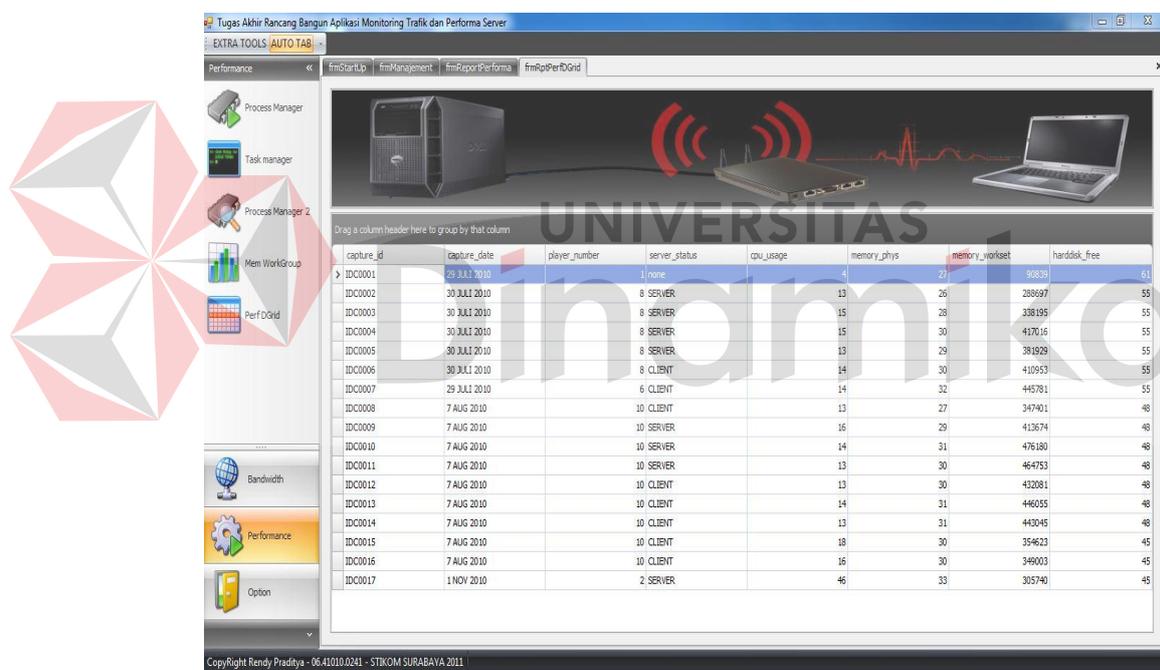
#### 4.4.3 Form Laporan Performa



Gambar 4.3 Form Laporan Performa

Penjelasan gambar 4.3, *form* laporan performa digunakan untuk melihat hasil *capture* performa yang telah tersimpan didalam *database*. Penyajian laporan performa disajikan dengan beberapa grafik batang dengan kriteria tertentu agar mudah dalam melakukan perbandingan hasil *capture*. Kriteria yang disajikan statis, tidak dapat menerima inputan dari *user*. Beberapa kriteria yang wajib untuk disajikan adalah CPU, *physical memory*, *harddisk* dan *private Warcraft III memory workset*.

#### 4.4.4 Form Data Monitor Performa

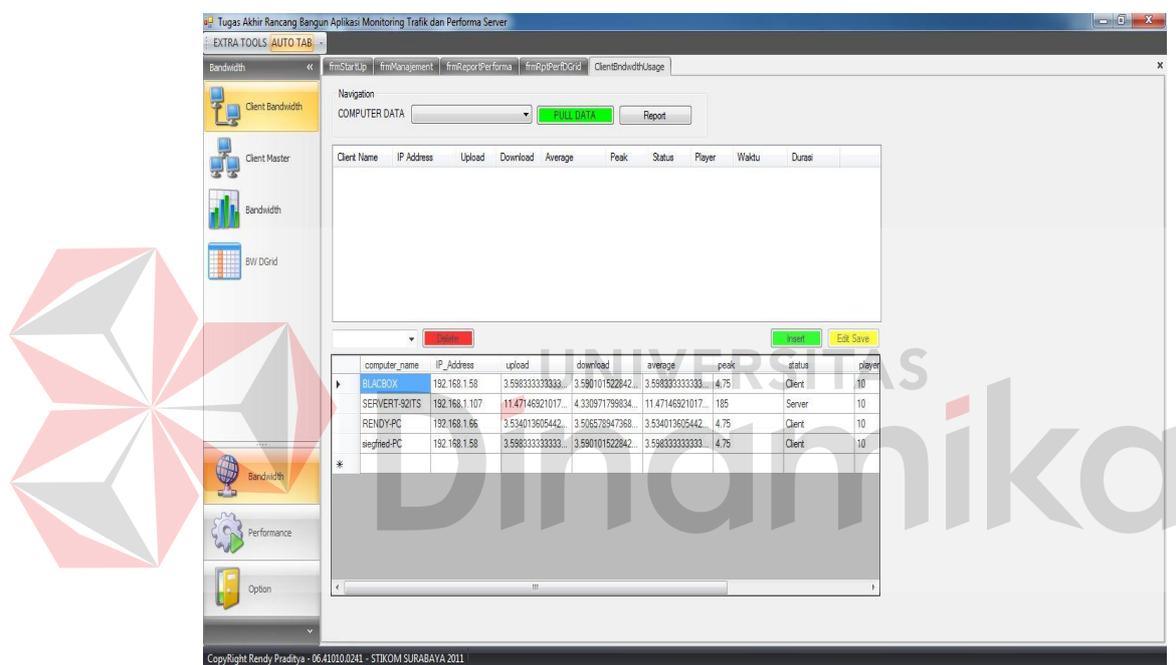


Gambar 4.4 Form Data Monitor Performa

Penjelasan gambar 4.4, *form* data monitor performa berisi semua data yang tersimpan dalam *database* performa yang disajikan secara berurutan. Fungsinya hampir sama dengan laporan grafik tetapi *form* data monitor performa memberikan data secara detail. *Form* data monitor performa tidak memberikan fungsi menyimpan atau mengubah data yang ada didalam *database* performa,

karena *form* data monitor performa hanya bersifat menampilkan data saja. Fungsi *sorting* data dapat dilakukan di tabel data, untuk mempermudah melihat data berdasarkan kolom. Fungsi lainnya yaitu fungsi pengelompokan data berdasarkan kolom, fungsi pengelompokan data untuk mempermudah melihat data yang dikelompokkan.

#### 4.4.5 Form Client dan Server Bandwidth



Gambar 4.5 Form Client dan Server Bandwidth

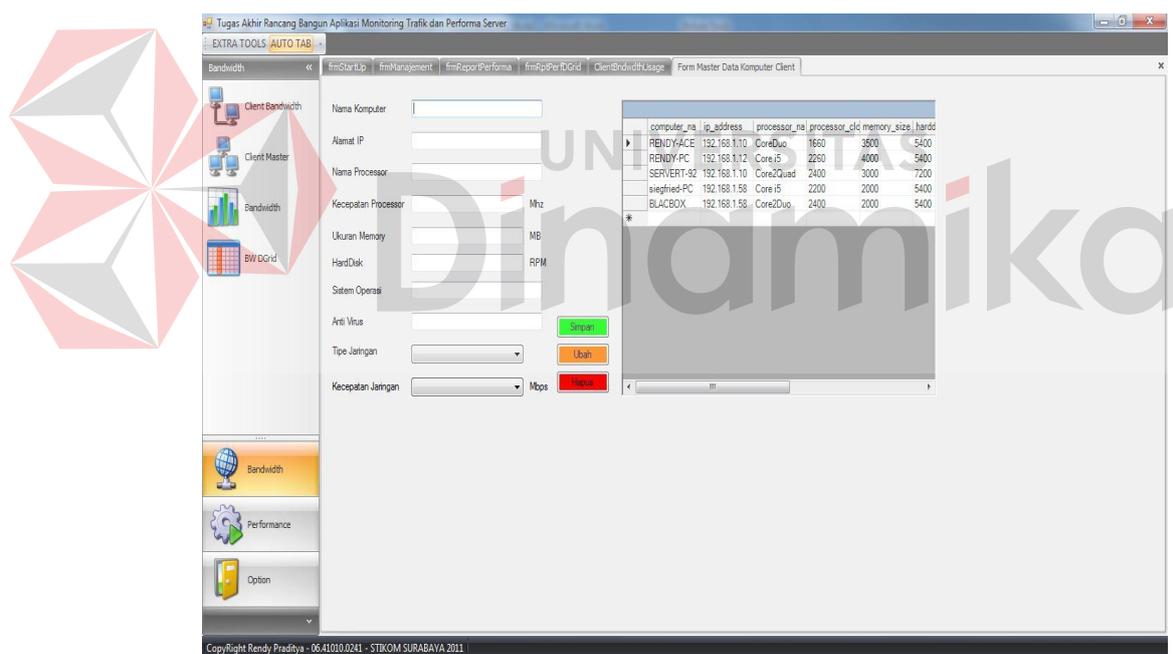
Penjelasan gambar 4.5, *form client* dan *server bandwidth* digunakan untuk mengambil data – data *bandwidth* dari *client* maupun *server* kemudian diproses untuk menghasilkan *upload*, *download*, *average*, *peak* dan *broadcast*. Cara kerja dari proses pengambilan data *bandwidth* dengan cara :

1. Setiap *client* terhubung dengan jaringan lokal, termasuk *server*.
2. Setiap *client* melakukan *sharing folder sniffer* yang diletakkan di partisi C:\ agar *server* dapat melakukan *query*.

3. Setiap *client* dan *server* harus menonaktifkan *firewall* baik yang bawaan dari Microsoft Windows atau dari *antivirus*.
4. Setiap *client* harus terdaftar pada *database* master *client*.

Jika 4 syarat tadi sudah terpenuhi, langkah selanjutnya adalah memilih nama *client* dari *combobox* *client* kemudian secara otomatis proses *query* dan perhitungan berjalan. Setelah proses *query* dan perhitungan selesai kemudian tombol *save* yang digunakan untuk menyimpan kedalam *database* *bandwidth* untuk disajikan kedalam laporan.

#### 4.4.6 Form Master Client dan Server

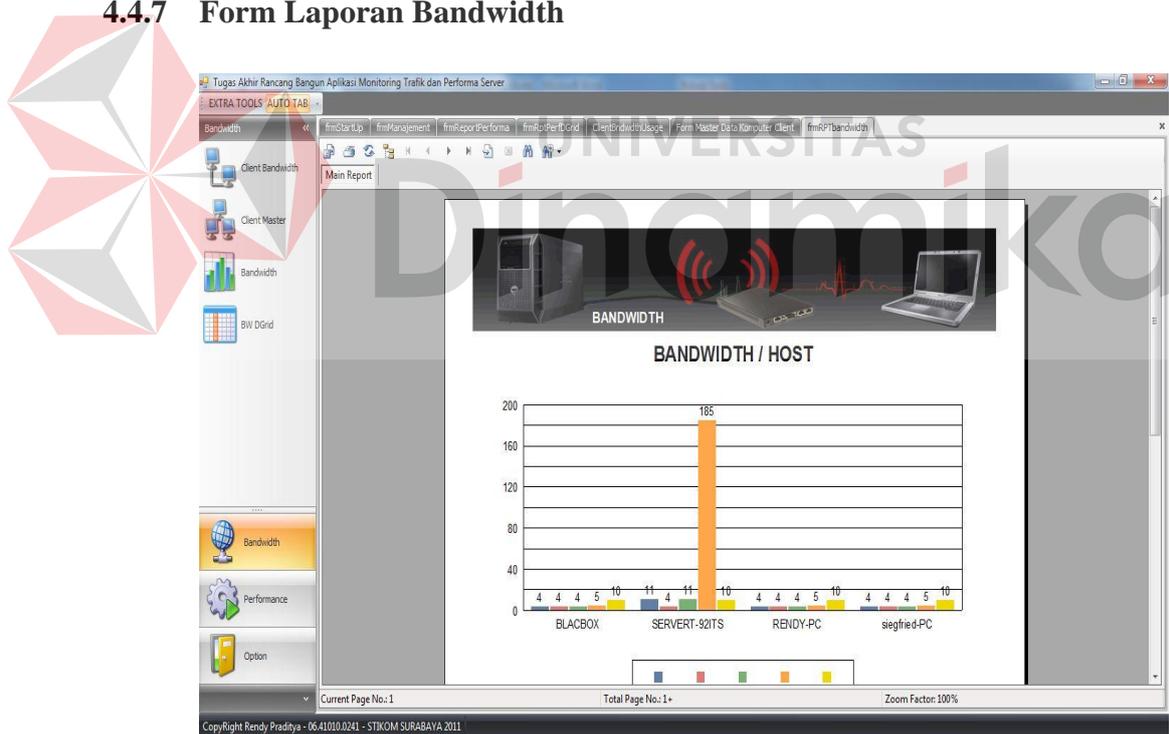


Gambar 4.6 Form Master Client dan Server

Penjelasan gambar 4.6, *form* master *client* dan *server* digunakan untuk menyimpan data spesifikasi *hardware* dan *software* semua *client* termasuk *server*. Cara kerja dari *form* master *client* dan *server* cukup dengan mengisi semua data teknis *hardware* dan *software* dari setiap mesin yaitu nama komputer, alamat *IP*

*Address*, nama *processor*, kecepatan *processor*, ukuran *memory*, kecepatan *harddisk*, sistem operasi, *antivirus*, tipe jaringan dan kecepatan jaringan, kemudian menekan tombol simpan, semua data teknis tadi akan disimpan kedalam *database*. Untuk melakukan perubahan data teknis yang ada didalam *database*, cukup dengan memilih *client* atau *server* yang ada didaftar kemudian secara otomatis data akan muncul kedalam kotak inputan, untuk menyimpan perubahan cukup dengan menekan tombol edit pada *form*. Untuk fungsi hapus hampir sama dengan fungsi ubah, pilih *client* atau *server* yang ada dalam daftar kemudian menekan tombol hapus.

#### 4.4.7 Form Laporan Bandwidth

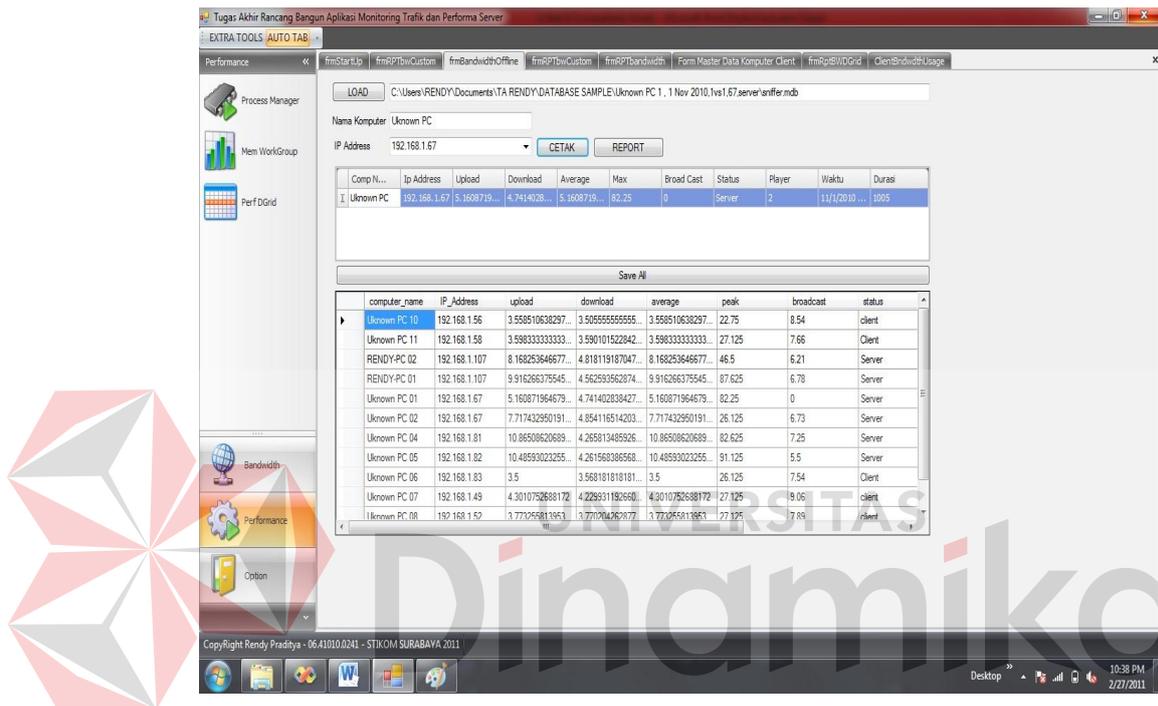


Gambar 4.7 Form Laporan Bandwidth

Penjelasan gambar 4.7, *form* laporan *bandwidth* berguna untuk menampilkan laporan grafik batang *bandwidth* yang isinya hasil *query* dari *database bandwidth*. Laporan *bandwidth* menampilkan beberapa grafik batang

untuk mempermudah perbandingan dari beberapa kriteria tertentu. Laporan grafik batang *bandwidth* menampilkan keseluruhan data yang ada didalam *database bandwidth*.

#### 4.4.8 Form Client dan Server Bandwidth Offline



Gambar 4.8 Form Client dan Server Bandwidth Offline

Penjelasan gambar 4.8, *form client dan server bandwidth offline* digunakan untuk mengambil data – data *bandwidth* dari *client* maupun *server* kemudian diproses untuk menghasilkan *upload*, *download*, *average*, *peak* dan *broadcast* secara *offline*. Cara kerja dari pengambilan data *bandwidth* dengan mengambil semua *database sniffer* yang ada di *client* atau *server*. Berbeda dengan *form bandwidth client dan server* secara *on-line*, pengambilan data di *form client dan server bandwidth offline* tidak membutuhkan komputer *client* lagi karena *database* harus berada didalam *server* dengan cara dikopi. *Form client dan server*

*bandwidth offline* menjadi alternatif jika pengumpulan data secara *on-line* menjadi sulit karena keberadaan komputer *client* tidak selalu *ready* atau berada ditempat.

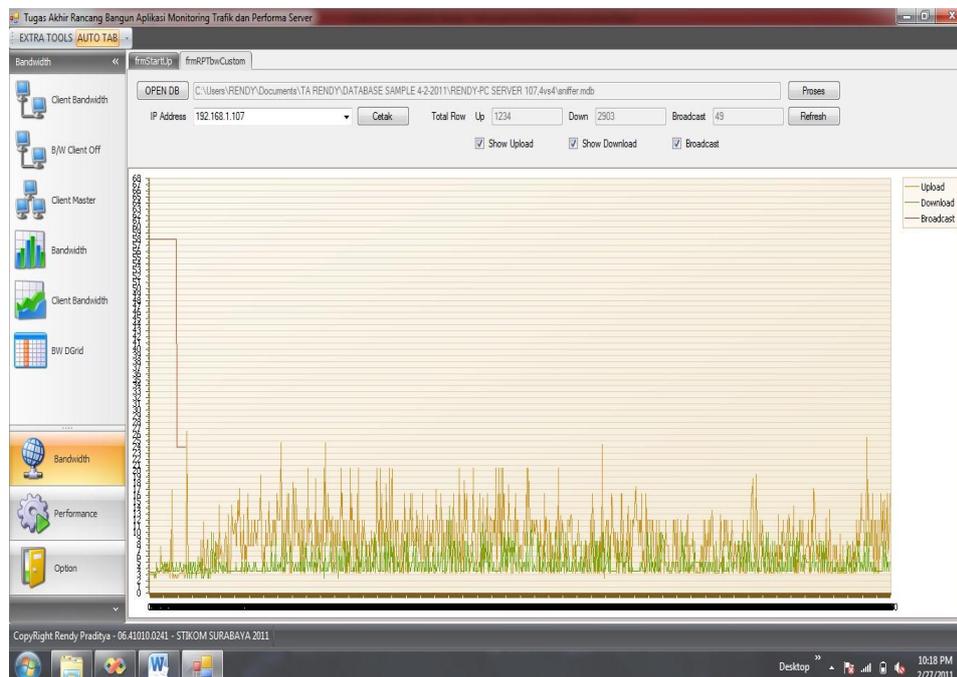
#### 4.4.9 Form Data Monitor Trafik

| computer_name | IP_Address    | upload           | download         | average          | peak | status | player | waktu        | durasi |
|---------------|---------------|------------------|------------------|------------------|------|--------|--------|--------------|--------|
| BLACBOX       | 192.168.1.58  | 3.59833333333333 | 3.59010152284264 | 3.59833333333333 | 4.75 | Client |        | 10/10/2010   | 40     |
| SERVERT-92TS  | 192.168.1.107 | 11.471469210174  | 4.33097179983412 | 11.471469210174  | 185  | Server |        | 10/7/29/2010 | 33     |
| RENDY-PC      | 192.168.1.66  | 3.53401360544218 | 3.50657894736842 | 3.53401360544218 | 4.75 | Client |        | 10/7/7/2010  | 40     |
| siegfried-PC  | 192.168.1.58  | 3.59833333333333 | 3.59010152284264 | 3.59833333333333 | 4.75 | Client |        | 10/7/30/2010 | 40     |

Gambar 4.9 Form List Data Monitor Trafik

Penjelasan gambar 4.9, *form* data monitor trafik digunakan untuk menampilkan seluruh data *bandwidth* yang tersimpan didalam database *bandwidth*. Fungsi dari *form* data monitor trafik hampir sama dengan laporan grafik tetapi menampilkan data secara detil. Data yang ditampilkan yaitu nama komputer, *IP Address*, *upload*, *download*, *average*, *peak*, *broadcast*, status, jumlah *player*, waktu dan durasi dari *database bandwidth*. Data yang ada didalam *list* tabel tidak dapat dirubah, hanya untuk ditampilkan saja. Fungsi *sorting* data dapat dilakukan di tabel data, untuk mempermudah melihat data berdasarkan kolom. Fungsi lainnya yaitu fungsi pengelompokan data berdasarkan kolom, fungsi pengelompokan data untuk mempermudah melihat data yang dikelompokan.

#### 4.4.10 Form Report Grafik Garis Bandwidth



Gambar 4.10 Form Report Grafik Garis Bandwidth

Penjelasan gambar 4.10, *form report* grafik garis *bandwidth* digunakan untuk laporan grafik garis dari setiap *database sniffer client*. Berbeda dengan *form* laporan grafik batang *bandwidth*, *form report* grafik garis *bandwidth* menunjukkan grafik naik turunnya *bandwidth* dari awal permainan sampai permainan usai. Fungsi laporan garis ini untuk melihat pola *bandwidth upload*, *download* dan *broadcast* selama permainan sehingga fluktuasi *bandwidth* terlihat. Cara untuk menampilkan grafik garis dari setiap *database* dengan menekan tombol *open database* kemudian dilanjutkan dengan memilih *IP Address* yang akan dianalisa, setelah itu menekan tombol proses sebagai eksekusi *query* dari *database sniffer*.

#### 4.4.11 Form Sniffer

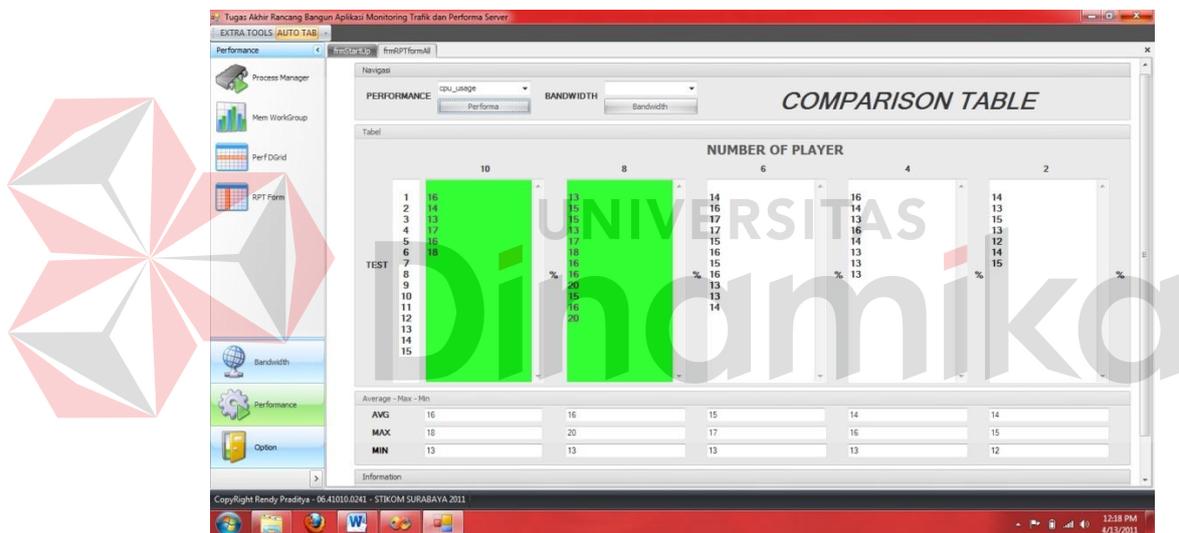
|   | Source_IP     | Destination_IP | Source_Port | Destination_Port | Packet_Byte | Protocol |
|---|---------------|----------------|-------------|------------------|-------------|----------|
| ▶ | 192.168.1.101 | 192.168.1.107  | 1660        | 139              | 20          | TCP      |
|   | 192.168.1.101 | 192.168.1.107  | 1662        | 139              | 92          | TCP      |
|   | 192.168.1.101 | 192.168.1.107  | 1662        | 139              | 157         | TCP      |
|   | 192.168.1.101 | 192.168.1.107  | 1662        | 139              | 260         | TCP      |
|   | 192.168.1.101 | 192.168.1.107  | 1662        | 139              | 372         | TCP      |
|   | 192.168.1.101 | 192.168.1.107  | 1662        | 139              | 20          | TCP      |
|   | 192.168.1.101 | 192.168.1.107  | 1662        | 139              | 20          | TCP      |
|   | 192.168.1.107 | 192.168.1.101  | 139         | 1664             | 28          | TCP      |
|   | 192.168.1.107 | 192.168.1.101  | 139         | 1664             | 24          | TCP      |

Gambar 4.11 Form Sniffer

Penjelasan gambar 4.11, *form sniffer* digunakan untuk melakukan *sniffing* paket data di setiap komputer, *client* maupun *server*. *Sniffing* paket data berguna mengkoleksi semua paket data yang melewati trafik jaringan, termasuk paket data yang bukan dari permainan Warcraft III. *Sniffing* dilakukan saat permainan berlangsung, dari awal permainan sampai berakhir. Setelah *sniffing* selesai data *sniffing* otomatis tersimpan kedalam *database sniffer* yang ada di *folder program sniffer*. Cara untuk melakukan *sniffing* dengan memilih *network interface* yang akan digunakan dalam koneksi antar komputer, kemudian memasukkan durasi, secara *default* 9999 karena selisih waktu akan dikurangi menjadi durasi waktu

permainan dalam detik, setelah mengisi durasi permainan *user* memasukkan status komputer yang akan di *sniffer* yaitu status pemain, jumlah pemain, untuk *IP Address* program *sniffer* secara otomatis mengambil dari *IP Address* komputer. Setelah semua inputan terisi diakhiri dengan menekan tombol *start*, program *sniffer* akan melakukan *sniffing* dalam jaringan. Setelah permainan berakhir, pengguna *sniffer* cukup menekan tombol *stop* maka proses *sniffing* akan berakhir dan program *sniffer* otomatis menyimpan data kedalam *database sniffer*.

#### 4.4.12 Form Laporan Tabel Perbandingan



Gambar 4.12 Form Laporan Tabel Perbandingan

Penjelasan gambar 4.12, *form* laporan tabel perbandingan digunakan untuk melakukan perbandingan performa dan trafik *server* dengan jumlah pemain yang disajikan dengan menampilkan 5 tabel jumlah pemain mulai dari dua pemain sampai sepuluh pemain. Data yang ditampilkan bersumber dari *database* performa dan trafik server. Fungsi dari laporan tabel perbandingan hampir sama dengan laporan data monitor performa dan trafik server, tetapi menampilkan data dengan memisahkan dan membandingkan data performa dan trafik server dengan jumlah

pemain. Data yang dibandingkan untuk performa server yaitu *cpu usage*, *cpu max*, *cpu min*, *physical memory*, *private memory working set*, *harddisk read* dan *harddisk write*, dan untuk trafik yaitu *upload*, *download*, *average*, *peak* dan *broadcast*. Data yang ada didalam tabel tidak dapat dirubah, hanya untuk ditampilkan dan dibandingkan saja. Cara menggunakan laporan tabel perbandingan cukup memilih variabel yang akan dibandingkan antara lain *cpu usage*, *cpu max*, *cpu min*, *physical memory*, *private memory working set*, *harddisk read* dan *harddisk write*, dan untuk trafik yaitu *upload*, *download*, *average*, *peak* dan *broadcast*, kemudian *form* laporan akan menampilkan semua data kedalam tabel pembanding menurut jumlah pemain.

#### 4.5. Uji Coba Fungsionalitas Sistem Monitor Performa dan Trafik Server

Adapun tahapan evaluasi sistem terbagi menjadi dua yaitu Evaluasi hasil uji coba sistem dan Analisis hasil uji coba sistem. Evaluasi hasil uji coba sistem dilakukan untuk mengkoscek kembali semua tahapan yang sudah dilakukan dan analisis hasil uji coba sistem bertujuan untuk menarik kesimpulan terhadap semua hasil uji coba yang dikerjakan terhadap sistem. Uji coba dilakukan dalam tahapan beberapa uji coba (*testing*) yang telah disiapkan sebelumnya.

Proses pengujian menggunakan *BlackBox Testing* dimana aplikasi akan diuji dengan melakukan berbagai percobaan untuk membuktikan apakah aplikasi yang telah dibuat sudah sesuai dengan tujuan yang akan dicapai. Aplikasi akan diuji dengan data untuk membuktikan apakah aplikasi tersebut sesuai dengan *output* yang diharapkan.

#### 4.6. Evaluasi Hasil Uji Coba Sistem

Untuk mendapatkan sistem yang sesuai dengan apa yang akan dicapai maka dilakukan beberapa uji coba. Uji coba meliputi pengujian terhadap fitur dasar aplikasi, uji coba perhitungan dan uji coba validasi pengguna terhadap pemakaian aplikasi dengan menggunakan *blackbox testing*. Uji coba yang dilaksanakan adalah sebagai berikut.

#### 4.7. Scenario Base Testing Untuk Performa Server

Proses uji coba performa yang dicoba dengan melakukan permainan dilakukan oleh 10 orang, 9 *client* dan 1 *server*. Semua komputer terkoneksi dengan *wireless* atau *switch hub* dan setiap komputer mendapatkan IP Address. Setelah terkoneksi, *server* membuat *room* permainan diikuti dengan *client* yang masuk *room* yang telah dibuat. Tujuan dari uji coba ini adalah melihat hubungan jumlah pemain dengan performa *server*.

Tabel 4.1 Uji Coba Performa Server

| Test case ID | Tujuan   | Input   | Output yang diharapkan                                      |                                   |                                    |
|--------------|--|---|---|-----------------------------------|------------------------------------|
| 1            | Mengambil sampel data performa <i>server</i> jika jumlah pemain 10 orang | 10 pemain, terdiri dari 9 <i>client</i> dan 1 <i>server</i> | Hubungan antara 10 pemain dengan performa <i>server</i>     |                                   |                                    |
|              | Hasil keluaran sistem  |   |   |                                   |                                    |
|              | range CPU usage 10 % sampai 20 %   | range physical memory usage 20 % sampai 40 %                | range memory workset 400000 Kilobyte sampai 500000 Kilobyte | range harddisk read 2 % sampai 3% | range harddisk write 2 % sampai 3% |

Tabel 4.1 Uji Coba Performa Server (Lanjutan)

| Test case ID | Tujuan  | Input  | Output yang diharapkan                                      |                                   |                                    |
|--------------|---|--|---|-----------------------------------|------------------------------------|
| 2            | Mengambil sampel data performa <i>server</i> jika jumlah pemain 8 orang | 8 pemain, terdiri dari 7 <i>client</i> dan 1 <i>server</i> | Hubungan antara 8 pemain dengan performa <i>server</i>      |                                   |                                    |
|              | Hasil keluaran sistem   |  |   |                                   |                                    |
|              | range CPU usage 10 % sampai 20 %  | range physical memory usage 20 % sampai 40 %               | range memory workset 200000 Kilobyte sampai 500000 Kilobyte | range harddisk read 1 % sampai 3% | range harddisk write 1 % sampai 2% |
| 3            | Mengambil sampel data performa <i>server</i> jika jumlah pemain 6 orang | 6 pemain, terdiri dari 5 <i>client</i> dan 1 <i>server</i> | Hubungan antara 6 pemain dengan performa <i>server</i>      |                                   |                                    |
|              | Hasil keluaran sistem   |  |   |                                   |                                    |
|              | range CPU usage 10 % sampai 20 %  | range physical memory usage 30 % sampai 40 %               | range memory workset 300000 Kilobyte sampai 400000 Kilobyte | range harddisk read 1 % sampai 3% | range harddisk write 0 % sampai 1% |
| 4            | Mengambil sampel data performa <i>server</i> jika jumlah pemain 4 orang | 4 pemain, terdiri dari 3 <i>client</i> dan 1 <i>server</i> | Hubungan antara 4 pemain dengan performa <i>server</i>      |                                   |                                    |
|              | Hasil keluaran sistem   |  |   |                                   |                                    |
|              | range CPU usage 10 % sampai 20 %  | range physical memory usage 30 % sampai 40 %               | range memory workset 300000 Kilobyte sampai 400000 Kilobyte | range harddisk read 0 % sampai 3% | range harddisk write 1 % sampai 2% |
| 5            | Mengambil sampel data performa <i>server</i> jika jumlah pemain 2 orang | 2 pemain, terdiri dari 1 <i>client</i> dan 1 <i>server</i> | Hubungan antara 2 pemain dengan performa <i>server</i>      |                                   |                                    |
|              | Hasil keluaran sistem   |  |   |                                   |                                    |
|              | range CPU usage 10 % sampai 20 %  | range physical memory usage 30 % sampai 40 %               | range memory workset 200000 Kilobyte sampai 400000 Kilobyte | range harddisk read 2 % sampai 4% | range harddisk write 1 % sampai 4% |

Tabel 4.2 Data Performa

| performance |              |               |               |               |           |        |        |             |                |               |                  |                  |
|-------------|--------------|---------------|---------------|---------------|-----------|--------|--------|-------------|----------------|---------------|------------------|------------------|
| capture_id  | capture_date | computer_name | player_number | server_status | cpu_usage | CPUmax | CPUmin | memory_phys | memory_workset | harddisk_free | HDD_read         | HDD_write        |
| IDC0001     | 7/29/2010    | RENDY-PC      | 1             | NONE          | 4         | 0      | 0      | 27          | 90839          | 43            | 0.12090235720392 | 0.15298582570202 |
| IDC0002     | 7/30/2010    | RENDY-PC      | 8             | SERVER        | 13        | 0      | 0      | 26          | 288697         | 43            | 2.23515264743923 | 1.93284235423986 |
| IDC0003     | 7/30/2010    | RENDY-PC      | 8             | SERVER        | 15        | 0      | 0      | 28          | 338195         | 43            | 1.32527567562474 | 1.34562956297023 |
| IDC0004     | 7/30/2010    | RENDY-PC      | 8             | SERVER        | 15        | 0      | 0      | 30          | 417016         | 43            | 1.82473284621712 | 1.25274567562986 |
| IDC0005     | 7/30/2010    | RENDY-PC      | 8             | SERVER        | 13        | 0      | 0      | 29          | 381929         | 43            | 2.23242756487563 | 2.14437292734514 |
| IDC0006     | 7/30/2010    | RENDY-PC      | 8             | CLIENT        | 14        | 0      | 0      | 30          | 410953         | 43            | 1.21103817461015 | 1.12998310239802 |
| IDC0007     | 7/29/2010    | RENDY-PC      | 6             | CLIENT        | 14        | 0      | 0      | 32          | 445781         | 43            | 1.32023452005782 | 1.22792346248208 |
| IDC0008     | 8/7/2010     | RENDY-PC      | 10            | CLIENT        | 13        | 0      | 0      | 27          | 347401         | 43            | 2.13019147021093 | 1.14025357235362 |
| IDC0009     | 8/7/2010     | RENDY-PC      | 10            | SERVER        | 16        | 0      | 0      | 29          | 413674         | 43            | 2.34456237558913 | 2.12386527335627 |
| IDC0010     | 8/7/2010     | RENDY-PC      | 10            | SERVER        | 14        | 0      | 0      | 31          | 476180         | 43            | 4.87254019286346 | 2.71142472147831 |
| IDC0011     | 8/7/2010     | RENDY-PC      | 10            | SERVER        | 13        | 0      | 0      | 30          | 464753         | 43            | 3.92164002394056 | 2.92141153257303 |
| IDC0012     | 8/7/2010     | RENDY-PC      | 10            | CLIENT        | 13        | 0      | 0      | 30          | 432081         | 43            | 2.72354236525031 | 1.45892456342402 |
| IDC0013     | 8/7/2010     | RENDY-PC      | 10            | CLIENT        | 14        | 0      | 0      | 31          | 446055         | 43            | 3.41329245423564 | 2.12814014384235 |
| IDC0014     | 2/28/2011    | RENDY-PC      | 4             | CLIENT        | 16        | 0      | 0      | 32          | 302885         | 43            | 1.92198235462911 | 1.18323856292032 |
| IDC0015     | 2/28/2011    | RENDY-PC      | 4             | SERVER        | 16        | 0      | 0      | 32          | 313128         | 43            | 1.12632315982426 | 1.41923610335868 |
| IDC0016     | 2/28/2011    | RENDY-PC      | 6             | SERVER        | 14        | 0      | 0      | 30          | 332038         | 43            | 1.32825263989235 | 1.85912423862902 |
| IDC0017     | 2/28/2011    | RENDY-PC      | 6             | SERVER        | 16        | 0      | 0      | 30          | 332836         | 43            | 2.23649761227412 | 1.21325696572254 |
| IDC0018     | 2/28/2011    | RENDY-PC      | 6             | SERVER        | 17        | 0      | 0      | 30          | 346016         | 43            | 1.14348014730596 | 0.23325683462325 |

Tabel 4.2 Data Performa (Lanjutan)

| performance |              |               |               |               |           |        |        |             |                |               |                   |                   |
|-------------|--------------|---------------|---------------|---------------|-----------|--------|--------|-------------|----------------|---------------|-------------------|-------------------|
| capture_id  | capture_date | computer_name | player_number | server_status | cpu_usage | CPUmax | CPUmin | memory_phys | memory_workset | harddisk_free | HDD_read          | HDD_write         |
| IDC0019     | 2/28/2011    | RENDY-PC      | 6             | SERVER        | 17        | 0      | 0      | 31          | 356313         | 43            | 1.56128126123986  | 0.32375483239754  |
| IDC0020     | 3/4/2011     | RENDY-PC      | 6             | SERVER        | 13        | 0      | 0      | 30          | 327604         | 43            | 2.17647058823529  | 1.76696832579186  |
| IDC0021     | 3/4/2011     | RENDY-PC      | 6             | SERVER        | 13        | 0      | 0      | 31          | 360796         | 63            | 1.39151321056845  | 0.99839871897518  |
| IDC0022     | 3/4/2011     | RENDY-PC      | 6             | SERVER        | 14        | 0      | 0      | 31          | 369435         | 63            | 2.17333867093675  | 0.68134576060849  |
| IDC0023     | 3/4/2011     | RENDY-PC      | 8             | SERVER        | 15        | 0      | 0      | 31          | 375019         | 63            | 1.66463898439566  | 1.48479238296747  |
| IDC0024     | 3/4/2011     | RENDY-PC      | 8             | SERVER        | 16        | 0      | 0      | 32          | 380866         | 63            | 1.46063618290258  | 2.07375745526839  |
| IDC0025     | 3/4/2011     | RENDY-PC      | 8             | SERVER        | 20        | 0      | 0      | 44          | 331789         | 64            | 10.3041841473179  | 5.26741393711449  |
| IDC0026     | 3/4/2011     | RENDY-PC      | 10            | CLIENT        | 18        | 0      | 0      | 45          | 352217         | 64            | 7.90592473979183  | 3.15772618094476  |
| IDC0027     | 3/4/2011     | RENDY-PC      | 10            | CLIENT        | 16        | 0      | 0      | 45          | 358904         | 64            | 5.68428075793969  | 2.42752420336269  |
| IDC0028     | 3/7/2011     | RENDY-PC      | 2             | SERVER        | 13        | 0      | 0      | 32          | 292073         | 42            | 3.17529215535893  | 4.51752921535893  |
| IDC0029     | 3/7/2011     | RENDY-PC      | 2             | SERVER        | 12        | 0      | 0      | 31          | 305583         | 42            | 4.24326470749269  | 2.88837567359507  |
| IDC0030     | 3/7/2011     | RENDY-PC      | 2             | SERVER        | 14        | 0      | 0      | 32          | 311152         | 42            | 2.31256133334642  | 1.74857703631011  |
| IDC0031     | 3/7/2011     | RENDY-PC      | 2             | SERVER        | 15        | 0      | 0      | 32          | 316882         | 42            | 2.77238521662404  | 2.05334307636098  |
| IDC0032     | 3/7/2011     | RENDY-PC      | 4             | SERVER        | 13        | 0      | 0      | 30          | 328311         | 42            | 3.77237851662404  | 2.74195301094567  |
| IDC0033     | 3/8/2011     | RENDY-PC      | 4             | SERVER        | 14        | 0      | 0      | 32          | 316513         | 42            | 3.24230235769131  | 1.91343747129731  |
| IDC0034     | 3/8/2011     | RENDY-PC      | 4             | SERVER        | 13        | 0      | 0      | 31          | 362812         | 40            | 0.233787029623699 | 2.47798238590873  |
| IDC0035     | 3/8/2011     | RENDY-PC      | 4             | SERVER        | 16        | 0      | 0      | 33          | 365954         | 41            | 2.39906469212783  | 1.91777084957132  |
| IDC0036     | 4/11/2011    | RENDY-PC      | 6             | SERVER        | 15        | 34     | 0      | 36          | 373267         | 34            | 0.222639149468418 | 0.723577235772358 |

Tabel 4.2 Data Performa (Lanjutan)

| performance |              |               |               |               |           |        |        |             |                |               |                   |                   |
|-------------|--------------|---------------|---------------|---------------|-----------|--------|--------|-------------|----------------|---------------|-------------------|-------------------|
| capture_id  | capture_date | computer_name | player_number | server_status | cpu_usage | CPUmax | CPUmin | memory_phys | memory_workset | harddisk_free | HDD_read          | HDD_write         |
| IDC0037     | 4/12/2011    | RENDY-PC      | 10            | SERVER        | 17        | 58     | 2      | 36          | 386794         | 34            | 0.839669834917459 | 0.852676338169085 |
| IDC0038     | 4/12/2011    | RENDY-PC      | 8             | SERVER        | 17        | 40     | 0      | 33          | 443482         | 34            | 2.32276897414512  | 0.703085904920767 |
| IDC0039     | 4/12/2011    | RENDY-PC      | 8             | SERVER        | 18        | 43     | 0      | 39          | 432782         | 34            | 0.898686679174484 | 0.801751094434021 |
| IDC0040     | 4/12/2011    | RENDY-PC      | 8             | SERVER        | 16        | 40     | 0      | 32          | 340092         | 34            | 0.570981863664791 | 0.462163852407755 |
| IDC0041     | 4/12/2011    | RENDY-PC      | 8             | SERVER        | 16        | 50     | 0      | 32          | 355801         | 34            | 0.474358974358974 | 0.506879299562226 |
| IDC0042     | 4/12/2011    | RENDY-PC      | 8             | SERVER        | 20        | 51     | 7      | 32          | 365769         | 34            | 0.339587242026266 | 0.523660621221597 |
| IDC0043     | 4/13/2011    | RENDY-PC      | 10            | SERVER        | 16        | 56     | 0      | 31          | 460682         | 34            | 0.852855356398499 | 0.703085069562546 |
| IDC0044     | 4/13/2011    | RENDY-PC      | 6             | SERVER        | 16        | 44     | 0      | 31          | 340092         | 34            | 0.879855356658313 | 0.812855356398499 |
| IDC0045     | 4/13/2011    | RENDY-PC      | 6             | SERVER        | 15        | 59     | 0      | 31          | 346212         | 34            | 0.812033435601256 | 0.709413603464533 |
| IDC0046     | 4/13/2011    | RENDY-PC      | 6             | SERVER        | 16        | 45     | 0      | 32          | 364819         | 34            | 0.709241540986543 | 0.501456010432501 |
| IDC0047     | 4/14/2011    | RENDY-PC      | 4             | SERVER        | 14        | 48     | 0      | 31          | 323592         | 34            | 0.419023511029852 | 0.610525498347923 |
| IDC0048     | 4/14/2011    | RENDY-PC      | 4             | SERVER        | 13        | 40     | 0      | 31          | 311352         | 34            | 0.525243050343466 | 0.641857549124642 |
| IDC0050     | 4/14/2011    | RENDY-PC      | 4             | SERVER        | 13        | 47     | 0      | 33          | 336371         | 34            | 0.591281439856119 | 0.712521856234566 |
| IDC0051     | 4/14/2011    | RENDY-PC      | 2             | SERVER        | 14        | 43     | 0      | 32          | 303542         | 34            | 0.412945623652395 | 0.424510142650192 |
| IDC0052     | 4/15/2011    | RENDY-PC      | 2             | SERVER        | 13        | 38     | 0      | 33          | 314634         | 34            | 0.213073463263201 | 0.323495823420216 |
| IDC0053     | 4/15/2011    | RENDY-PC      | 2             | SERVER        | 15        | 45     | 0      | 31          | 318365         | 34            | 0.253496200893457 | 0.148439251623406 |
| IDC0054     | 4/15/2011    | RENDY-PC      | 10            | SERVER        | 18        | 59     | 0      | 39          | 460682         | 34            | 0.824542356342901 | 0.552156160245634 |

#### 4.7.1 Scenario Base Testing untuk Bandwidth Server

Proses uji coba *bandwidth* yang dicoba dengan melakukan permainan dilakukan oleh 10 orang, 9 *client* dan 1 *server*. Semua komputer terkoneksi dengan *wireless* atau *switch hub* dan setiap komputer mendapatkan IP Address. Setelah terkoneksi, *server* membuat *room* permainan diikuti dengan *client* yang masuk *room* yang telah dibuat. Tujuan dari uji coba ini adalah melihat hubungan jumlah pemain dengan penggunaan *bandwidth*.

Tabel 4.3 Uji Coba Bandwidth Server

| Test case ID | Tujuan  | Input   | Output yang diharapkan              |
|--------------|---|---|-------------------------------------|
| 6            | Mengambil sampel data bandwidth jika jumlah pemain 10 orang | 10 pemain, terdiri dari 9 client dan 1 server | Kebutuhan bandwidth untuk 10 pemain |
|              | Hasil keluaran sistem                                       |   |                                     |
|              | Bandwidth range upload 9 KBps sampai 10 KBps                | range download 4 KBps sampai 5 KBps           | range average 9 KBps sampai 10 KBps |
| 7            | Mengambil sampel data bandwidth jika jumlah pemain 8 orang  | 10 pemain, terdiri dari 7 client dan 1 server | Kebutuhan bandwidth untuk 8 pemain  |
|              | Hasil keluaran sistem                                       |   |                                     |
|              | Bandwidth range upload 8 KBps sampai 9 KBps                 | range download 4 KBps sampai 5 KBps           | range average 8 KBps sampai 9 KBps  |

Tabel 4.3 Uji Coba Bandwidth Server (Lanjutan)

| Test case ID | Tujuan   | Input   | Output yang diharapkan             |                                   |                                      |
|--------------|--|---|------------------------------------|-----------------------------------|--------------------------------------|
| 8            | Mengambil sampel data bandwidth jika jumlah pemain 6 orang | 10 pemain, terdiri dari 5 client dan 1 server | Kebutuhan bandwidth untuk 6 pemain |                                   |                                      |
|              | Hasil keluaran sistem                                      |   |                                    |                                   |                                      |
|              | Bandwidth range upload 7 KBps sampai 8 KBps                | range download 4 KBps sampai 5 KBps           | range average 7 KBps sampai 8 KBps | range peak 20 KBps sampai 30 KBps | range broadcast 6 KBps sampai 7 KBps |
| 9            | Mengambil sampel data bandwidth jika jumlah pemain 4 orang | 10 pemain, terdiri dari 3 client dan 1 server | Kebutuhan bandwidth untuk 4 pemain |                                   |                                      |
|              | Hasil keluaran sistem                                      |   |                                    |                                   |                                      |
|              | Bandwidth range upload 7 KBps sampai 8 KBps                | range download 5 KBps sampai 6 KBps           | range average 7 KBps sampai 8 KBps | range peak 30 KBps sampai 70 KBps | range broadcast 6 KBps sampai 8 KBps |

Tabel 4.4 Data Bandwidth

| bandwidth     |               |                  |                  |                  |        |           |        |        |           |        |
|---------------|---------------|------------------|------------------|------------------|--------|-----------|--------|--------|-----------|--------|
| computer_name | IP_Address    | upload           | download         | average          | peak   | broadcast | status | player | waktu     | durasi |
| loupez        | 192.168.1.103 | 3.75             | 3.98091603053435 | 3.75             | 45     | 0         | Client | 10     | 2/4/2011  | 2540   |
| RENDY-PC 01   | 192.168.1.107 | 9.91626637554585 | 4.56259356287425 | 9.91626637554585 | 87.625 | 6.78      | Server | 10     | 2/4/2011  | 2526   |
| RENDY-PC 02   | 192.168.1.107 | 8.16825364667747 | 4.81811918704788 | 8.16825364667747 | 46.5   | 6.21      | Server | 8      | 2/4/2011  | 2120   |
| RENDY-PC 03   | 192.168.1.107 | 7.61091279959204 | 4.37679211469534 | 7.61091279959204 | 46.5   | 6.89      | Server | 6      | 2/4/2011  | 4400   |
| RENDY-PC 04   | 192.168.1.54  | 7.27142857142857 | 4.31481481481481 | 7.27142857142857 | 38.5   | 7.68      | Server | 6      | 2/28/2011 | 2106   |
| RENDY-PC 05   | 192.168.1.54  | 7.32692765349833 | 5.61171693735499 | 7.32692765349833 | 39     | 6.56      | Server | 4      | 2/28/2011 | 2471   |
| RENDY-PC 12   | 192.168.1.56  | 8.77689147879251 | 4.88350250461376 | 8.77689147879251 | 38.5   | 7.58      | Server | 8      | 3/4/2011  | 3391   |
| RENDY-PC 13   | 192.168.1.56  | 8.81607891061453 | 4.83191527878349 | 8.81607891061453 | 32.5   | 6.53      | Server | 8      | 3/4/2011  | 2436   |
| RENDY-PC 18   | 192.168.1.58  | 7.99520109689214 | 5.13979244114002 | 7.99520109689214 | 45.5   | 6.09      | Server | 4      | 3/8/2011  | 1926   |
| RENDY-PC 19   | 192.168.1.58  | 6.87317677946324 | 5.1049321086262  | 6.87317677946324 | 45.5   | 6.98      | Server | 4      | 3/8/2011  | 2592   |
| RENDY-PC 20   | 192.168.1.58  | 6.96470494923858 | 4.93952380952381 | 6.96470494923858 | 45.5   | 6.5       | Server | 4      | 3/8/2011  | 2261   |
| RENDY-PC 21   | 192.168.1.58  | 7.4413753298153  | 5.11312528682882 | 7.4413753298153  | 45.5   | 6.88      | Server | 4      | 3/9/2011  | 2126   |

Tabel 4.4 Data Bandwidth (Lanjutan)

| bandwidth     |               |                  |                  |                  |        |           |        |        |            |        |
|---------------|---------------|------------------|------------------|------------------|--------|-----------|--------|--------|------------|--------|
| computer_name | IP_Address    | upload           | download         | average          | peak   | broadcast | status | player | waktu      | durasi |
| tommy_pc      | 192.168.1.105 | 3.99743047158404 | 4.03178879310345 | 3.99743047158404 | 21     | 6.54      | Client | 8      | 2/4/2011   | 2930   |
| Unknown PC 02 | 192.168.1.67  | 7.71743295019157 | 4.85411651420318 | 7.71743295019157 | 26.125 | 6.73      | Server | 6      | 11/26/2010 | 1955   |
| Unknown PC 04 | 192.168.1.81  | 10.8650862068966 | 4.26581348592638 | 10.8650862068966 | 82.625 | 7.25      | Server | 10     | 8/28/2010  | 2875   |
| Unknown PC 05 | 192.168.1.82  | 10.4859302325581 | 4.26156838656839 | 10.4859302325581 | 91.125 | 5.5       | Server | 10     | 8/28/2010  | 2177   |
| Unknown PC 06 | 192.168.1.83  | 3.5              | 3.56818181818182 | 3.5              | 26.125 | 7.54      | Client | 10     | 8/28/2010  | 2240   |
| Unknown PC 07 | 192.168.1.49  | 4.3010752688172  | 4.22993119266055 | 4.3010752688172  | 27.125 | 9.06      | Client | 10     | 8/13/2010  | 2721   |
| Unknown PC 08 | 192.168.1.52  | 3.77325581395349 | 3.77020426287744 | 3.77325581395349 | 27.125 | 7.89      | Client | 10     | 8/13/2010  | 2821   |
| Unknown PC 09 | 192.168.1.54  | 3.74939024390244 | 3.59191919191919 | 3.74939024390244 | 27.125 | 8.33      | Client | 10     | 8/13/2010  | 2452   |
| Unknown PC 10 | 192.168.1.56  | 3.55851063829787 | 3.50555555555556 | 3.55851063829787 | 22.75  | 8.54      | Client | 10     | 8/14/2010  | 2760   |
| Unknown PC 11 | 192.168.1.58  | 3.59833333333333 | 3.59010152284264 | 3.59833333333333 | 27.125 | 7.66      | Client | 10     | 10/29/2010 | 2421   |

#### 4.8. Evaluasi Sistem

Performa sistem yang akan dievaluasi, berdasarkan pada indikator performa sistem yang telah dijabarkan di subbab 2.12, mencakup CPU, *physical memory*, *harddisk* dan lalu lintas jaringan.

##### 4.8.1 CPU, PHYSICAL MEMORY, HARDDISK

Performa CPU diukur dengan menangkap penggunaan prosesor dari *server* yang sama untuk beberapa skenario pemain, pengambilan data dilakukan selama permainan berlangsung perdetik dan diambil rata-ratanya.

Tabel 4.5 Performa CPU Server

| Uji Coba | Performa CPU (%) |          |          |          |          |
|----------|------------------|----------|----------|----------|----------|
|          | 10 Pemain        | 8 Pemain | 6 Pemain | 4 Pemain | 2 Pemain |
| 1        | 16               | 13       | 14       | 13       | 15       |
| 2        | 14               | 15       | 16       | 14       | 14       |
| 3        | 13               | 15       | 17       | 16       | 12       |

Berdasarkan pada tabel 4.5, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa performa CPU berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 13% sampai dengan 16%. Hal ini membuktikan bahwa performa CPU tidak dipengaruhi oleh jumlah pemain.

Tabel 4.6 Performa Physical Memory Server

| Uji Coba | Performa Physical Memory (%) |          |          |          |          |
|----------|------------------------------|----------|----------|----------|----------|
|          | 10 Pemain                    | 8 Pemain | 6 Pemain | 4 Pemain | 2 Pemain |
| 1        | 30                           | 26       | 31       | 32       | 32       |
| 2        | 31                           | 28       | 31       | 32       | 31       |
| 3        | 29                           | 30       | 30       | 30       | 32       |

Berdasarkan pada tabel 4.6, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa penggunaan *phisycal memory* berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 26% sampai dengan 32%. Hal ini membuktikan bahwa penggunaan *physical memory* tidak dipengaruhi oleh jumlah pemain.

Tabel 4.7 Performa Warcraft III Private Memory Workset

| Uji Coba | Performa Warcraft III Private Memory Workset (KB) |          |          |          |          |
|----------|---|----------|----------|----------|----------|
|          | 10 Pemain   | 8 Pemain | 6 Pemain | 4 Pemain | 2 Pemain |
| 1        | 413674  | 288697   | 332038   | 313128   | 292073   |
| 2        | 476180  | 338195   | 332836   | 316513   | 305583   |
| 3        | 464753  | 417016   | 346016   | 328311   | 311152   |

Berdasarkan pada tabel 4.7, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa penggunaan *private memory workset* berdasarkan jumlah pemain terdapat sedikit perbedaan penggunaan *private memory workset*, yaitu berkisar antara 288697 KB sampai dengan 464753 KB. Hal ini membuktikan bahwa penggunaan *private memory workset* dipengaruhi oleh jumlah pemain.

Tabel 4.8 Performa Harddisk Read Server

| Uji Coba | Performa Harddisk Read Server (%) |          |          |          |          |
|----------|-----------------------------------|----------|----------|----------|----------|
|          | 10 Pemain                         | 8 Pemain | 6 Pemain | 4 Pemain | 2 Pemain |
| 1        | 2.34                              | 2.12     | 1.32     | 1.12     | 3.17     |
| 2        | 4.87                              | 2.71     | 2.23     | 3.24     | 4.24     |
| 3        | 3.92                              | 2.92     | 1.14     | 3.77     | 2.31     |

Berdasarkan pada tabel 4.8, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa perfoma *harddisk read* berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 1.12%

sampai dengan 4.87%. Hal ini membuktikan bahwa performa *harddisk read* tidak dipengaruhi oleh jumlah pemain.

Tabel 4.9 Performa Harddisk Write Server

| Uji Coba | Performa Harddisk Write Server (%) |          |          |          |          |
|----------|------------------------------------|----------|----------|----------|----------|
|          | 10 Pemain                          | 8 Pemain | 6 Pemain | 4 Pemain | 2 Pemain |
| 1        | 2.12                               | 1.93     | 1.85     | 1.41     | 4.51     |
| 2        | 2.71                               | 1.34     | 1.21     | 1.91     | 2.88     |
| 3        | 2.92                               | 1.25     | 0.23     | 2.74     | 1.74     |

Berdasarkan pada tabel 4.9, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa performa *harddisk write* berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 0.23% sampai dengan 2.92%. Hal ini membuktikan bahwa performa *harddisk write* tidak dipengaruhi oleh jumlah pemain.

Tabel 4.10 Performa CPU Max

| Uji Coba | Performa CPU Max (%) |          |          |          |          |
|----------|----------------------|----------|----------|----------|----------|
|          | 10 Pemain            | 8 Pemain | 6 Pemain | 4 Pemain | 2 Pemain |
| 1        | 58                   | 40       | 44       | 48       | 43       |
| 2        | 56                   | 50       | 59       | 40       | 38       |
| 3        | 59                   | 51       | 45       | 47       | 45       |

Berdasarkan pada tabel 4.10, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa performa *cpu max* berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 38% sampai dengan 59%. Hal ini membuktikan bahwa performa *cpu max* tidak dipengaruhi oleh jumlah pemain.

Tabel 4.11 Performa CPU Min

| Uji Coba | Performa CPU Min (%) |          |          |          |          |
|----------|----------------------|----------|----------|----------|----------|
|          | 10 Pemain            | 8 Pemain | 6 Pemain | 4 Pemain | 2 Pemain |
| 1        | 2                    | 0        | 0        | 0        | 0        |
| 2        | 0                    | 0        | 0        | 0        | 0        |
| 3        | 0                    | 7        | 0        | 0        | 0        |

Berdasarkan pada tabel 4.11, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa performa *cpu min* berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 0% sampai dengan 7%. Hal ini membuktikan bahwa performa *cpu min* tidak dipengaruhi oleh jumlah pemain.

#### 4.8.2 BANDWIDTH

Tabel 4.12 Kebutuhan Bandwidth Upload Server

| Uji Coba | Kebutuhan Bandwidth Upload Server (KBps) |          |          |          |
|----------|--|----------|----------|----------|
|          | 10 Pemain                                | 8 Pemain | 6 Pemain | 4 Pemain |
| 1        | 9.91                                     | 8.16     | 7.61     | 7.32     |
| 2        | 10.86                                    | 8.77     | 7.71     | 7.99     |
| 3        | 10.48                                    | 8.81     | 7.27     | 6.87     |

Berdasarkan pada tabel 4.12, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa penggunaan *bandwidth upload* berdasarkan jumlah pemain memiliki perbedaan, yaitu berkisar antara 6.87 KBps sampai dengan 10.48 KBps. Dari setiap jumlah pemain ada peningkatan *bandwidth upload* sebesar 1 KBps. Hal ini membuktikan bahwa penggunaan *bandwidth upload* dipengaruhi oleh jumlah pemain.

Tabel 4.13 Kebutuhan Bandwidth Download Server

| Uji Coba | Kebutuhan Bandwidth Download Server (KBps) |          |          |          |
|----------|--|----------|----------|----------|
|          | 10 Pemain                                  | 8 Pemain | 6 Pemain | 4 Pemain |
| 1        | 4.56                                       | 4.81     | 4.37     | 5.61     |
| 2        | 4.26                                       | 4.88     | 4.85     | 5.13     |
| 3        | 4.26                                       | 4.83     | 4.31     | 5.10     |

Berdasarkan pada tabel 4.13, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa penggunaan *bandwidth download* berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 4.26 KBps sampai dengan 5.61 KBps. Hal ini membuktikan bahwa penggunaan *bandwidth download* tidak dipengaruhi oleh jumlah pemain.

Tabel 4.14 Kebutuhan Bandwidth Average Server

| Uji Coba | Kebutuhan Bandwidth Average Server (KBps) |          |          |          |
|----------|---|----------|----------|----------|
|          | 10 Pemain                                 | 8 Pemain | 6 Pemain | 4 Pemain |
| 1        | 9.91                                      | 8.16     | 7.61     | 7.32     |
| 2        | 10.86                                     | 8.77     | 7.71     | 7.99     |
| 3        | 10.48                                     | 8.81     | 7.27     | 6.87     |

Berdasarkan pada tabel 4.14, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa penggunaan *bandwidth average* berdasarkan jumlah pemain memiliki perbedaan, yaitu berkisar antara 6.87 KBps sampai dengan 10.48 KBps. Dari setiap jumlah pemain ada peningkatan *bandwidth average* sebesar 1 KBps. Hal ini membuktikan bahwa penggunaan *bandwidth average* dipengaruhi oleh jumlah pemain.

Tabel 4.15 Kebutuhan Bandwidth Peak Server

| Uji Coba | Kebutuhan Bandwidth Peak Server (KBps) |          |          |          |
|----------|--|----------|----------|----------|
|          | 10 Pemain                              | 8 Pemain | 6 Pemain | 4 Pemain |
| 1        | 87.625                                 | 46.5     | 46.5     | 39       |
| 2        | 82.625                                 | 38.5     | 38.5     | 45.5     |
| 3        | 91.125                                 | 32.5     | 38.5     | 45.5     |

Berdasarkan pada tabel 4.15, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa penggunaan *bandwidth peak* berdasarkan jumlah pemain memiliki perbedaan, yaitu berkisar antara 32.5 KBps sampai dengan 91.125 KBps. Dari setiap jumlah pemain ada peningkatan *bandwidth peak*. Hal ini membuktikan bahwa penggunaan *bandwidth peak* dipengaruhi oleh jumlah pemain.

Tabel 4.16 Kebutuhan Bandwidth Broadcast Server

| Uji Coba | Kebutuhan Bandwidth Broadcast Server (KBps) |          |          |          |
|----------|---|----------|----------|----------|
|          | 10 Pemain                                   | 8 Pemain | 6 Pemain | 4 Pemain |
| 1        | 6.78  | 6.21     | 6.89     | 6.56     |
| 2        | 7.25  | 6.54     | 6.73     | 6.09     |
| 3        | 5.5   | 7.58     | 7.68     | 6.98     |

Berdasarkan pada tabel 4.16, hasil 3 kali uji coba yang dilakukan untuk tiap jumlah pemain memperlihatkan bahwa penggunaan *bandwidth broadcast* berdasarkan jumlah pemain tidak memiliki perbedaan yang cukup berarti, yaitu berkisar antara 6.09 KBps sampai dengan 7.89 KBps. Hal ini membuktikan bahwa penggunaan *bandwidth broadcast* tidak dipengaruhi oleh jumlah pemain.

Analisa dari percobaan untuk performa *server* dan kebutuhan *bandwidth*, didapatkan kesimpulan sebagai berikut :

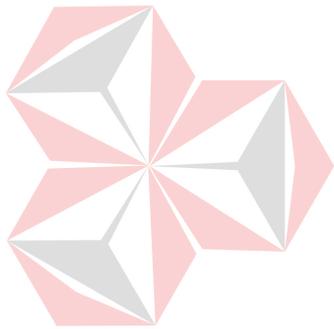
1. Performa CPU dan *memory physical server* dan *client* memiliki bobot *resource* yang sama karena saat permainan berjalan, *client* dan *server* sama-sama melakukan proses *rendering* atau sinkronisasi permainan. Performa *server* untuk *memory working set* terdapat hubungan dengan jumlah pemain. Permainan dengan jumlah pemain yang berbeda akan menggunakan

*resource* yang berbeda. Semakin sedikit jumlah pemain yang bergabung maka *memory working set* yang digunakan *server* juga semakin sedikit. Dengan demikian untuk mendapatkan performa yang baik, *server* harus memiliki *memory physical* yang besar. Performa *client* yang rendah akan mengganggu jalannya permainan, ini disebut dengan *bottleneck*, karena saat permainan dijalankan semua komputer harus melakukan sinkronisasi permainan yang sama termasuk *server*. Jika *client* atau *server* mendapat masalah dengan performanya maka sinkronisasi permainan akan terganggu sehingga membuat *delay* dan *client* lainnya akan menunggu sampai *client* atau *server* yang bermasalah tersebut stabil. Tetapi jika *client* bermasalah tidak kunjung stabil maka *client* yang lain termasuk *server* memiliki hak untuk melakukan *kick* atau akan keluar secara otomatis jika *timeout* tercapai.

2. *Bandwidth client* dan *server* berbeda, *server* menggunakan *bandwidth* yang lebih besar karena *server* menjadi pusat dari lalu lintas trafik permainan.

*Bandwidth server* dipengaruhi dengan banyaknya jumlah pemain, semakin banyak jumlah pemain maka *bandwidth server* akan semakin besar. Dari hasil penelitian dan pengumpulan data jumlah pemain 10 komputer sampai dengan 4 komputer terdapat perbedaan *bandwidth* yang melalui *server*. *Bandwidth* yang tidak signifikan berbeda dengan banyaknya jumlah pemain adalah *download* dan *broadcast* sedangkan *upload*, *average* dan *peak* menunjukkan perbedaan yang signifikan karena jumlah pemain akan menambah trafik *bandwidth upload*, *average* dan *peak* yang digunakan. Perbedaan yang signifikan ini dikarenakan setiap *client* memberikan sinyal kendali dan informasi terkait ke *client* lainnya termasuk *server*. *Bandwidth*

*broadcast* antar *client* dan *server* hampir sama karena *broadcast* adalah *listener* atau komunikasi sesama komputer, sedangkan *bandwidth broadcast* tidak dipengaruhi oleh banyaknya pemain.



UNIVERSITAS  
**Dinamika**

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Setelah dilakukan analisis, perancangan sistem dan pembuatan aplikasi monitor trafik dan performa *server* permainan komputer (Studi Kasus Warcraft III) ini serta dilakukan evaluasi hasil penelitiannya, maka dapat diambil kesimpulan sebagai berikut:

1. Telah dibuat aplikasi yang dapat melakukan *sniffer packet data* sehingga performa *network* dapat dimonitor.
2. Aplikasi telah dapat digunakan untuk mengidentifikasi kebutuhan performa *server* dalam permainan warcraft III. Dari hasil indentifikasi kebutuhan performa *server* yang terdiri dari CPU, *memory*, *harddisk* dan trafik didapatkan hasil bahwa kebutuhan utama performa *server* yang paling dipengaruhi oleh banyaknya pemain adalah trafik.

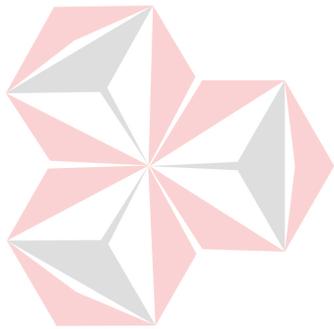
#### 5.2. Saran

Adapun beberapa saran yang dapat disampaikan untuk mengembangkan aplikasi yang telah dibuat antara lain:

1. Aplikasi Monitor Trafik dan Performa Permainan Komputer (Studi Kasus Warcraft III) ini dapat dikembangkan sehingga dapat menambah variabel pembanding baru, sebagai contoh monitor performa dapat diintegrasikan dengan *database hardware* komputer dari berbagai macam merk dagang yang

didalamnya memasukkan L2 *cache*, L3 *chace*, fitur *turbo*, fitur *hypertreading*, fitur *hypertransport*, *memory clock latency*, *memory type* (DDR 2 atau DDR 3) dan VGA (*Video graphic Array*). Dengan begitu akan memberikan data yang lebih akurat tentang trafik dan performa.

2. Aplikasi Monitor Trafik dan Performa Permainan Komputer (Studi Kasus Warcraft III) ini dapat diimplementasikan dengan banyak permainan komputer, sehingga kasus yang dapat digunakan masih sangat luas.



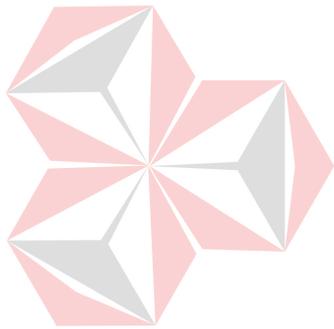
UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

- Adams, Dan. 2006. The State of the RTS, (Online), (<http://pc.ign.com/articles/700/700747p1.html>, diakses 18 April 2010).
- Clarke, Glen E. 2009. CompTIA Network+ Certification Study Guide, Fourth Edition. New York: Mcgraw-Hill.
- Dewo, E. S. 2003. Bandwidth dan Throughput. *Artikel Populer Ilmu Komputer.com*: 1-3.
- Dharwiyanti, Sri dan Wahono, Romi Satria. 2003. *Pengantar Modified Modeling Language (UML)*, (Online), (<http://www.ilmukomputer.com/umum/yanti-uml.php>), diakses tanggal 12 Januari 2010.
- Foust, Faust; Chellis, James; Sheltz, Matthew; London, Suzan Sage. 2003. MCSE Windows Server 2003 network infrastructure planning and maintenance. Indianapolis: Wiley Publishing.
- Harry. 2005. Penggunaan Snoop(Sniffer) Dalam Network Kita, (Online), (<http://hr2009.blogspot.com/2005/10/penggunaan-snoop-sniffer-dalam-network.html>, diakses 11 Januari 2007).
- Irawan, Jusak & Sukmaaji, Anjik. 2003. *Manajemen Jaringan Komputer (Edisi Pertama)*. Surabaya: Stikom.
- Jones, Don. 2004. The Definitive Guide to Active Directory Troubleshooting and Auditing. Phoenix: Realtimepubliser.com
- Purbo, O. W. 2003. Memahami TCP Header. *Majalah Neotek, Vol. II No. 9. 2002, Edisi Bulan Juni* : 29-31.
- Purbo, O. W. & Taufan, R. 2001. *Manajemen Jaringan TCP/IP*. Jakarta: PT. Elex Media Komputindo.
- Rajani, Renu. 2005. Software Testing. New Delhi : Tata McGraw-Hill Publising Company Limited.
- Riantory, Ragowo. 2008. Pembuatan Aplikasi Sniffer Untuk Analisa kerja Pada Sebuah Jaringan. Surabaya : Stikom.
- Sukmaaji, Anjik. 2003. Jaringan Komputer. Surabaya: Stikom.
- Suryadi. 1997. *TCP/IP dan Internet Sebagai Jaringan Komunikasi Global*. Jakarta: PT. Elex Media Komputindo.

Prayudi. 2005. Computer Games Research Group, (Online), (<http://cgrg.wordpress.com/category/teori-game-computer/>, diakses 13 Maret 2008).

Tanenbaum, Andrew S. 1996. Jaringan Komputer, Jilid Satu. Jakarta: Prenhallindo.



UNIVERSITAS  
**Dinamika**