

APLIKASI BILLING WARNET BERBASIS CLIENT SERVER
MENGUNAKAN VISUAL BASIC 2005

Laporan Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Persyaratan
Menyelesaikan Program Studi Strata Satu (S1)
Program Studi Teknik Informatika

Oleh :

Chandra Aziz Setyo

(01503-047)



JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA

2009

LEMBAR PENGESAHAN

Yang bertanda tangan dibawah ini menyatakan bahwa laporan Tugas akhir dari mahasiswa berikut ini :

Nama : Chandra Aziz Setyo
NIM : 01503-047
Jurusan : Teknik Informatika
Fakultas : Ilmu Komputer
Judul : “Aplikasi Billing Warnet Berbasis Client Server Menggunakan Visual Basic 2005“.

Telah diperiksa dan disetujui sebagai Laporan Tugas Akhir

Disetujui Oleh :

(Ahmad Kodar, ST., MT)
Dosen Pembimbing I

(Andrew Fiade, ST.,M.Kom)
Dosen Pembimbing II

Mengesahkan,

Menyetujui

(Abdusy Syarif, ST, MT)
Ketua Program Studi
Teknik Informatika - UMB

(Devi Fitrihanah S.Kom, MTI)
Koordinator Tugas Akhir
Teknik Informatika

ABSTRAK

Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 dibuat untuk memudahkan proses monitoring komputer client yang terhubung di dalam sebuah jaringan Local Area Network (LAN) dan biasanya digunakan di warung internet. Proses monitoring dapat dilakukan langsung dari komputer server.

Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 dapat berfungsi sebagai aplikasi pengolahan data. Proses pengolahan data tersebut dapat dilakukan dengan mudah dan cepat karena Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 memiliki salah satu kelebihan yaitu tampilan yang *user friendly*.

Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 menggunakan MySQL sebagai database server-nya. Sehingga data-data administrator, operator, anggota dan nonanggota yang berkaitan dengan warung internet dapat terjaga keamanannya, terintegrasi, serta dapat dengan mudah diolah dan diakses.

Dengan adanya sistem billing, user juga dapat dengan mudah mengatur serta mengkonfigurasi sendiri jumlah biaya atau jumlah waktu yang ingin mereka gunakan dalam mengakses internet. Hal ini mungkin karena Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 menyediakan fasilitas yang dapat digunakan oleh user untuk mengatur dan mengkonfigurasi jumlah biaya serta jumlah waktu pemakaian internet.

Kata kunci : Implementasi Billing warnet client server, Socket Programming, VB 2005, MySql 5.0

ABSTRACT

Internet Billing Based On Client Server Using Visual Basic 2005 is built to facilitate and make easy the monitoring process of client computers attached to a Local Area Network (LAN) and it is usually used in internet rental store. Monitoring process can be done directly from a computer server.

Internet Billing Based On Client Server Using Visual Basic 2005 can function as data processing application. The Data Processing can be done easier and faster because all the existing data are integrated at the server. One of other excess of this application is that it has a user friendly interface.

Internet Billing Based On Client Server Using Visual Basic 2005 using MySQL as the database server. Therefore the data related to administrator, operator, member and nonmember of internet rental store can be protected, integrated, managed and accessed easily.

With billing system, user can arrange and configure the price and the duration they want to use in accessing the internet. It is possible because Internet Billing Based On Client Server Using Visual Basic 2005 provides facility that can be used by user in arranging the price or the time for internet access.

Key words : *Implementation Billing internet of client server, Socket Programming, VB 2005, MySql 5.0*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT yang telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan penyusunan laporan tugas akhir ini untuk diajukan sebagai salah satu syarat untuk pengambilan tugas akhir guna memperoleh gelar Sarjana Strata Satu (S-1) pada Program Studi Teknik Informatika Universitas Mercu Buana.

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada pihak yang telah membantu sehingga penulisan Tugas akhir ini dapat diselesaikan, ucapan terima kasih dan penghargaan yang sebesar-besarnya penulis sampaikan kepada :

1. Kedua Orang tua tercinta, yang telah memberikan dukungan baik moril maupun materil.
2. Bapak Ahmad Kodar, ST., MT sebagai dosen pembimbing I yang telah membimbing penulis selama masa perkuliahan dan tugas akhir.
3. Bapak Anderw Fiade S.T., M.Kom sebagai dosen pembimbing II yang telah meluangkan waktunya dan membimbing penulis dengan sabar dan untuk sumber referensi Pressman yang luar biasa.
4. Bapak Abdusy Syarif, S.Kom., MT sebagai Kepala Jurusan Teknik Informatika yang telah membantu proses terlaksananya tugas akhir ini.
5. Teman-teman seperjuangan Team Mobo (Iip, Sahal, Abe, Ivan, Igit, Boand, Rofi, Zaenal dan Bowe), yang selalu memberikan motivasi dan dukungan kepada penulis. ☺ ☺

6. Fajar Melati teman terbaik diwaktu SMU yang memberikan dukungan dan semangat untuk dapat menyelesaikan Studi S1 dengan proses yang baik. ☺
7. Saudara, teman-teman angkatan 2003 dan semua pihak yang namanya tidak dapat disebutkan satu persatu, baik secara langsung maupun tidak langsung telah berperan serta dalam mewujudkan laporan Tugas akhir ini.

Akhir kata dengan segala kerendahan hati penulis memohon maaf yang sebesar-besarnya atas kekurangan dan keterbatasan yang terdapat dalam laporan Tugas akhir ini dan untuk itu semua saran dan kritik yang sifatnya membangun sangat diharapkan demi kesempurnaan laporan Tugas akhir ini serta besar harapan penulis semoga laporan Tugas akhir ini dapat bermanfaat bagi pihak-pihak yang membutuhkan.

Jakarta, Januari 2009

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	i
ABSTRAK	ii
ABSTRACT	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penulis	2
1.4 Batasan masalah	3
1.5 Ruang Lingkup dan Batasan Masalah	3
1.6 Metode Penulisan	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	
2.1 Pengertian Basis Data	6
2.1.1 Kamus Data	7
2.1.2 <i>Entity Relationship (ER)</i>	7
2.1.2.1 Komponen Pembentuk Suatu <i>Entity Relationship</i>	7
2.1.2.2 Derajat Kardinalitas dari Relasi	8
2.1.3 <i>Entity Relationship Diagram</i>	9
2.1.3.1 Tahapan Pembuatan <i>Entity Relationship Diagram</i>	9
2.1.4 <i>Data Flow Diagram</i>	11
2.1.4.1 Proses	14
2.1.4.2 Aliran Data (<i>Data Flow</i>)	14
2.1.4.3 Penyimpanan Data (<i>Data Store</i>)	15
2.1.4.4 Entitas	15

2.2 Model <i>Waterfall</i>	15
2.3 Jaringan Komputer	17
2.3.1 TCP / IP	19
2.3.2 TCP Protocol	19
2.3.3 <i>Internet Protocol (IP)</i>	19
2.3.4 <i>Port Number</i>	20
2.4 <i>Socket Programming</i>	20
2.5 Pengenalan MySql	21
2.5.1 Memulai Dengan MySql	22
2.5.2 Perintah Query Pada MySql	23
2.5.2.1 <i>Statement Create Table</i>	23
2.5.2.2 <i>Statement Select From</i>	23
2.5.2.3 <i>Statement Insert Into</i>	24
2.5.2.4 <i>Statement Update</i>	24
2.6 Pengenal Visual Studio 2005	25
2.6.1 .Net Framework	25
2.6.2 <i>Socket Programming pada Visual Basic 2005</i>	26
2.6.2.1 <i>Bagaimana Socket dapat digunakan</i>	27

BAB III ANALISA DAN PERANCANGAN

3.1 Analisa	31
3.1.1 Analisa Masalah	32
3.1.2 Analisa Sistem Berjalan	32
3.1.2.1 Deskripsi Prosedur Kerja	32
3.1.2.2 Identifikasi Kebutuhan Pemakai	33
3.1.3 Analisa Kebutuhan Sistem	34
3.1.3.1 Kebutuhan Informasi	34
3.1.3.2 Kebutuhan Aplikasi	34
3.1.3.3 Kebutuhan Perangkat Keras	35
3.1.4 Analisa Kebutuhan Perangkat Lunak	35
3.1.4.1 Deskripsi Kebutuhan Fungsional	35
3.1.4.2 Pemodelan Desain	36

3.1.4.3 Deskripsi Tabel	42
3.2 Perancangan Sistem Informasi	46
3.2.1 Perancangan Prosedur	46
3.2.2 Perancangan Perangkat Keras	49
3.2.2.1 Konfigurasi Perangkat Keras	49
3.2.2.2 Spesifikasi Perangkat Keras	50
3.2.2.3 Spesifikasi Perangkat Lunak Sistem	50
3.2.2.4 Perancangan Perangkat Lunak	51
BAB IV IMPLEMENTASI DAN PENGUJIAN	
4.1 Lingkungan Implementasi	59
4.2 Konfigurasi <i>Client Server</i>	61
4.3 Implementasi Proses	63
4.3.1 Proses Koneksi Basis Data	63
4.3.2 Proses Perhitungan Harga dan Waktu	64
4.3.3 Proses Koneksi Client Server	65
4.3.4 Proses Pengiriman Data	67
4.4 Sistem Interface	72
4.5 Pengujian Sistem	77
4.5.1 Lingkup Pengujian	77
4.5.2 Data Hasil Pengujian	80
4.6 Analisa Hasil Pengujian	82
BAB VI PENUTUP	
6.1 Kesimpulan	84
6.2 Saran	85
DAFTAR PUSTAKA	
LAMPIRAN	
LAMPIRAN PROGRAM	

DAFTAR GAMBAR

	Halaman
Gambar 2.1. : Diagram <i>E-R</i>	10
Gambar 2.2. : <i>Schema Digaram</i>	11
Gambar 2.3. : Diagram kontek aplikasi pemutahiran data penduduk	12
Gambar 2.4. : DFD Level 1 Aplikasi pemutahiran data penduduk.....	13
Gambar 2.5. : Simbol Proses.....	14
Gambar 2.6 : Simbol Aliran Data	14
Gambar 2.7. : Simbol <i>Data Store</i>	15
Gambar 2.8. : Simbol Entitas	15
Gambar 2.9. : <i>Source Code</i> aplikasi <i>console</i> pada <i>client</i>	28
Gambar 2.10. : <i>Source Code</i> aplikasi <i>console</i> pada <i>server</i>	29
Gambar 2.11. : Tampilan aplikasi <i>console client</i>	29
Gambar 2.12. : Tampilan aplikasi <i>console server</i>	30
Gambar 3.1. : Flowchar Sistem Berjalan	33
Gambar 3.2. : Digram Kontek Aplikasi Billing Internet <i>Client Server</i>	37
Gambar 3.3. : DAD Tingkat 1 Pada aplikasi Billing Internet <i>Client Server</i>	38
Gambar 3.4. : DAD Tingkat 2 Pada aplikasi Billing Internet <i>Client Server</i>	39
Gambar 3.5. : DAD Tingkat 3 Pada aplikasi Billing Internet <i>Client Server</i>	40
Gambar 3.6. : ER Diagram	45
Gambar 3.7. : Deskripsi Prosedur Kerja Baru	48
Gambar 3.8. : Konfigurasi Perangkat Keras	49
Gambar 3.9 : Menu Login (<i>client</i>)	51
Gambar 3.10. : Rancangan Menu Login (<i>server</i>)	52
Gambar 3.11. : Rancangan Status Pemakaian (<i>client</i>).....	52
Gambar 3.12. : Struktur Menu	53
Gambar 3.13. : Struktur Menu (lanjutan 1)	54
Gambar 3.14. : Struktur Menu (lanjutan 2).....	54
Gambar 3.15. : Struktur Menu (lanjutan 3).....	54
Gambar 3.16. : Rancangan Menu Utama	55
Gambar 3.17. : Rancangan konfigurasi harga warnet	56

Gambar 3.18.	: Rancangan Laporan Pemakaian harian	57
Gambar 4.1.	: Skema Implementasi pengguna pada Billing Internet	61
Gambar 4.2.	: Konfirmasi login pada <i>Client</i>	62
Gambar 4.3.	: Ganti nama komputer <i>Client</i>	62
Gambar 4.4.	: <i>Source code Class</i> yang berobjek <i>Classconnect</i>	63
Gambar 4.5.	: <i>Source Code</i> membaca data harga dalam database	64
Gambar 4.6	: Query stored prosedur <i>sp_harga</i>	65
Gambar 4.7.	: <i>Source Code Class</i> waktu pemakaian Billing	66
Gambar 4.8.	: <i>Source Code</i> Koneksi pada <i>Client</i>	66
Gambar 4.9	: <i>Source Code</i> Koneksi pada <i>Server</i>	67
Gambar 4.10.	: Source Code Pengiriman data Pemakaian Internet	68
Gambar 4.11.	: Source Code Penerimaan pemakaian data pada server	69
Gambar 4.12.	: Source Code fungsi pengiriman data pada server	69
Gambar 4.13.	: Flowchart pengiriman data pada server (<i>remote</i>)	70
Gambar 4.14.	: Source Code fungsi penerimaan data pada client	71
Gambar 4.15.	: Flowchart Penerimaan data pada client (<i>remote</i>).....	72
Gambar 4.16.	: Tampilan Login (<i>Server</i>).....	72
Gambar 4.17.	: Tampilan Login (<i>Client</i>)	73
Gambar 4.18	: <i>Tampilan Proses (Client)</i>	73
Gambar 4.19.	: Tampilan Menu Utama pada aplikasi dengan konsep MDI	74
Gambar 4.20.	: Tampilan Set harga pengguna	76
Gambar 4.21.	: Tampilan Laporan data pemakaian harian	77

DAFTAR TABEL

	Halaman
Tabel 3.1. : Analisa Kebutuhan Sistem	34
Tabel 3.2. : Analisa Kebutuhan Perangkat Lunak <i>Server</i>	35
Tabel 3.3. : Analisa Kebutuhan Perangkat Lunak <i>Client</i>	36
Tabel 3.4. : Tabel t_harga	42
Tabel 3.5. : Tabel t_anggota	43
Tabel 3.6. : Tabel t_users	43
Tabel 3.7. : Tabel t_ip	43
Tabel 3.8. : Tabel t_biaya_pemakaian	44
Tabel 3.9. : Tabel t_hrgblnanggota	44
Tabel 3.10. : Tabel temp	45
Tabel 3.11. : Spesifikasi Perangkat Lunak Sistem	50
Tabel 4.1. : Lingkungan perangkat keras	59
Tabel 4.2. : Lingkungan perangkat lunak	59
Tabel 4.3. : Modul perangkat lunak pada <i>server</i>	78
Tabel 4.4. : Modul perangkat lunak pada <i>client</i>	78
Tabel 4.5. : Pengujian reliabilitas perangkat lunak	79
Tabel 4.6. : Pengujian modul – modul	80
Tabel 4.7. : Reliabilitas perangkat lunak	81
Tabel 4.8. : Pengujian Penghitungan biaya	83

BAB I

PENDAHULUAN

1. Latar Belakang

Dewasa ini internet sudah bukan merupakan hal yang asing dan baru lagi. Kata – kata *internet* sudah sangat akrab di telinga siapa saja. Mulai dari kalangan anak–anak yang bermain game *on-line* sampai kalangan orang dewasa yang berbisnis *online*, sudah sangat mengenal dengan apa yang di sebut dengan *internet*. *Internet* memungkinkan siapa saja untuk saling bertukar informasi tanpa harus dibatasi oleh ruang dan waktu.

Dengan semakin berkembangnya teknologi *internet* tersebut, maka semakin hari semakin banyak pula fasilitas-fasilitas *internet* untuk umum yang biasanya disebut dengan warung *internet* atau yang biasa dikenal dengan nama Warnet, yang hadir untuk menyediakan layanan *internet* bagi masyarakat, dimana masyarakat bisa menggunakan fasilitas yang disediakan oleh warung *internet* tersebut dalam mengakses layanan-layanan *internet*, misalnya untuk *browsing*, *email* dan lain sebagainya.

Dengan cukup besarnya minat masyarakat dalam mempergunakan jasa warung

internet tersebut, maka untuk mengelola *user* atau masyarakat dalam melakukan pemakaian terhadap perangkat komputer yang disediakan oleh warung *internet*, diperlukan sebuah aplikasi yang nantinya dapat mencatat hal-hal yang bersangkutan dengan pemakaian perangkat komputer yang disediakan oleh warung *internet* tersebut.

Dari hal tersebut di atas, maka penulis mencoba untuk mengambil judul Aplikasi Billing Warnet Berbasis Client Server Menggunakan Visual Basic 2005, dimana nantinya aplikasi tersebut dapat dipergunakan untuk menghitung jumlah biaya pemakaian *internet* oleh *user*.

2. Rumusan Masalah

Masalah yang diangkat dalam penulisan ini adalah bagaimana perancangan dan pembuatan aplikasi billing warnet client server menggunakan Visual basic 2005 dengan database Mysql 5.0

3. Tujuan Penulis

Tujuan dari pembahasan laporan tugas akhir ini adalah sebagai berikut :

1. Melakukan perancangan basis data dan perancangan aplikasi dari aplikasi yang akan dibuat.
2. Membuat aplikasi warung internet menggunakan Visual Basic 2005 dengan database Mysql 5.0
3. Melakukan pengujian (*test*) terhadap aplikasi yang telah dibuat untuk memastikan bahwa aplikasi tersebut dapat digunakan oleh pengguna / *user*.

4. Ruang Lingkup dan Batasan Masalah

Ruang lingkup masalah yang akan dibahas dalam pembuatan Billing Warnet Berbasis Client Server Menggunakan Visual Basic 2005 adalah dengan menggunakan *library socket namespace* dengan memanfaatkan *system.net* dan *system.net.socket* untuk komunikasi antara aplikasi *server* dan aplikasi *client* dalam sebuah jaringan Local Area Network (LAN). Serta beberapa perancangan aplikasi tersebut.

5. Metode Penulisan.

Metode yang penulis gunakan untuk mengumpulkan data-data dalam penulisan ini, yaitu:

1. Studi Pustaka.

Dengan mengumpulkan informasi untuk mendapatkan teori yang ada hubungannya dalam penulisan ini yaitu dengan cara membaca buku secara *literature* yang berkaitan dengan penulisan ini.

2. Studi Lapangan.

Penulis melakukan penelitian dengan cara mencari informasi dan mengumpulkan data-data yang ada sehingga dapat digunakan sebagai acuan pada penulisan ini.

3. Metodologi perancangan aplikasi dengan menggunakan Waterfall

Metode waterfall ini menggambarkan teknik yang praktis, model yang dilakukan tanpa melakukan operasi dan pemeliharaan.

6. Sistematika Penulisan

Untuk mendapatkan gambaran yang jelas dan mempermudah penulisan ini, maka penulis menyusunnya menjadi beberapa bab yaitu :

1. BAB I : PENDAHULUAN.

Pada bab ini penulis membahas dan mengemukakan tentang latar belakang masalah, batasan masalah, tujuan penulisan, dan metode penulisan yang dilengkapi dengan sistematika penulisan.

2. BAB II : LANDASAN TEORI

Bab ini berisikan tentang uraian mengenai berbagai macam teori dan aplikasi dari *software* pendukung dalam pembuatan aplikasi, seperti teori mengenai Visual Basic 2005, MySQL dan metode perancangan.

3. BAB III : ANALISA DAN PERANCANGAN

Bab ini menguraikan tentang analisis yaitu meliputi uraian tentang analisis sistem yang dilakukan, perancangan sistem yang meliputi penjelasan dari tahap-tahap merancang sistem dalam pembuatan Billing Warnet Berbasis Client Server Menggunakan Visual Basic 2005.

4. BAB IV : IMPLEMENTASI DAN PENGUJIAN

Di bab ini akan Menguraikan tentang Implementasi yaitu meliputi kebutuhan antarmuka eksternal, antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak dan batasan aplikasi. Dan juga akan dibahas pengujian sistem, hasil pengujian serta analisa hasil pengujian.

5. BAB V : KESIMPULAN DAN SARAN

Bab ini Menguraikan tentang kesimpulan dari keseluruhan pembuatan dan penulisan laporan Tugas Akhir, saran untuk pengembangan dan perbaikan serta untuk implementasi aplikasi agar berjalan sesuai pada saat pembuatannya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

LANDASAN TEORI

2.1. Pengertian Basis Data

Basis data merupakan kumpulan data yang saling berhubungan (relasi). Relasi biasanya ditunjukkan dengan kunci dari tiap *file* yang ada. Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field* yang saling berhubungan menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. (Harianto, 2001). Dengan basis data kita dapat dengan mudah mengolah data yang kita miliki. Disamping itu, pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) diantaranya :

1. Mencegah terjadinya redundansi dan inkonsistensi data
2. Menjaga integritas (*integrity*) dari data
3. Menjaga keamanan (*security*) dari data
4. Menjaga kebebasan data (*independent of data*)
5. Untuk efisiensi ruang penyimpanan (*space*)
6. Mengontrol pemakaian data secara bersama-sama

2.1.1 Kamus Data

Kamus data adalah daftar yang mencatat tentang banyaknya proses yang terjadi dalam sebuah sistem. Secara umum kamus data diklasifikasikan menjadi dua yaitu:

- a. Kamus data elementer yaitu daftar tentang semua elemen data yang berhubungan dengan sistem sehingga data yang mengalir dapat didefinisikan dan dapat tersimpan secara lengkap.
- b. Kamus Data Komposit, yaitu daftar tentang semua elemen data yang berhubungan dengan sistem dimana elemen data komponen ini terdiri dari dua elemen data elementer yang saling berkaitan

2.1.2 Entity Relationship (ER)

Model *ER* adalah suatu penyajian data menggunakan *entity* dan *relationship*. Pada model *ER*, semesta data yang ada di “dunia nyata” diterjemahkan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data yang umum disebut *Entity Relationship Diagram*.

2.1.2.1. Komponen Pembentuk Suatu *Entity Relationship*

Sesuai dengan namanya ada dua komponen utama pembentuk model *entity relationship*, yaitu entitas (*entity*) dan relasi (*relation*). Kedua komponen ini dideskripsikan lebih jauh melalui sejumlah atribut/ properti. *Entity* adalah objek yang dapat dibebankan dalam dunia nyata. *Relationship* adalah hubungan yang terjadi antara satu atau lebih *entity*.

Sedangkan atribut adalah karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut. Jenis-jenis atribut :

- a. *Key* atribut yang digunakan untuk menentukan suatu *entity* secara unik.
- b. Atribut *simple* yaitu atribut yang bernilai tunggal.
- c. Atribut *multivalued* yaitu atribut yang memiliki sekelompok nilai untuk setiap *instant entity*.
- d. Atribut komposit yaitu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.
- e. Atribut *derivative* yaitu suatu atribut yang dihasilkan dari atribut lain.

2.1.2.2. Derajat Kardinalitas dari Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan yang lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas dapat berupa:

1. Satu ke Satu (*one to one*)

Satu entitas dalam suatu himpunan dapat berhubungan hanya dengan satu entitas pada himpunan yang lain.

2. Satu ke banyak (*one to many*)

Satu entitas dalam suatu himpunan dapat berhubungan dengan lebih dari satu entitas pada himpunan yang lain.

3. Banyak ke Banyak (*many to many*)

Yang berarti setiap entitas dalam suatu himpunan dapat berhubungan dengan banyak entitas pada himpunan yang lain.

2.1.3. Entity Relationship Diagram

Entity Relationship Diagram merupakan gambaran sistematis model *Entity Relationship* yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta.

2.1.3.1. Tahapan pembuatan *Entity Relationship Diagram*

Entity Relationship Diagram selalu dibuat secara bertahap. Paling tidak ada dua kelompok tahapan yang biasa ditempuh dalam pembuatan diagram *Entity Relationship Diagram* yaitu:

- a. Tahap pembuatan diagram *entity relationship* awal (*preliminary design*)
- b. Tahap optimasi diagram *entity relationship* (*final design*)

Langkah-langkah teknis yang dapat kita lakukan untuk menghasilkan diagram *entity relationship* awal adalah:

- a. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat
- b. Menentukan atribut-atribut *key* dari masing-masing himpunan entitas

Ada empat macam kunci (*key*), antara lain:

1. *Candidate Key*

Candidate key adalah atribut atau set atribut yang mengidentifikasi secara unik kejadian spesifik suatu entity.

2. *Primary Key*

Primary Key adalah suatu atribut atau set atribut yang tidak mengidentifikasi secara unik suatu kejadian spesifik, tetapi dapat juga mewakili setiap kejadian dari suatu *entity*.

3. Alternate Key

Kunci kandidat yang tidak dipakai sebagai kunci primer.

4. Foreign Key

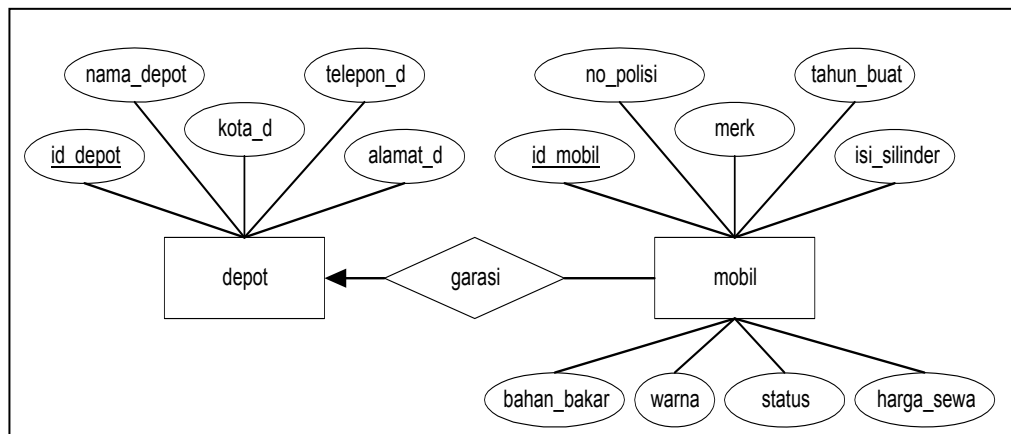
Satu atau set atribut yang dilengkapi satu relasi yang menuju ke induknya.

5. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta *foreign-key* nya

6. Menentukan derajat kardinalitas relasi untuk setiap himpunan relasi.

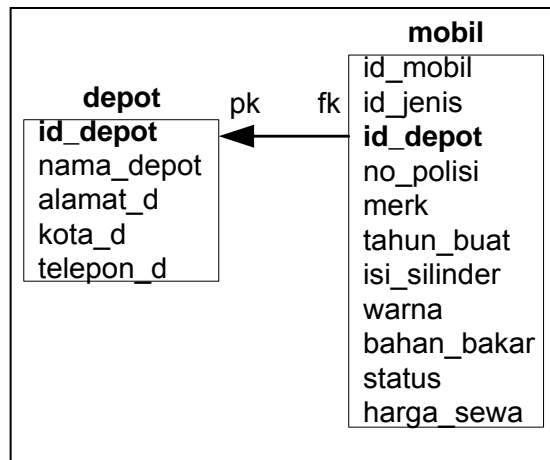
7. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif.

Adapun contoh dari *ERD* (*Entitas Relation Diagram*) yang merelasikan antara dua entitas, depot dengan mobil.



Gambar 2.1 Diagram E-R

Di dapat dari *ERD* diatas dapat dibuat *Schema Diagram* yang meliputi beberapa faktor



Gambar 2.2 *Schema Diagram*

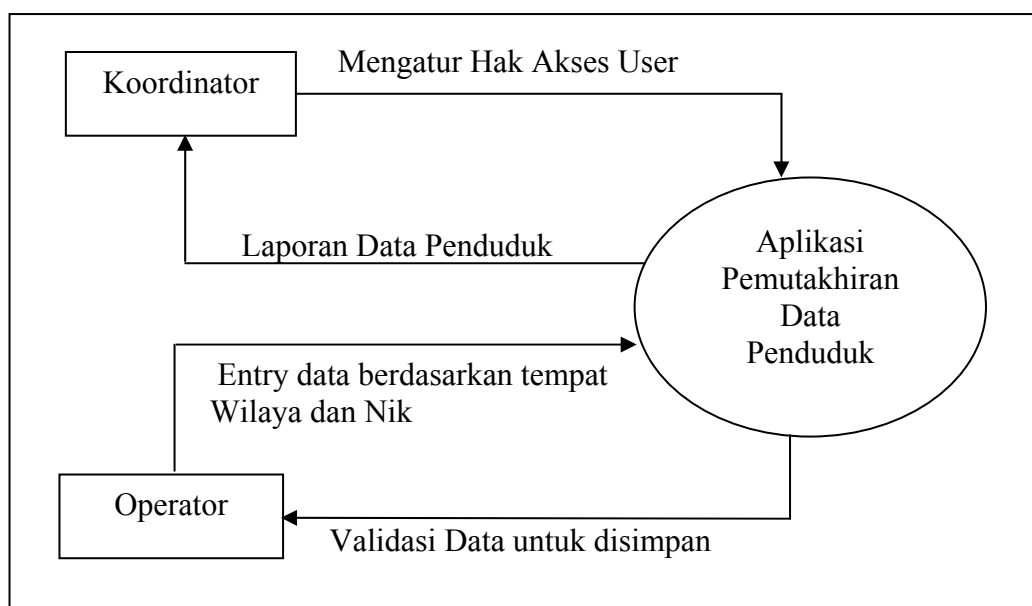
2.1.4 Data Flow Diagram

Diagram aliran data (DAD-DFD/*data flow diagram*) memperlihatkan hubungan fungsional dari nilai yang dihitung oleh sistem, termasuk nilai masukan, nilai keluaran, serta tempat penyimpanan internal.

Levelisasi pada *DFD* terbagi menjadi tiga bagian yaitu :

- a. Diagram Konteks yaitu penggambaran sistem secara keseluruhan.
- b. Diagram Level Satu yaitu penggambaran sistem dari diagram konteks.
- c. Diagram Detailed yaitu penggambaran dari masing-masing proses atau primitive (tidak dapat dipecahkan lagi menjadi yang lebih kecil).

Berikut dari diagram konteks dapat ditunjukkan pada gambar 2.3

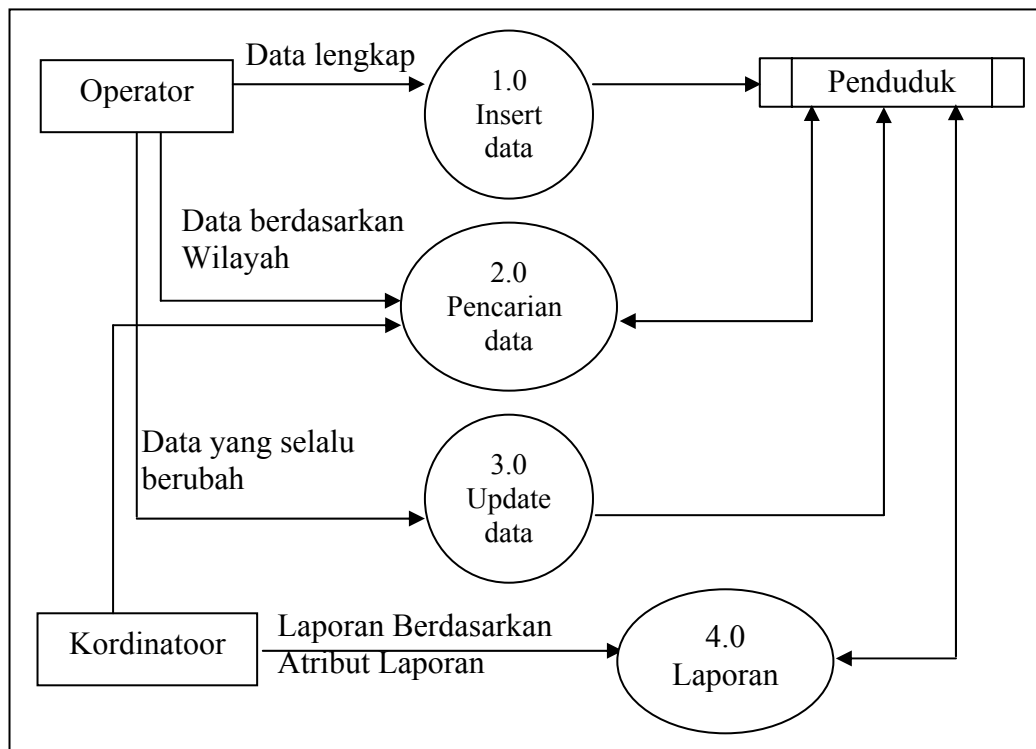


Gambar 2.3 Diagram Konteks Aplikasi Pemuktahiran Data Penduduk

Diagram konteks yaitu penggambaran sistem secara keseluruhan pada satu proses, pada contoh di atas menunjukkan hanya terdapat satu proses yaitu proses pada Aplikasi pemutahiran data penduduk, dan terdapat dua entitas yang mengendalikan objek aktif dengan menerima sistem input dan output.

Pada diagram konteks hanya terdapat 3 komponen karena proses yang begitu sederhana, sedangkan pada diagram *DFD level 1* penggambaran masing-masing proses dapat diperkecil lagi sehingga terdapat 4 komponen karena prosesnya yang begitu detail.

Setelah didapatkan diagram konteks maka untuk memperjelaskan yaitu pembuatan diagram selanjutnya yang menggambarkan dari masing-masing proses dapat dapat dipecahkan lagi menjadi yang lebih kecil. Berikut dari *DFD level 0* dapat ditunjukkan pada gambar 2.4.



Gambar 2.4 DFD Level 1 Aplikasi Pemutahiran Data Penduduk

Didalam *Data Flow Diagram (DFD)* terdapat empat komponen, yang masing-masing komponen memiliki fungsi. Keempat komponen tersebut yaitu :

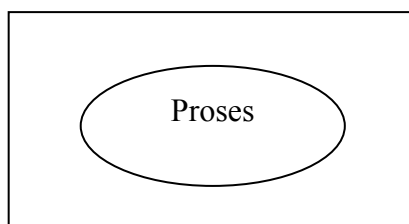
1. Proses (*Prosess*)
2. Aliran Data (*Data Flow*)
3. Penyimpanan Data (*Data Store*)
4. Entitas

Contoh dari masing – masing ke empat komponen *Data Flow Diagram* dapat ditunjukana pada halaman berikutnya, yaitu :

2.1.4.1 Proses (*Prosess*)

Proses adalah sesuatu yang melakukan transformasi terhadap data, yaitu mentransformasikan data dari masukan menjadi keluaran. Dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran atau sebaliknya.

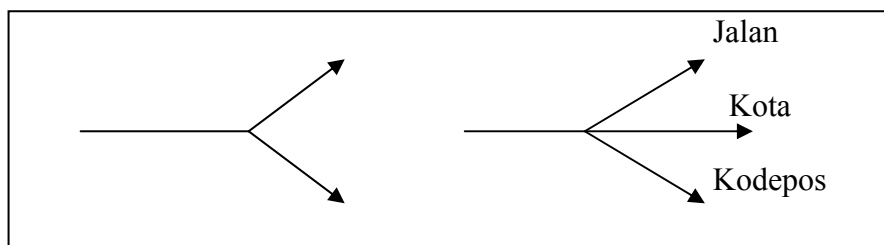
Proses direpresentasikan dalam bentuk lingkaran. Simbol proses ditunjukkan oleh gambar 2.5.



Gambar 2.5. Simbol Proses

2.1.4.2 Aliran Data (*Data Flow*)

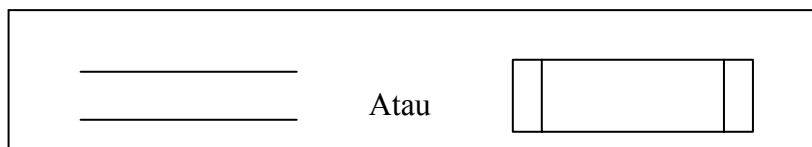
Aliran data menghubungkan keluaran dari suatu objek atau proses yang terjadi pada suatu masukan. Digunakan untuk menggambarkan gerakan paket data atau informasi dari satu bagian ke bagian lain dari sistem penyimpanan mewakili lokasi penyimpanan data. *Simbol data flow* direpresentasikan oleh panah yang menuju ke atau dari proses seperti yang diperlihatkan oleh Gambar 2.6.



Gambar 2.6. Simbol Aliran Data

2.1.4.3 Penyimpanan Data (*Data Store*)

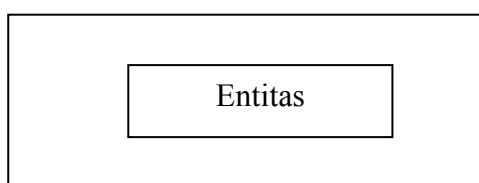
Data store adalah objek pasif dalam *DFD* yang menyimpan data untuk penggunaan lebih lanjut atau paket data. *Data store* ini dinotasikan dengan garis sejajar, atau persegi panjang. *Simbol* untuk penyimpanan *data store* ini diperlihatkan oleh Gambar 2.7



Gambar 2.7 *Simbol Data Store*

2.1.4.4 Entitas

Entitas adalah objek aktif yang mengendalikan aliran data dengan memproduksi atau mengkonsumsi data. Komponen ini menyediakan sistem *input* data dan menerima *output* data pada sistem. Biasanya notasi ini melambangkan orang atau kelompok orang misalnya organisasi, departemen atau perusahaan. Dalam *DFD* komponen ini direpresentasikan menggunakan persegi panjang. Gambar 2.8 memperlihatkan simbol entitas.



Gambar 2.8. Simbol Entitas

2.2 Model *Waterfall*

Dalam perancangan suatu perangkat lunak dapat digunakan model *waterfall*, yaitu model yang dilakukan tanpa melakukan operasi dan pemeliharaan. Dalam model *waterfall* ini terdapat lima tahap, tahap-tahap tersebut antara lain :

1. Keperluan Analisis dan Definisi

Pelayanan sistem, Pembatasan dan hasil tidak dihindari dari konsultasi dengan sistem pengguna, mendefinisikan cara yang sama yang dapat dimengerti oleh kedua belah pihak pengguna.

2. Sistem dan Desain Perangkat Lunak

Sistem proses desain memisahkan keperluan untuk salah satu perangkat keras atau sistem perangkat lunak. Tidak dapat dihindari dalam arsitektur sistem desain perangkat lunak meliputi perwakilan fungsi sistem perangkat lunak didalam program yang mungkin juga berubah ke bentuk salah satu banyak program yang bisa dijalankan.

3. Implementasi dan Pengujian

Selama berjalan, desain perangkat lunak menyadari menguji apakah ada kesalahan beberapa program atau salah satu program. Pengujian unit meliputi pemeriksaan setiap unit yang lebih spesifikasi.

4. Integrasi dan Sistem Pengujian

Unit individual program atau program-program yang digabungkan (diintegrasikan) dan pengujian dengan keseluruhan sistem untuk memastikan bahwa keperluan perangkat lunak telah dilakukan, setelah pengujian, sistem perangkat lunak dikirim ke pengguna.

5. Operasi dan Pemeliharaan

Biasanya operasi ini adalah fase putaran pengujian yang cukup lama, sistem di instal dan disimpan dalam tempat yang praktis digunakan. Pemeliharaan meliputi pengoreksian error yang mana tidak dapat ditemukan di awal-awal fase putaran

pengujian yang lainnya. Memperbaiki implementasian dari unit sistem dan mengutamakan layanan sistem seperti keperluan baru yang telah ditemukan.

2.3 Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan (Haris N.,2004). Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama-sama menggunakan *hardware* atau *software* yang terhubung dengan jaringan. Setiap komputer, printer atau periferal yang terhubung dengan jaringan disebut node. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node. Jenis-jenis Jaringan komputer secara umum jaringan komputer dibagi atas lima jenis, yaitu :

1. *Local Area Network (LAN)*, merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. *LAN* seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan workstation dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai bersama sumberdaya (*resource*, misalnya printer) dan saling bertukar informasi.
2. *Metropolitan Area Network (MAN)*, pada dasarnya merupakan versi *LAN* yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan *LAN*. *MAN* dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum.

MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel (Haris N.,2004).

3. *Wide Area Network (WAN)*, jangkauannya mencakup daerah geografis yang luas, seringkali mencakup sebuah negara bahkan benua. WAN terdiri dari kumpulan mesin - mesin yang bertujuan untuk menjalankan program-program (aplikasi) pemakai.

4. Internet. Seringkali menggunakan perangkat keras dan perangkat lunak yang berbeda-beda. Orang yang terhubung ke jaringan sering berharap untuk bisa berkomunikasi dengan orang lain yang terhubung ke jaringan lainnya. Keinginan seperti ini memerlukan hubungan antara jaringan yang seringkali tidak sama dan berbeda. Biasanya untuk melakukan hal ini diperlukan sebuah mesin yang disebut *gateway* guna melakukan hubungan dan melaksanakan terjemahan yang diperlukan, baik perangkat keras maupun perangkat lunaknya. Kumpulan jaringan yang terinterkoneksi inilah yang disebut dengan internet.

5. Jaringan tanpa kabel, merupakan suatu solusi terhadap komunikasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel. Misalnya orang yang ingin mendapat informasi atau melakukan komunikasi walaupun sedang berada diatas mobil atau pesawat terbang, maka mutlak jaringan tanpa kabel diperlukan karena koneksi kabel tidaklah mungkin dibuat di dalam mobil atau pesawat. Saat ini jaringan tanpa kabel sudah umum digunakan dengan memanfaatkan jasa satelit dan mampu memberikan kecepatan akses yang lebih cepat dibandingkan dengan jaringan yang menggunakan kabel.

2.3.1 TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) merupakan kombinasi dari dua protokol terpisah. *IP* adalah *layer 3 protocol* - suatu layanan *connectionless* yang menyediakan layanan pengantar data terbaik dalam jaringan. *TCP* adalah *layer 4 protocol*, suatu layanan *connection-oriented* yang menyediakan pengontrolan aliran data yang sering disebut sebagai *reliability*. Penggabungan kedua protokol ini memungkinkan penyediaan layanan yang semakin luas.

2.3.2 TCP Protocol

Transmission Control Protocol (TCP) adalah sebuah *layer 4 protocol* yang bersifat *connection-oriented* yang menyediakan transmisi data *full-duplex* yang dapat diandalkan. *TCP* adalah bagian dari *TCP/IP protocol stack*.

2.3.3 Internet Protocol (IP)

Internet Protocol (IP) adalah *protocol* jaringan (*Network Layer* pada *OSI Layer Model* [2]), data dikapsulasi pada setiap *layer*. Pada *layer network*, data dikapsulasi dalam paket – paket (atau disebut juga *datagram*), *IP* menentukan bentuk dari *packet header* (yang mana termasuk pengalamatan atau *addressing* dan informasi kontrol lainnya) tetapi tidak peduli mengenai data yang sebenarnya, dia menerima apapun yang diberikan oleh *layer* di atasnya.

2.3.4 Port Number

Istilah port bagi pengguna internet tentu bukan hal yang asing lagi karena begitu sering dipakai pada saat mengakses layanan yang ada di Internet. Misalnya port 194 dipakai untuk protokol IRC, port 1080 untuk SOCKS (proxy IRC), port 21 untuk FTP, port 80 atau 8080 untuk HTTP (WWW), port 25 untuk SMTP (E-mail), port 22 untuk SSH (Secure Shell), dan port 23 untuk Telnet.

Internet Assigned Numbers Authority (IANA) yang merupakan sub dari Internet Architecture Board (IAB) - sebuah badan koordinasi dan penasihat teknis bagi masyarakat pengguna Internet, membagi port number ke dalam tiga jarak (range): (Sigit,2003)

1. Jarak pertama, port 0-1023, disebut *Well Known Port Numbers*.
2. Jarak kedua, port 1024-49151, disebut *Registered Port Numbers*.
3. Jarak ketiga, port 49152-65535, disebut *Dynamic Ports* atau *Private Ports*.

2.4 Socket Programming

Socket programming adalah metode atau cara menggunakan socket API (Application Socket Programming), untuk menghubungkan komunikasi antara proses remote dan local, programmer yang menggunakan ILE (Integrated Language Environment) C dapat menggunakan informasi untuk mengembangkan aplikasi socket. Anda juga dapat menggunakan kode socket API dari bahasa ILE lainnya, seperti RPG. Java juga mendukung alat penghubung untuk socket programmin dan maksud dari API adalah sebuah library atau fungsi windows untuk mengeksekusi sebuah perintah yang ada di dalam semua tampilan windows, sebagai contoh untuk tampilan desktop, screen saver, shutdown komputer dan lain – lain. (Andrew Fiade,2006)

Winsock dapat didefinisikan dan di dokumentasikan oleh standar *API* untuk pemrograman *protocols* jaringan. Umumnya digunakan untuk program *TCP / IP*. *Winsock* dapat digunakan untuk *DLL (Dynamic Link Library)* yang merupakan bagian dari *Win32*. Cara kerja dari *winsock* adalah perpaduan antara *IP Address* dan *port number*, kemudian apa yang dimaksud dengan *IP Address*. *IP Address* adalah sebuah alamat virtual pada setiap komputer untuk mengirimkan data, contoh dalam kehidupan nyata adalah setiap kita mengirim surat kita harus menyertakan alamat pengirim dan alamat tujuan, jika tidak ada maka paket tidak bisa dikirim karena tidak jelas alamat pengirim dan alamat penerima. contoh alamat *IP Address* adalah 192.168.1.1 Lalu yang dimaksud dengan *port number*, pada pelayanan jaringan di komputer, *port number* dapat dikatakan sebagai identifikasi khusus untuk setiap proses, contoh jika anda menggunakan layanan internet seperti google.com anda telah menggunakan *port number* tertentu, yaitu port 80. Jadi windows socket adalah perpaduan *IP Address* dan port. (Andrew Fiade,2006)

2.5 Pengenalan MySql

MySQL merupakan salah satu database relasional yang mendukung pemakaian *Structured Query Language (SQL)* dan dirancang untuk penggunaan aplikasi dengan arsitektur *client server*. MySQL memungkinkan pengguna untuk mengolah data didalam *database* yang tersentral pada komputer pusat yang disebut sebagai *server*. Sedangkan informasi yang dihasilkan dapat digunakan bersama-sama oleh beberapa *user*. Semua komunikasi yang terjadi mendukung perintah-perintah *SQL (Query)*.

Secara umum, kelebihan MySQL adalah tersedia gratis untuk banyak sistem operasi dan hardware, memiliki kecepatan proses dan kemampuan menangani *database* dengan sangat baik di banyak tipe hardware dan sistem operasi yang digunakan. Selain itu MySQL juga mudah digunakan karena bahasa yang digunakannya termasuk mudah, mendukung SQL (*Structured Query Language*), mampu bekerja dengan beban berat dengan banyak *client* yang terhubung dalam satu waktu secara bersamaan, dapat diakses dari mana saja dari *internet*, mudah diperoleh dan yang terakhir faktor keamanan yang terjamin.

2.5.1 Memulai Dengan MySQL

Program client yang biasanya digunakan untuk mengakses server MySQL di Linux adalah program `mysql`. Program interaktif ini memungkinkan Anda mengakses server MySQL, menjalankan query dan menyaksikan hasilnya.

Untuk konek ke server, biasanya dibutuhkan *username* MySQL ketika Anda menjalankan `mysql`, dan biasanya juga *password*. Jika servernya berbeda dari komputer tempat Anda *login*, Anda juga harus menuliskan *hostname*-nya. Contohnya seperti berikut :

```
$ mysql -h host -u user -p
Enter password: *****
```

Tanda `*****` menunjukkan *password* Anda, masukkanlah ketika `mysql` menampilkan `Enter password:`. Jika Anda berhasil login, Anda akan melihat informasi *introduction* yang diikuti oleh prompt `mysql>` :

```

shell> mysql -h host -u user -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 459 to server version: 3.22.20a-log
Type 'help' for help.

mysql>

```

2.5.2 Perintah *Query* Pada MySql

Query sebenarnya berarti permintaan. Dengan menggunakan *query*, kita bisa melihat, mengubah, dan menganalisis data dengan berbagai titik pandang yang kita kehendaki. Selain itu *query* juga dapat dipakai sebagai data bagi formulir, laporan dan bahkan halaman web.

Berikut ini diberikan contoh tentang bagaimana memecahkan masalah yang umum ditemukan pada MySQL :

2.5.2.1 Statement *Create Table*

Digunakan untuk Membentuk tabel dari basis data dengan menyebutkan spesifikasi dan batasan atribut pada tabel.

```

mysql> CREATE TABLE <nama Tabel>
(
  <Nama Field><Type Field>,
  <Nama Field><Type Field>,
  . . . .
  PRIMARY KEY (<Nama Field>)
);

```

2.5.2.2 Statement *Select . . . From*

Statement ini di gunakan untuk memperoleh atau menyajikan data dari sebuah atau beberapa tabel. Format penulisan *Select* yang lengkap, yaitu :

```

Mysql> SELECT (ALL | DISTINCT) <nama_kolom_table>
FROM <namatabel>;

```

contoh Sintaks menampilkan semua (*) pada tabel customer :

```

mysql> select * from customer;
+-----+-----+-----+-----+
| id_customer | nama_customer | alamat_c          | kota_c    |
+-----+-----+-----+-----+
|          1 | Amir         | Jl Tanjung no 42  | Yogyakarta |
|          2 | Adi Yanto     | Jl Pelajar no 76  | Solo       |
|          3 | Bondan        | Jl Pandean no 11  | Jakarta    |
+-----+-----+-----+-----+
3 rows in set (0.05 sec)

```

2.5.2.3 Statement Insert Into

Statement ini berguna untuk menambahkan data ke dalam basis data.

```

INSERT INTO <nama_tabel> ( <kolom1>,<kolom2>)
VALUE (nilai1,nilai2);

```

Contoh sintaks :

```

Mysql>INSERT INTO
customer(id_customer,nama_customer,alamat_c,kota_c)
VALUES (1,'Amir','Jl Tanjung no 42','Yogyakarta',);

```

2.5.2.4 Statement Update

Statement ini digunakan untuk mengubah isi data dalam basis data.

```

UPDATE <nama_tabel>
SET <kolom1>=<nilai_baru1>,<kolom2>=<nilai_baru2>,...
WHERE <kondisi>

```

Contoh sintaks :

```

mysql> UPDATE customer
SET nama = "Jasmine"
WHERE id_customer = 1;

```

Dan jika di tampilkan hasilnya akan :

id_customer	nama_customer	alamat_c	kota_c
1	Jasmine	Jl Tanjung no 42	Yogyakarta
2	Adi Yanto	Jl Pelajar no 76	Solo
3	Bondan	Jl Pandean no 11	Jakarta

2.6 Pengenalan Visual Studio 2005

Visual Studio 2005 merupakan *core* dari pembuatnya aplikasi berbasis.net. merupakan lingkungan pemrograman yang mempermudah tahap desain, *development*, *debuging*, dan *deployment* dari aplikasi berbasis .Net dan XML dan *Web Service*, serta meningkatkan efisiensi developer dengan menyediakan lingkungan pemrograman yang sudah biasa di gunakan dan bisa di *share*. (Suryanto Suharli ,2005:1-3)

Sebagai *development tool*, Visual Studio 2005 menyediakan dukungan untuk berbagai bahasa pemrograman, *tools* untuk membuat web, windows dan XML *Web service*, data akses, dan *error hendling*, termasuk *debugging* lokal, *remote* dan *tracing*, bahkan *cross language debugging*.

2.6.1 .Net Framework

Microsoft mengembangkan *framework* baru, *.Net framework* merupakan infrastruktur untuk pengembangan dan eksekusi aplikasi. Framework mengatur semua aspek eksekusi program termasuk pengalokasian memori untuk penyimpanan data dan instruksi, hak akses dari aplikasi, eksekusi aplikasi, juga mengatur realokasi memori *resource* yang sudah tak dipakai lagi.

Pada *.Net framework*, komponen dibuat berdasarkan landasan yang sama. Kita tak perlu lagi menulis program untuk mengizinkan suatu objek untuk berinteraksi satu sama lain secara langsung.

.Net Framework didesain agar menyesuaikan lintas bahasa, komponen-komponen *.net* bisa saling berinteraksi tanpa tergantung pada bahasa pemrograman yang dipakai untuk pembuatannya. Sebagai contoh, aplikasi yang dibuat dengan Visual Basic.net bisa mengakses DLL yang dibuat dengan Microsoft C#, C++, atau bahasa *.Net* lain.

2.6.2 Socket Programming pada Visual Basic 2005

Bagaimana cara aplikasi *client* dan *server* dapat berkomunikasi dengan menggunakan socket pada visual basic.net, sebuah class dalam *namespace System.net.socket* menyediakan sebuah implementasi yang sangat populer pada windows sockets(winsock) untuk penggunaan dengan bahasa *.Net*.

Sockets sebagai bagian dari *unix* operasi sistem, dan secara luas sudah digunakan di internet serta dapat digunakan di antara program di atas satu mesin, *sockets* dapat di ilustrasikan seperti telepon untuk berkomunikasi di antara karyawan di perusahaan berbeda. jika saya ingin menghubungi anda dari telepon saya, yang saya perlukan adalah nomor telepon dan nomor extension anda. dalam *socket communication*, nomor telepon adalah *IP address* yang ingin dituju dan *extension* adalah *port number* karena pasti saja setiap orang di kantor mempunyai beberapa *extension* nomor telepon yang sama.

Sejumlah port digunakan dari 0 – 1024, disamping untuk web server dan mail server, anda secara bebas menggunakan port di atas 1024 untuk penggunaan baru (pribadi). Proses server dapat memesan sejumlah *port* dan menunggu panggilan

masuk. Proses client membuat panggilan untuk membuka sebuah *socket* menetapkan *IP address* dan *port number* yang dituju, jika *ip address* dan *port number* benar, dua proses akan dihubungkan.

2.6.2.1 Bagaimana *Socket* dapat digunakan

Setelah dibahas sebelumnya pada bab 2.6.2, *namespace system.net.socket* berisi *class socket* yang dapat melakukan apa yang kita inginkan. *microsoft* menganjurkan untuk menggunakan dua class, *TCPClient* mengarah ke *client* dan *TCPListener* mengarah ke *server*.

Pada *end user (client)*, kita membuat *TCPclient*, melalui *IP address* pada mesin yang akan dikoneksikan dan port yang digunakan oleh proses server.

Ketika aplikasi terkoneksi, maka tidak dapat dihindari metode *getstream()* mengirimkan kembali ke objek *stream*. yang telah digunakan untuk membaca dan menulis data berupa *bytes* melalui *socket*.

```
Dim readStream as stream = serverListener.GetStream
```

Metode *Stream.Read()* dan *Write()* dapat digunakan untuk mengirim data melalui *socket*, tetapi karena menggunakan *bytes*, itu perlu dilakukan konversi data karakter ke bytes sebelum dikirim. Sebagai alternatif kita dapat menggunakan metode *send()* dan *recieve()* pada *socket* yang juga dapat berkerja pada pengaturan *byte*. (Fariz Butt,2003).

TCPListener dapat membantu dalam pengimplementasian *server* pada *socket connection*. *TCPListener* dapat membuat dan menerima keterangan pada *socket* yang mana sesungguhnya mempunyai kesamaan pada client lain yang akan di panggil dalam keadaan aktif.

```
Dim ClientListener As New TcpListener(8585)
```

Metode *Start()* digunakan untuk menjalankan metode objek *Listening connection* dalam jaringan, ada dua arah ketika *server* dapat terhubung ke dalam *clients*, satu arah digunakan untuk memanggil *AcceptSocket()* dan *AcceptTCPClient()*, dan keduanya dapat diblok sampai *client* telah terhubung atau server pada waktu tertentu dapat memanggil metode *TCPListener.pending()* yang mana mengembalikan nilai *true* jika beberapa *client* dalam keadaan menunggu untuk koneksi, pemanggilan *AcceptSocket()* atau *AcceptTCPClient()* akan terhubung dengan seketika. (Fariz Butt,2003)

Berikut ini adalah contoh membuat aplikasi server dan client dapat berkomunikasi menggunakan *socket* :

```

1. Imports System.IO
2. Imports System.Net.Sockets
3. Module Module1
4.     Sub Main()
5.         Try
6.             Console.WriteLine("Connecting to 8585 Local Host")
7.             Dim serverListener As New TcpClient("localhost", 8585)
8.             Dim readStream As Stream = serverListener.GetStream
9.             serverListener.SendBufferSize = 256
10.            Console.WriteLine("Input Lines:")
11.            Dim str As String = Console.ReadLine() ` membuat
12. variabel str dengan tipe data string
13.            While True
14.                Dim sendBuff As Byte() = _
15. System.Text.Encoding.ASCII.GetBytes(str) ` merubah karakter
16.menjadi bytes pada variable str dan mengirim data ke server
17.                readStream.Write(sendBuff, 0, sendBuff.Length)
18.                If str.StartsWith(".") Then
19.                    GoTo Done
20.                End If
21.                str = Console.ReadLine()
22.            End While
23. Done: Console.WriteLine("Done")
24.            Catch exp As Exception
25.                Console.WriteLine("Exception: " + exp.ToString())
26.            End Try
27.        End Sub
28.    End Module

```

Gambar 2.1 source code aplikasi console pada client

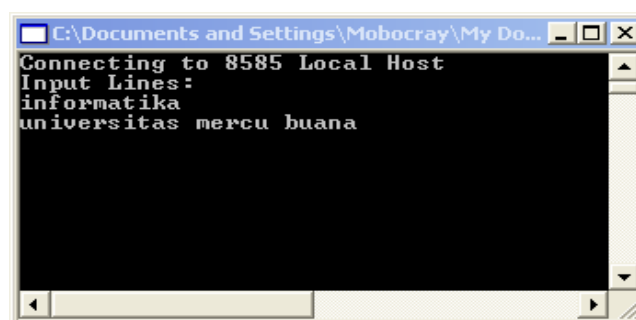

```

1. Imports System.IO
2. Imports System.Net.Sockets
3. Module Module1
4.     Sub Main()
5.         Console.WriteLine("")
6.         Dim clientListener As New TcpListener(8585) ' port 8585
7.         clientListener.Start() ' server dalam keadaan aktif atau
8. menunggu koneksi client
9.         Console.WriteLine("")
10.        Dim mySocket As Socket = clientListener.AcceptSocket()
11.        Console.WriteLine("")
12.        Dim receiveBuff(225) As Byte
13.        mySocket.Receive(receiveBuff, receiveBuff.Length,
14. SocketFlags.None)
15.        Dim str As String =
16. System.Text.Encoding.ASCII.GetString(receiveBuff, 0,
17. receiveBuff.Length).Trim(Microsoft.VisualBasic.ChrW(0))
18.        While Not str.StartsWith(".")
19.            Console.WriteLine(str)
20.            mySocket.Receive(receiveBuff, receiveBuff.Length,
21. SocketFlags.None)
22.            str=System.Text.Encoding.ASCII.GetString(receiveBuff,
23. 0, receiveBuff.Length).Trim(Microsoft.VisualBasic.ChrW(0))
24. ' merubah string(str)menjadi character
25.        End While
26.        Console.WriteLine("")
27.        clientListener.Stop()
28.    End Sub
29. End Module

```

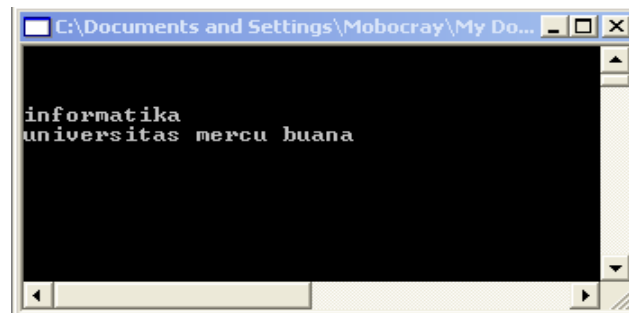
Gambar 2.2 *source code aplikasi console pada server*

Tampilan Aplikasi *client* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Tampilan aplikasi *console client*.

Tampilan Aplikasi *server* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Tampilan aplikasi *console server*.

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisis

Analisis merupakan suatu penguraian sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan.

Seperti langkah-langkah yang dilakukan pada salah satu model proses rekayasa perangkat lunak yaitu model waterfall, maka pada bab ini akan dibahas tentang tahap-tahap dalam membangun perangkat lunak. Proses analisis merupakan salah satu tahap yang harus dilalui rekayasa piranti lunak karena melalui analisis definisi masalah menjadi lebih jelas, kebutuhan sistem dapat dispesifikasi sehingga kriteria yang harus dipenuhi sistem dapat ditentukan supaya sistem yang dihasilkan nantinya dapat menjadi solusi dari masalah tersebut.

3.1.1 Analisis Masalah

Masalah yang didefinisikan dalam tugas akhir ini yaitu kesulitan yang dihadapi oleh warung internet pada pengolahan harga pemakaian non anggota maupaun anggota , keanggotaan pengguna dan laporan harian serta bulanan.

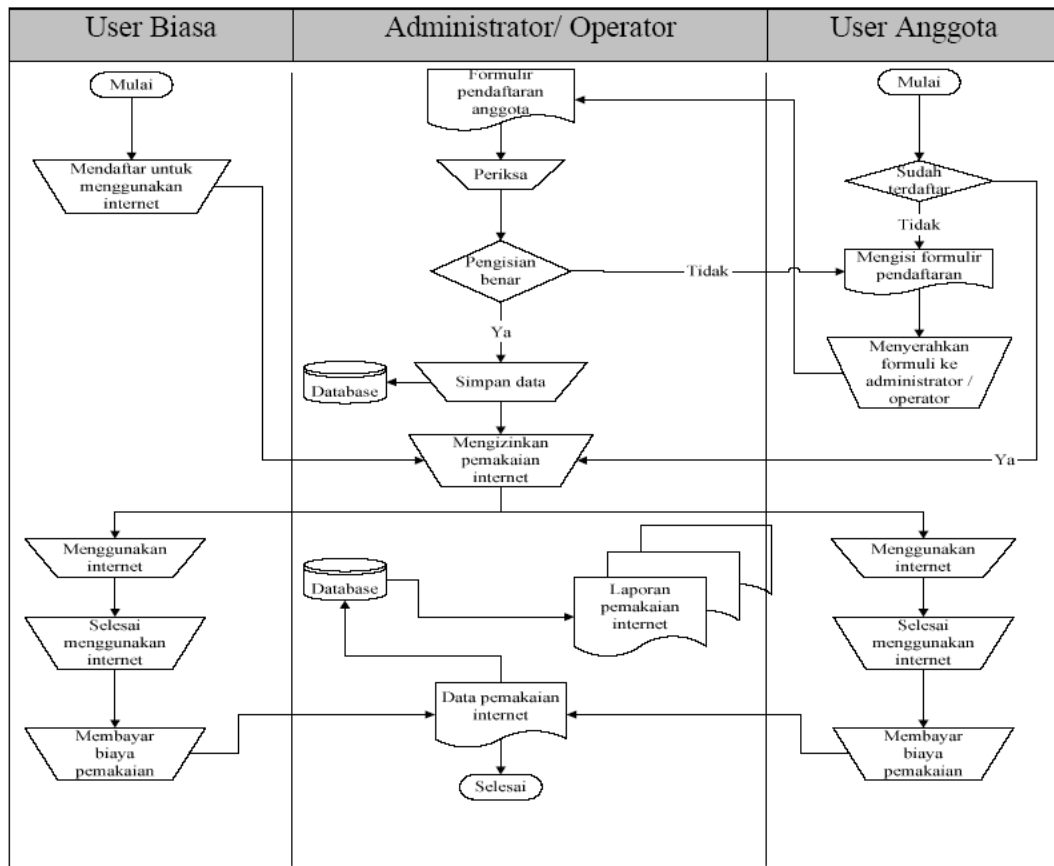
Melihat masalah di atas penulis memberikan alternatif pemecahan masalah yaitu dengan membangun aplikasi menggunakan visual basic.net dengan database Mysql, yang ketahu Mysql adalah server basis data yang kecil dan dapat menangani transaksi dengan kecepatan tinggi.

3.1.2 Analisis Sistem Berjalan

3.1.2.1 Deskripsi Prosedur Kerja

Pelaksanaan prosedur kerja oleh administrator atau operator pada warung internet adalah administrator atau operator bertugas untuk mengatur data – data yang berkaitan dengan pemakaian internet oleh user, baik itu berupa data user itu sendiri ataupun data biaya pemakaian internet yang dilakukan secara manual.

Proses pemakaian internet oleh user di tunjukan oleh gambar 3.1



Gambar 3.1 Flowchart Sistem Berjalan

3.1.2.2 Identifikasi Kebutuhan Pemakai

1. Sistem harus dapat membantu mempermudah pekerjaan administrator dan operator.
2. Sistem harus dapat memberikan informasi yang dibutuhkan oleh user.
3. Sistem harus dapat dioperasikan oleh administrator, operator dan user.
4. Data – data yang diperlukan harus di *entry* melalui aplikasi server dan hanya dapat di kelola oleh administrator atau operator, tetapi tetap dapat menginformasikan beberapa data untuk user pada aplikasi client.
5. Semua data disimpan secara terpusat.
6. Ada laporan harian dari Administrator/ Operator.
7. Ada laporan bulanan dari Administrator/ Operator.

8. Semua proses yang berkaitan dengan pengolahan data harus terintegrasi dalam perangkat lunak aplikasi dan hanya dilakukan pada aplikasi server.

3.1.3 Analisis Kebutuhan Sistem

Kebutuhan sistem yang terdapat pada sistem berjalan untuk sebuah warung internet meliputi :

3.1.3.1 Kebutuhan Informasi

Tabel 3.1 Analisis Kebutuhan Sistem

No	Informasi yang dibutuhkan	Tujuan	Frekuensi
1	Laporan data Anggota	Administrasi / Operator	Saat diperlukan
2	Laporan Pemakaian internet	Administrasi/ Operator	Setiap hari/ setiap bulan
3	Laporan pemakaian internet	User	Setiap pemakaian internet

3.1.3.2 Kebutuhan Aplikasi

Kebutuhan informasi yang harus terpenuhi dalam proses pengolahan data Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 adalah :

1. Client

Menampilkan informasi harga atau biaya untuk pemakaian internet bagi user biasa dan user anggota.

2. Server

Mengolah data user, data anggota, data biaya serta membuat laporan – laporan yang diperlukan untuk pemakaian internet.

3.1.3.3 Kebutuhan Perangkat Keras

Jaringan komputer lokal (LAN) dengan protokol TCP/ IP :

1. Server

Untuk menjalankan aplikasi server dan melakukan pengolahan data.

2. Workstation

Untuk menjalankan aplikasi client, menampilkan informasi yang diperlukan oleh user yang berkaitan dengan penggunaan internet.

3.1.4 Analisis Kebutuhan Perangkat Lunak

3.1.4.1 Deskripsi Kebutuhan Fungsional

A. Server

Tabel 3.2 Analisis Kebutuhan Perangkat Lunak Server

No	Dekripsi Kebutuhan	Keterangan
1	Login	Melakukan login Administrator / Operator
2	Switch User	Melakukan Pergantian Operator
3	Exit	Mengakhiri Aplikasi
6	Shutdown / Restart Client	Melakukan shutdown atau restart pada aplikasi client
7	Setting biaya	Melakukan setting biaya pemakaian internet
8	Daftar baru	Input data anggota baru
	Table anggota	Menampilkan tabel daftar anggota
9	Cari / edit anggota	Mencari / mengedit anggota yang sudah ada
10	Tambah operator	Input data operator baru
11	Cari / edit operator	Dari / ubah data operator yang sudah ada
12	Tabel Operator	Menampilkan seluruh data operator
13	Laporan data pemakaian	Mencetak laporan biaya pemakaian internet
14	Laporan data operator	Mencetak data daftar Operator
15	Laporan data anggota	Mencetak data daftar Administrator
17	password	Melakukan pergantian password untuk administrator dan operator

B. Client

Berikut ini adalah tabel analisis kebutuhan perangkat lunak pada client ditunjukkan oleh tabel 3.3.

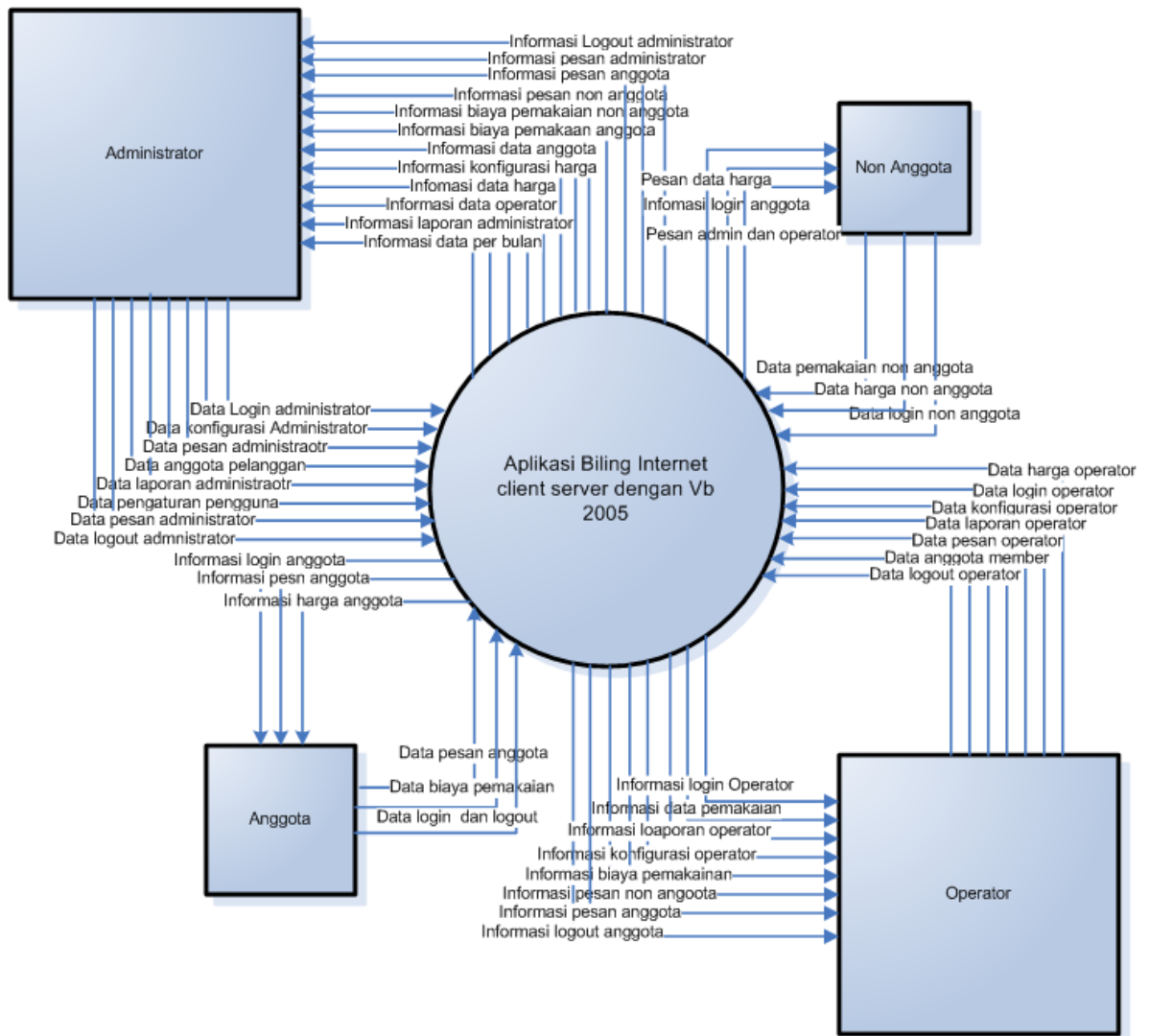
Tabel 3.3 Analisis Kebutuhan Perangkat Lunak Client

No	Deskripsi	Keterangan
1	Login pemakaian	User melakukan login dan memulai pemakaian internet
2	Logout	User melakukan logout dan mengakhiri pemakaian internet
3	konfigurasi	Administrator / Operator melakukan login untuk mengkonfigurasi client
4	Exit	Administrator / operator menutup aplikasi

3.1.4.2 Pemodelan desain

Pada pembuatan model desain suatu aplikasi Billing internet *client - server*, penulis menggunakan model DFD (*Data Flow Diagram*) yang berfungsi untuk mengetahui alur data dari suatu proses ke yang lain dalam beberapa tingkatan atau level. DAD ini memberikan kemudahan dalam menentukan data yang digunakan sebagai masukan maupun keluaran data.

A. Diagram Konteks (*Context Diagram*)

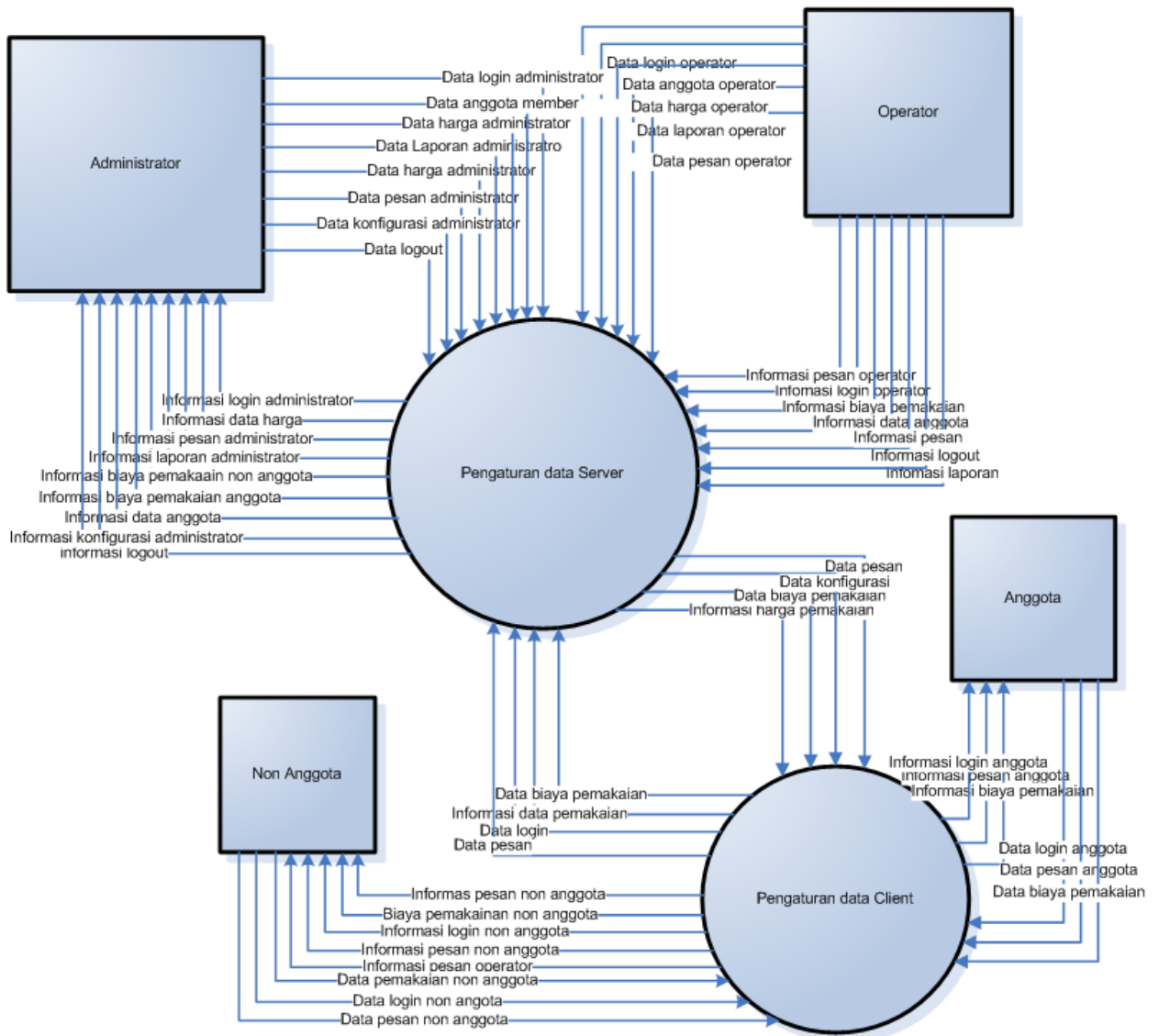


Gambar 3.2 Diagram Kontek Aplikasi Billing Internet client server

Keterangan dari diagram kontek pada gambar 3.2 menjelaskan, terdapat empat *users* yang terdiri dari Administrator, operator, anggota dan non anggota yang menggunakan atau memproses aplikasi billing internet client server. Masing-masing *users* berbeda dalam mengakses aplikasi dan mendapatkan informasi pada sistem aplikasi.

B. Diagram Alir Data (*Data Flow Diagram*)

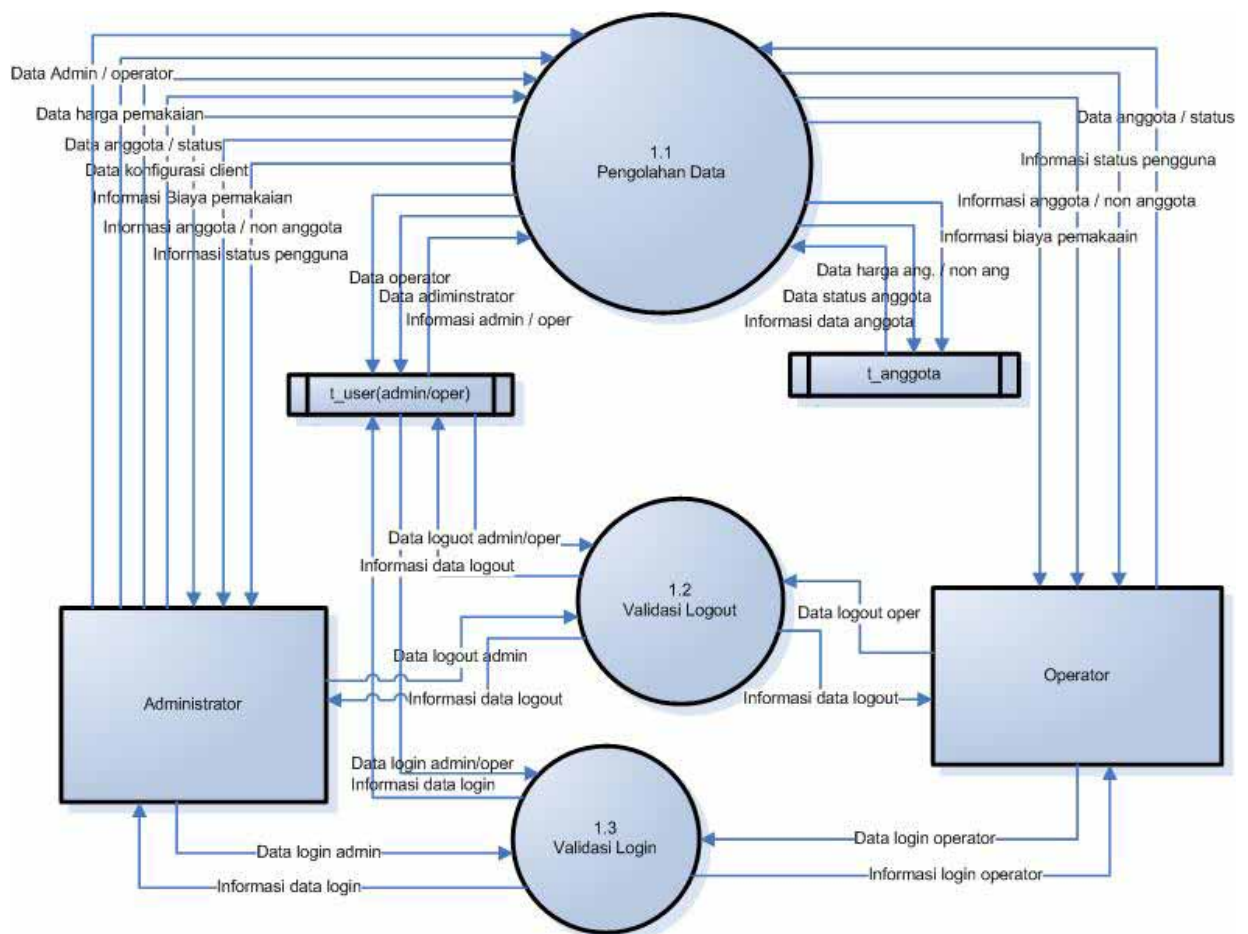
1. Diagram Alir Data Tingkat – 1



Gambar 3.3 DAD Tingkat -1 pada aplikasi Billing Internet Client Server

Pada DAD tingkat-1 penggambaran masing-masing proses dapat diperkecil lagi sehingga terdapat dua komponen yaitu pengaturan data server dan pengaturan data client seperti di tunjukan pada gambar 3.3

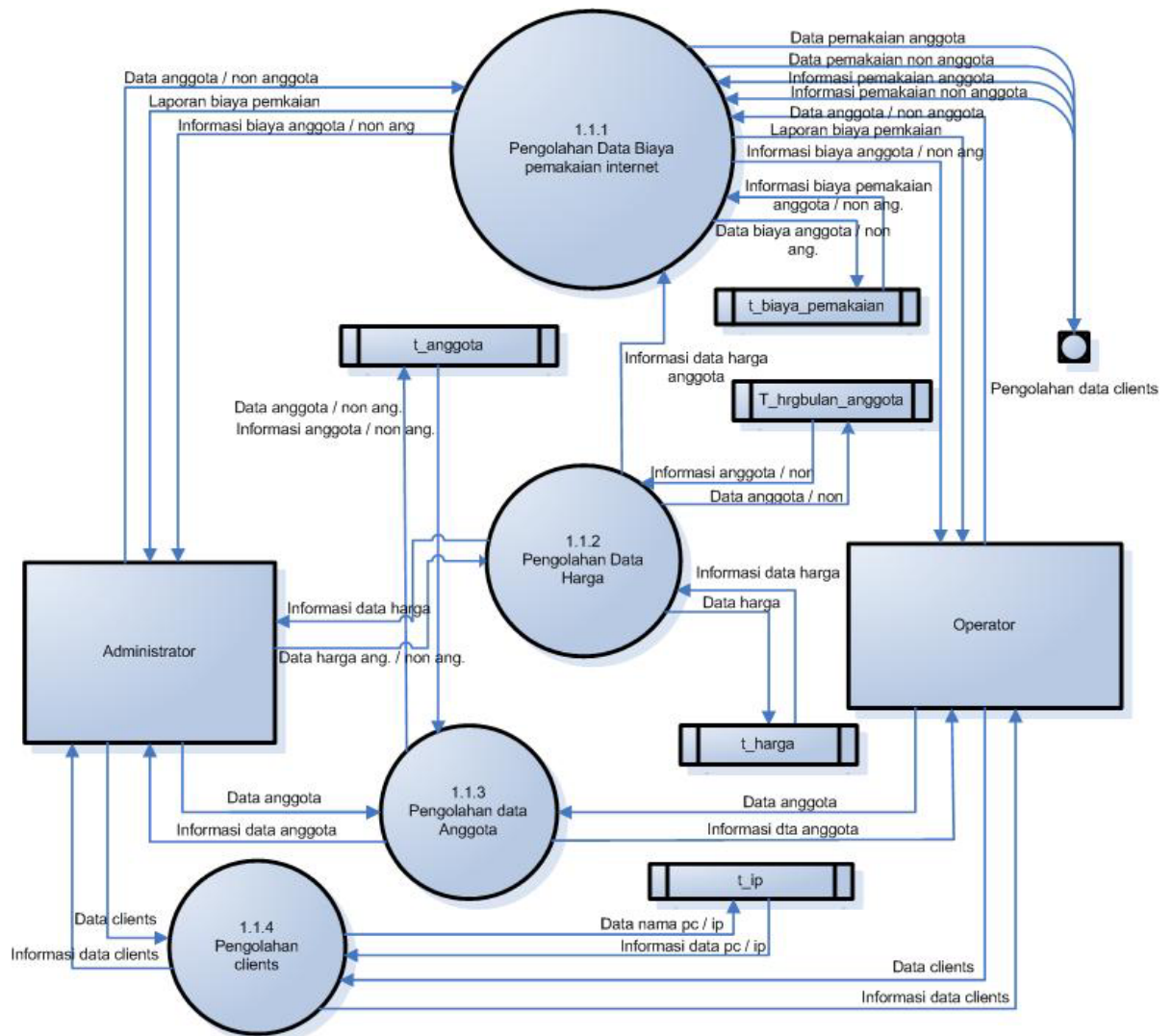
2. Diagram Alir Data Tingkat – 2



Gambar 3.4 DAD tingkat - 2 pada aplikasi Billing Internet Client Server

Keterangan dari DAD tingkat - 2 pada gambar 3.4 menjelaskan, terdapat tiga proses yaitu pengolahan data, validasi logout, dan validasi login. Di masing-proses memiliki alur dalam mengirim informasi dan mendapatkan data yang berbeda pada setiap pengguna.

3. Diagram Alir Data Tingkat – 3



Gambar 3.5 DAD tingkat 3 pada aplikasi Billing Internet Client Server

D. Kamus Data (*Data Dictionary*)

1. T_Admin = Nomor + Nama + @Password
2. T_Operator = Nomor + Nama + @Password
3. T_Anggota = Nomor + Nama + @Password + Tanggal_Daftar +
Tanggal_Akhir + Status
4. T_IP = @Nomor + IP
5. T_User = @Nomor + Nama
6. T_Harga = Nomor + @Jenis + Biaya_Per_Koneksi + Biaya_Per_Menit
7. T_Biaya_Pemakaian = @Nomor + Nama + Jenis + Mulai + Selesai + Durasi
+ Tot_Biaya + Tanggal + Operator

E. Spesifikasi Proses

Spesifikasi proses dari DAD tingkat - 3 pada gambar 3.5 menjelaskan :

1. Proses 1.1.1 Pengolahan biaya pemakaian internet

Administrator maupun Operator melakukan pengolahan data pemakaian internet yang data tersebut di peroleh dari informasi pengolahan data *clients* dan disimpan ke dalam file *t_biaya_pemakaian*, yang nantinya data tersebut di proses kembali untuk mendapatkan informasi berupa tampilan yang dikeluarkan ke layar maupun laporan cetak harian maupun bulanan.

2. Proses 1.1.2 Pengolahan data harga

Pada proses ini Administrator dapat mengatur data harga pemakaian internet yang disimpan ke dalam file *t_harga* dan juga mengatur data harga anggota perbulan yang disimpan ke dalam file *t_hrganggota_bulan*.

3. Proses 1.1.3 Pengolahan data anggota dan *non* anggota

Administrator maupun operator dapat menyimpan data anggota dan disimpan ke dalam file *t_anggota* yang nantinya data tersebut di proses kembali untuk mendapatkan informasi data anggota dan data tersebut dikeluarkan ke layar

4. Proses 1.1.4 Pengolahan *clients*

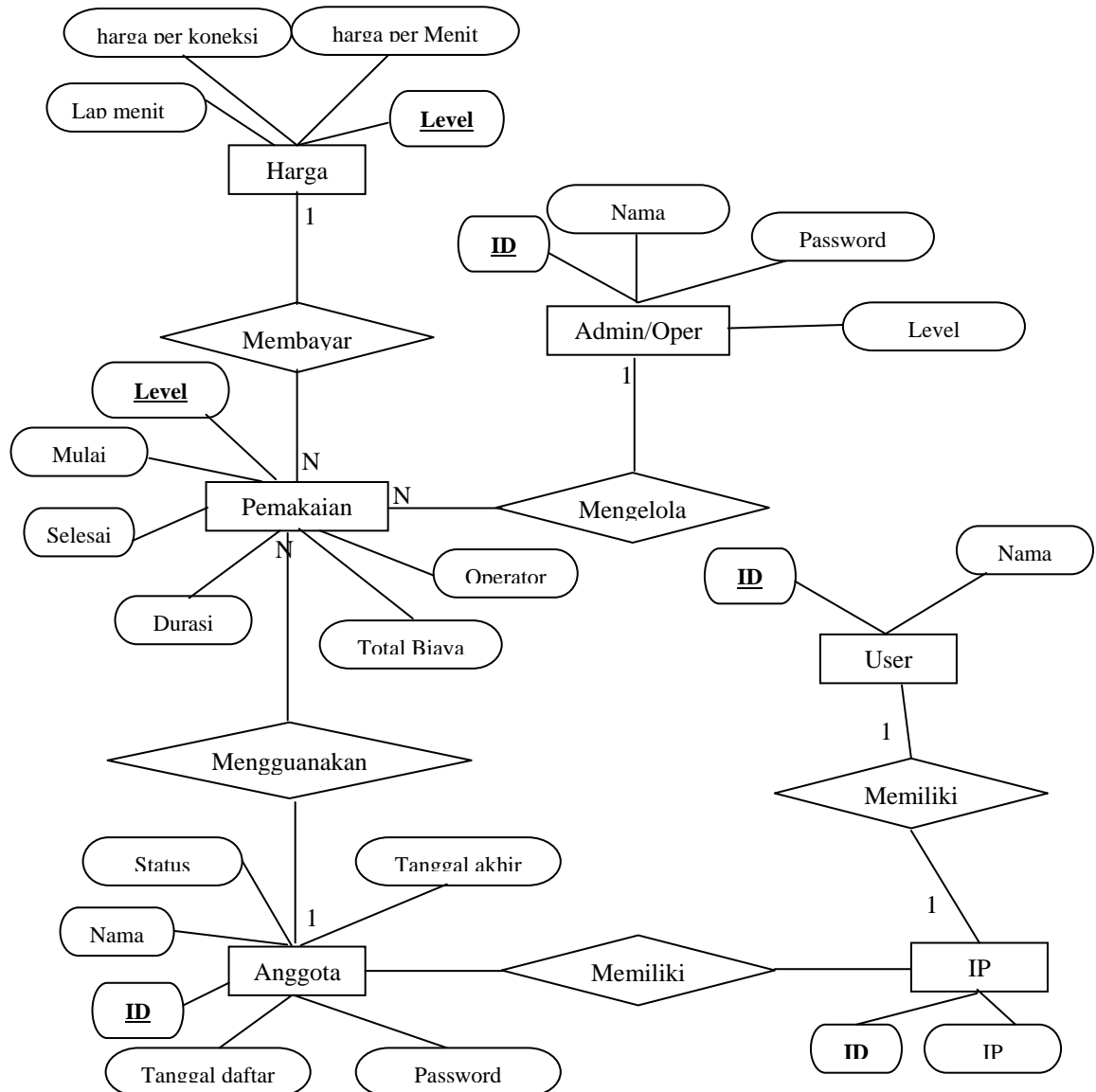
Sistem pada server akan memberikan pengaturan shutdown, restart dan stand by pada client berdasarkan Ip client yang di ambil dari file *t_ip*.

5. *Pengolahan data clients*

Sistem Server mengirimkan data pemakaian anggota dan pemakaian non anggota ke sistem client dan diproses oleh sistem client kemudia dikembalikan lagi ke sistem server berupa informasi data anggota dan non anggota.

3.1.4.3 ER Diagram

Berikut adalah perancangan ER Diagram yang ditunjukkan pada gambar 3.6



Gambar 3.6 ER Diagram

Pada gambar 3.6 menjelaskan hubungan antara beberapa tabel – tabel atau entity yang mempunyai beberapa atribut dengan mendefinisikan key dari beberapa atribut.

3.1.4.4 Deskripsi Tabel

A. Tabel Harga

Fungsi : Menyimpan Data Harga

Primary Key : level

Foreign Key : -

Tabel 3.4 Tabel t_harga

No	Nama Field	Jenis	Lebar	keterangan
1	level	Varchar	15	Jenis <i>account</i>
2	hrgperkoneksi	Int	11	Biaya per koneksi
3	lapmenit	Int	11	Lama menit
4	hrgperlapmenit	Int	11	Biaya per lama menit

B. Tabel Anggota

Fungsi : Menyimpan Data Anggota

Primary Key : ID

Foreign Key : -

Tabel 3.5 Tabel t_anggota

No	Nama Field	Jenis	Lebar	keterangan
1	ID	Int	6	Nomor urut
2	nama	Varchar	12	Nama anggota
3	level	Varchar	30	Jenis <i>account</i>
4	password	Varchar	20	Password anggota
5	tgldaftar	Date		Tanggal mulai daftar
6	tglakhir	Date		Tanggal akhir keanggotaan
7	status	Varchar	10	Anggota / non anggota
8	bulan	Int	2	bananyak bulan
9	harga	Int	10	Harga per bulan

C. Tabel Users (admin/operator)

Fungsi : Menyimpan Data Users Adiministrator dan operator

Primary Key : ID

Foreign Key : -

Tabel 3.6 Tabel t_users

No	Nama Field	Jenis	Lebar	keterangan
1	ID	Int	11	No urut user
2	nama	Varchar	30	Nama user
3	level	Varchar	15	Jenis <i>account</i>
4	password	Varchar	20	Password user

D. Tabel IP

Fungsi : Menyimpan Data IP dan Nama PC

Primary Key : PC

Foreign Key : -

Tabel 3.7 Tabel t_ip

No	Nama Field	Jenis	Lebar	keterangan
1	PC	Varchar	8	Nama pc
2	IP	Varchar	16	Ip address
3	STATUS	Varchar	15	Status pada pc

E. Tabel Biaya Pemakaian

Fungsi : Menyimpan Data biaya pemakaian internet

Primary Key : ID

Foreign Key : -

Tabel 3.8 Tabel t_biaya_pemakaian

No	Nama Field	Jenis	Lebar	keterangan
1	ID	Int	6	No urut
2	level	Varchar	15	Jenis <i>account</i>
3	nopc	Varchar	8	No pc
4	nama	Varchar	30	Nama pengguna
5	mulai	Time		Mulai pemakaian
6	selesai	Time		Selesai pemakaian
7	durasi	Time		Lama durasi
8	totalbiaya	Int	6	Total biaya pemakain
9	tanggal	Date		Tanggal pemakaain
10	operator	Varchar	30	Nama operator

F. Tabel Harga Bulan Anggota

Fungsi : Menyimpan Data Harga perbulan Anggota

Primary Key : level

Foreign Key : -

Tabel 3.9 Tabel t_hrgblnanggota

No	Nama Field	Jenis	Lebar	keterangan
1	level	Varchar	15	Jenis <i>account</i>
2	harga	Int	5	Harga keanggotaan perbulan

G. Tabel Temp

Fungsi : Menyimpan Data sementara

Primary Key : -

Foreign Key : -

Tabel 3.10 Tabel temp

No	Nama Field	Jenis	Lebar	keterangan
1	temp	Int	6	Tempat penyimpanan sementara untuk pc client

3.2 Perancangan Sistem Informasi

3.2.1 Perancangan Prosedur

Perancangan prosedur pelaksanaan pekerjaan yang baru dengan menyertakan komputer sebagai alat bantu proses pengolahan datanya.

3.2.1.1 Deskripsi Prosedur Kerja (Baru)

Prosedur kerja yang akan diberlakukan pada saat user menggunakan internet pada warung internet adalah sebagai berikut :

1. User melakukan login pada aplikasi client berdasarkan 3 pilihan, yaitu :

- Login tanpa limit waktu atau limit biaya.
- Login dengan limit waktu.
- Login dengan limit biaya.

2. Biaya akan dihitung sesuai dengan ketentuan sebagai berikut :

• **Data yang dibutuhkan :**

Lama pemakaian = ? menit

Lap per-menit = ? menit

Harga per-koneksi = ? Rp

Harga per lap menit ? Rp

• **Data yang di tanyakan :**

N lap per menit (menit)

Harga / total biaya (Rp)

Rumus :

- $N \text{ lap per-menit} = \text{lama pemakaian} / \text{lap permenit}$
- $\text{Total biaya} = \text{harga per koneksi} + (\text{harga permenit} \times N \text{ lap permenit})$

• **Misal data diset adalah :**

Lap per-menit = 10 menit

Harga perkoneksi = 1000 (Rp)

Harga per lap menit = 350 (Rp)

Contoh 1 :

Lama pemakaian = 93 menit

Jawab.

$N \text{ lap permenit} = \text{div} (\text{lama pemakaian} / \text{lap permenit})$

$N \text{ lap permenit} = \text{div} (90 / 10) = \text{div } 9.3 = 9$

Total biaya = harga per koneksi + (harga permenit x N lap permenit)

Total biaya = 1000 + (350 x 9) = 1000 + 3150

Total biaya = 4150 (Rp)

Contoh 2

Lama pemakaian = 7 menit

Jawab.

N lap permenit = div (lama pemakaian / lap permint)

N lap permenit = div (7 / 10) = div 0.5 = 0

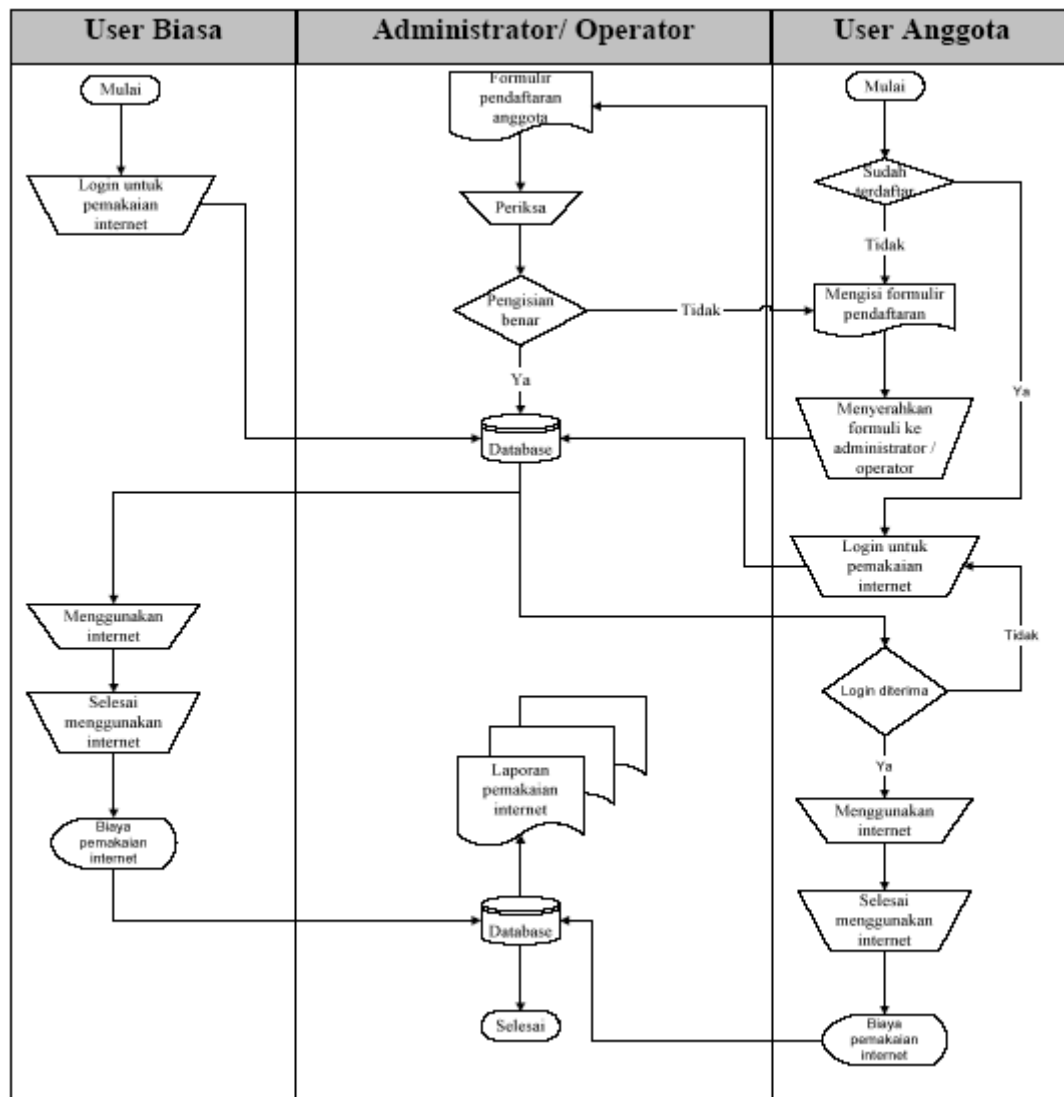
Total biaya = harga per koneksi + (harga permenit x N lap permenit)

Total biaya = 1000 + (350 x 0) = 1000 + 0

Total biaya = 1000 (Rp)

3. Jumlah biaya koneksi dan biaya per menit ditentukan oleh administrator sesuai dengan jenis user (anggota/ non anggota).
4. Jika user login tanpa menyertakan limit waktu, untuk menghentikan pemakaian internet, user harus melakukan logout dari aplikasi.
5. Untuk user yang melakukan dengan menggunakan limit waktu atau limit biaya, aplikasi akan memberikan validasi pesan dan dalam waktu 5 menit otomatis me-logout user saat batas limit waktu atau limit biaya tercapai.
6. Setelah menggunakan aplikasi, user membayar biaya pemakaian internet ke administrator atau operator.
7. Saat user menghentikan pemakaian internet, data-data user yang berkaitan dengan pemakaian internet akan secara otomatis disimpan ke dalam database.
8. Operator atau administrator bisa melakukan pengelolaan data baik yang berhubungan dengan data biaya, data anggota dan lain sebagainya, dengan menggunakan aplikasi yang terdapat pada komputer server.

Deskripsi prosedur kerja baru digambarkan sebagai berikut :



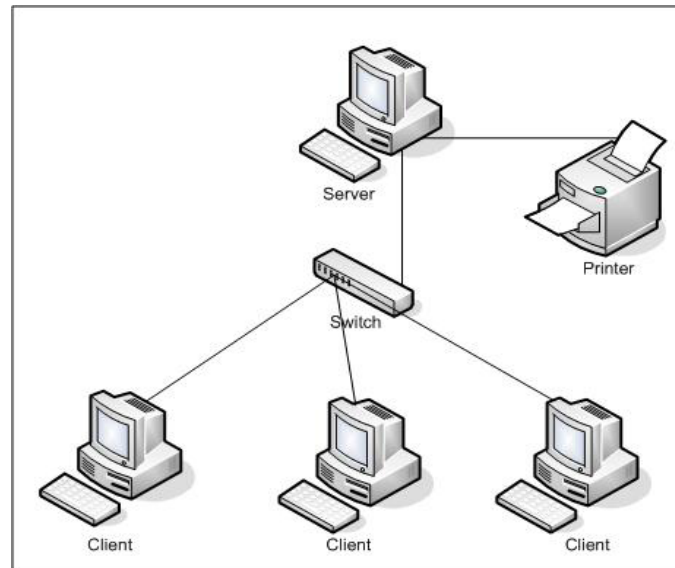
Gambar 3.7 Deskripsi Prosedur Kerja Baru

3.2.2 Perancangan Perangkat Keras

3.2.2.1 Konfigurasi Perangkat Keras

Jaringan komputer dengan topologi star atau topologi lainnya yang menggunakan protokol TCP/ IP sebagai standar protokol komunikasinya, yang menyediakan satu komputer yang bertindak sebagai komputer server dimana

nantinya aplikasi server dijalankan, serta beberapa komputer *workstation* yang bertindak sebagai komputer *client*, dimana nantinya aplikasi *client* dijalankan.



Gambar 3.8 Konfigurasi Perangkat Keras

Struktur pengalamatan *IP address* untuk protokol TCP/ IP pada konfigurasi jaringan di atas menggunakan format *IP address* untuk kelas C, yang memiliki range dari 192.0.0.xxx sampai 223.255.255.xxx. Pada saat implementasi Billing Internet Berbasis Client Server Menggunakan Visual Studio 2005, *IP address* yang digunakan adalah :

Server : 192.168.0.1

Client : 192.168.0.2

Jumlah maksimal client untuk Billing Internet Berbasis Client Server Menggunakan Visual Studio 2005 adalah sebanyak 10 client, sehingga *range IP Address* untuk client adalah dari 192.0.0.0 - 192.0.0.10.

3.2.2.2 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras pada saat pembuatan Billing Internet Berbasis

Client Server Menggunakan Visual Studio 2005 adalah sebagai berikut :

A. Server

- Processor 3000 MHz
- Memory DDR 1 Gb
- Hardisk 80 Gb
- LAN CARD 100 Mbps

B. Client

- Processor 1700 MHz
- Memory DDR 256 Mb
- Hardisk 20 Gb
- LAN CARD 100 Mbps

3.2.2.3 Spesifikasi Perangkat Lunak Sistem

Tabel 3.11 Spesifikasi Perangkat Lunak Sistem

No.	Jenis Perangkat Lunak Sistem	Server	Client
1	Sistem Operasi	Windows Xp Professional Sp2	Windows Xp Home Edition
2	Perangkat Implementasi	<ul style="list-style-type: none"> • Microsoft Visual Studio 2005 • Mysql connection (dll) 	
3	DBMS	MySQL 5.1	
4	Perangkat Lunak Pendukung	Sqlyog Enterprise – Mysql GUI v7.02	

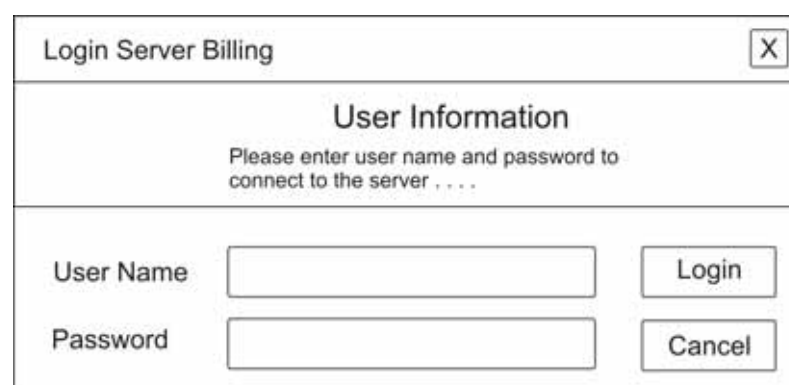
3.2.2.4 Perancangan Perangkat Lunak

Agar user dapat berkomunikasi atau berinteraksi dengan aplikasi yang dibangun, maka diperlukan interface (antarmuka) program dari aplikasi tersebut. Berikut rancangan dari interface form login yang dimaksud dapat dilihat pada gambar 3.9 dan gambar 3.10



The image shows a client login menu interface. It features a white background with a thin black border. At the top, there is a header area. Below the header, the form contains the following elements: a text label 'User name' followed by a rectangular input field; a text label 'Durasi' followed by four radio button options: 'No Limit', '2 Jam', '1 Jam', and '30 Menit'; a checkbox labeled 'Status Anggota'; a text label 'Password' followed by a rectangular input field; and a 'Start' button centered at the bottom.

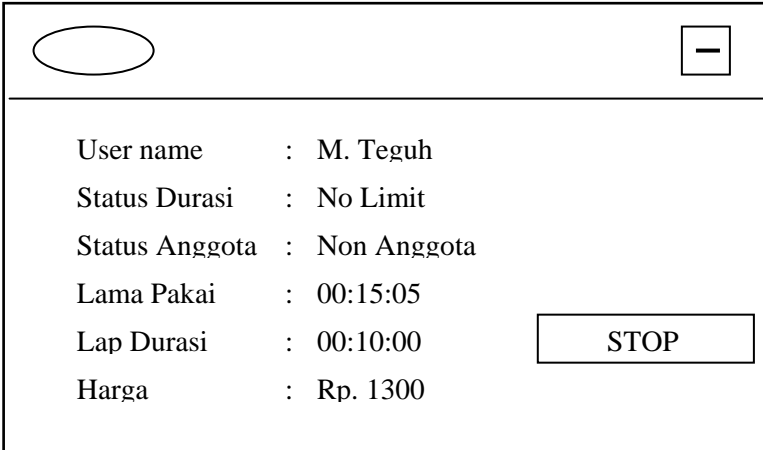
Gambar 3.9 Menu Login (*client*)



The image shows a server login menu interface. It is a window titled 'Login Server Billing' with a close button (X) in the top right corner. The window has a white background and a thin black border. The main content area is titled 'User Information' and contains the instruction: 'Please enter user name and password to connect to the server . . .'. Below this, there are two rows of input fields and buttons. The first row has a 'User Name' label, a rectangular input field, and a 'Login' button. The second row has a 'Password' label, a rectangular input field, and a 'Cancel' button.

Gambar 3.10 Rancangan Menu Login (*server*)

Pada Form login ini, penulis merancang bahwa ada dua hak *access* yang boleh masuk dalam sistem (*server*) yaitu Administrator dan Operator. Pemilihan ini biasanya dilakukan dengan menggunakan *login* Hak *access* dan *password*. Pertama yaitu login sebagai Operator, artinya user di berikan batasan dalam mengakses data seperti tidak diperbolehkan mengatur harga pemakaian internet, tetapi hanya boleh insert Anggota baru , *delete* dan *update* , yaitu dengan cara mengunci menu-menu tertentu. Kedua login sebagai Administrator, artinya pengguna yang di berikan *access* penuh dengan tidak ada batasan seperti mengatur harga pemakaian internet dan membuat *user name* dan *password* untuk operator maupun admin.

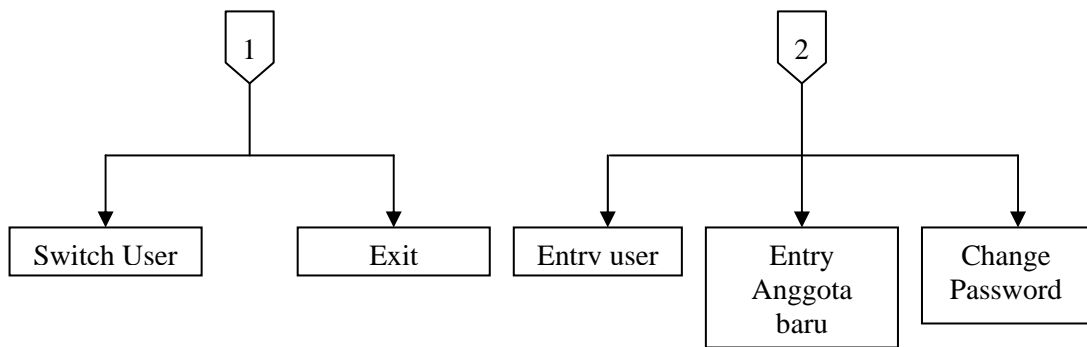


The image shows a window titled "Rancangan Menu Status Pemakaian (client)". The window contains a list of user details and a "STOP" button. The details are as follows:

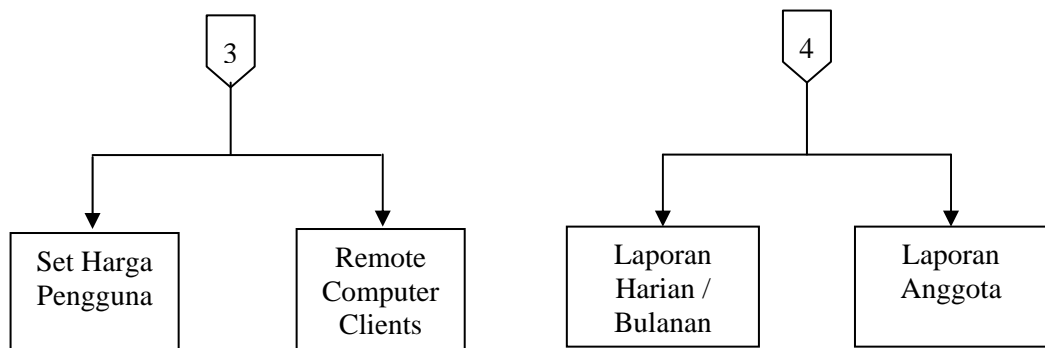
User name	: M. Teguh
Status Durasi	: No Limit
Status Anggota	: Non Anggota
Lama Pakai	: 00:15:05
Lap Durasi	: 00:10:00
Harga	: Rp. 1300

A "STOP" button is located to the right of the "Lap Durasi" field.

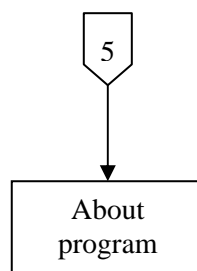
Gambar 3.11 Rancangan Menu Status Pemakaian (*client*)



Gambar 3.13 Struktur Menu (lanjutan 1)

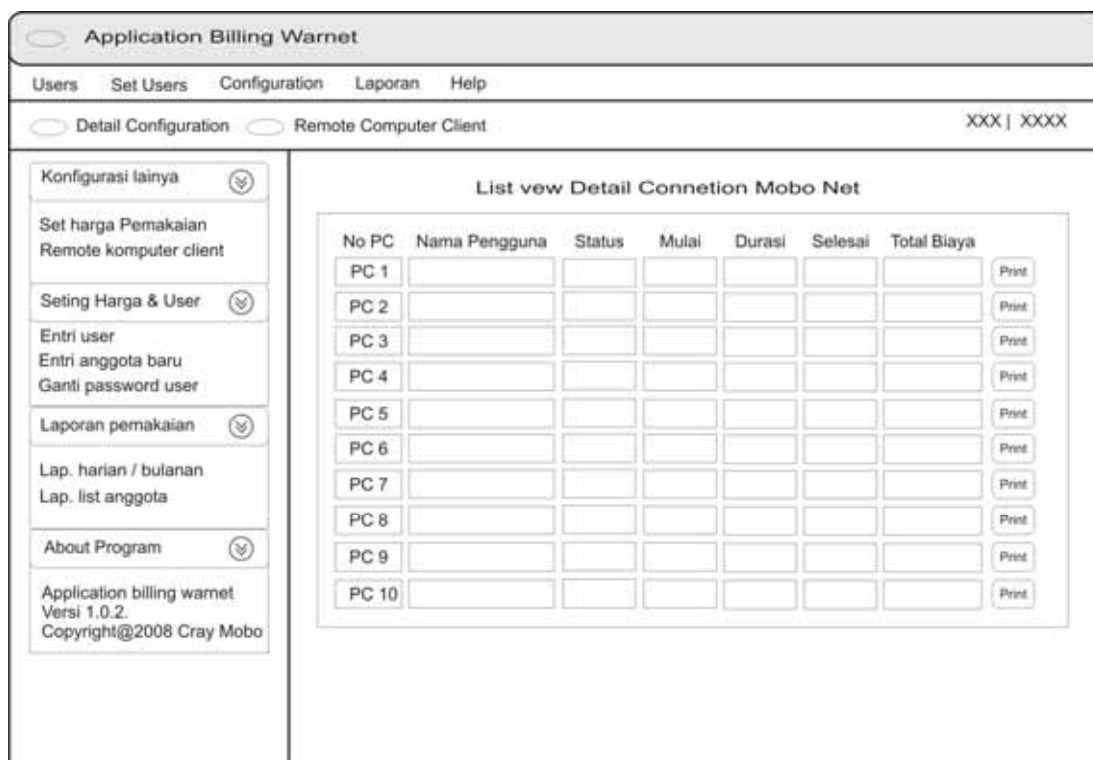


Gambar 3.14 Struktur Menu (lanjutan 2)



Gambar 3.15 Struktur Menu (lanjutan 3)

Struktur menu utama dari aplikasi Billing Internet *client server* di tunjukan oleh Gambar 3.12 dan untuk rancangan menu utamanya di tunjukan oleh gambar 3.16



Gambar 3.16 Rancangan Menu Utama

Seperti terlihat pada gambar 3.16 pada layar menu utama terdapat 5 bagian utama, yaitu : Users, Set Users, Konfigurasi, Laporan, dan Help. Dan pada bagian samping pada *box panel* terdapat juga 4 Tools Utama yaitu : Konfigurasi lainnya, Seting harga & users, Laporan pemakaian, dan About program. Menu utama dan *Box Panel* kegunaan dari keduanya sama, hanya saja user diberikan pilihan dalam menggunakan menu utama atau *box panel*.

1. Konfigurasi (*configuration*)

Menu Konfigurasi pada aplikasi ini merupakan pengaturan untuk komputer client Apabila kita klik bagian utama menu Konfigurasi maka akan keluar sub-menu, yaitu : Set Harga Pengguna dan Remote Computer Clients. Pada sub menu Remote computer clients, administrator maupun operator dapat melakukan shutdown, restart dan stand by client berdasarkan nomor atau nama pc clients

a. Entri harga warnet dan anggota

Digunakan untuk mengkonfigurasi harga pemakaian billing internet dari harga anggota dan harga non anggota serta juga harga pendaftaran anggota per bulan. Form tersebut hanya dapat di access oleh Administrator, yang di tunjukan oleh gambar 3.17

Gambar 3.17 Rancangan Konfigurasi harga warnet dan anggota

2. Laporan

Berfungsi untuk menampilkan data pemakaian internet yang dapat di print berdasarkan bulanan maupun harian atau tanggal yang dipilih. Gambar 3.18 menunjukkan rancangan dari laporan pemakaian harian

No	Status	NoPc	Nama	Mulai	Selesai	Dura	Total	Tanggal	Operato
1	Anggota	PC4	M. Teguh	12:44	13:45	01:05	3200	12-11-2005	mobo
x	xxxx	xxx	xxxx	xxxx	xxxx	xx	xxxx	xxxx	xxxx
x	xxxx	xxx	xxxx	xxxx	xxxx	xx	xxxx	xxxx	xxxx

Halaman : xx

Gambar 3.18 Rancangan Laporan pemakaian harian

3. Help

Apabila kita klik bagian menu utama Help maka akan keluar sub-menu About Program yang berfungsi untuk mengetahui asal atau versi dari aplikasi tersebut dan juga berfungsi untuk menghubungi email dari pembuat aplikasi.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dijelaskan tentang pengimplementasian program dan serangkaian uji coba yang dilakukan terhadap aplikasi *server* dan aplikasi *client*. Dengan melakukan serangkaian uji coba maka akan diketahui apakah aplikasi ini bisa berjalan dengan baik.

Pada tahapan implementasi terdapat empat cakupan yaitu Lingkungan implementasi, konfigurasi *client server* dan implementasi proses.

4.1. Lingkungan Implementasi

Lingkungan implementasi terdapat dua cakupan yaitu Lingkungan perangkat lunak dan lingkungan perangkat keras :

1. Lingkungan perangkat keras (*hardware*)

Lingkungan perangkat keras dapat ditunjukkan pada tabel 4.1

Tabel 4.1 Lingkungan perangkat keras

Komputer	Konfigurasi Perangkat keras
<i>Server</i>	<ol style="list-style-type: none"> 1. Prosesor AMD Athlon 64 bit 3000 Hz 2. Memori 1 Ghz Pc 3200 (<i>shared vmware</i> 512 Mb) 3. Monitor 17" dengan Resolusi layar 1024 x 768 fixel 4. Harddisk berkapasitas 80 GB Sata 5. DVD ROM Samsung 52 x max 6. MotherBoard MSI RS4820 7. VGA ATI X550 memory 512 mb 128bit 8. Printer hp deskjet 3920
<i>Client</i>	<ol style="list-style-type: none"> 1. Prosesor AMD Athlon 64 bit 3000 Hz 2. Memori 512 Mb Pc 3200 3. Monitor 17" dengan Resolusi layar 1024 x 768 fixel 4. Harddisk berkapasitas 80 GB Sata 5. DVD ROM Samsung 52 x max 6. MotherBoard MSI RS4820 7. VGA ATI X550 memory 512 mb 128bit

2. Lingkungan perangkat lunak (*Software*)

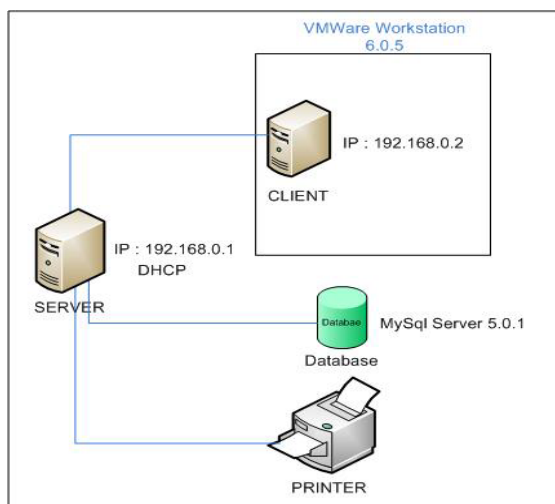
Lingkungan perangkat lunak dapat ditunjukkan pada tabel 4.2.

Tabel 4.2 Lingkungan perangkat lunak

Komputer	Konfigurasi Perangkat Lunak
<i>Server</i>	<ol style="list-style-type: none"> 1. Sistem operasi : Microsoft Windows XP SP2 Gold 2. Visual Studio 2005 Professional 3. Database MySql 5.0.1 4. Laporan Crystal Report 10
<i>Client</i>	<ol style="list-style-type: none"> 1. Sistem operasi : Microsoft Windows XP SP2 Gold 2. VMware workstation 6.0.5 3. .Net framework 2.0

4.2 Konfigurasi *Client Server*

Perangkat lunak yang akan penulis gunakan dalam implementasi koneksi *client server* adalah dengan menggunakan Vmware workstation 6.0.5 pada komputer *client*, pada gambar 4.1 menunjukkan skema pengguna billing internet :



Gambar 4.1 Skema Implementasi pengguna pada Billing internet dengan Vmware workstation

Struktur pengalaman IP address untuk protokol TCP/ IP pada konfigurasi jaringan di atas menggunakan format IP address untuk kelas C, yang memiliki range dari 192.0.0.xxx sampai 223.255.255.xxx. Pada saat implementasi Billing Internet Berbasis *Client Server* Menggunakan Visual basic 2005, IP address yang digunakan adalah :

Server : 192.168.0.1 – Port 9900

Client : 192.168.0.2 – Port 9900

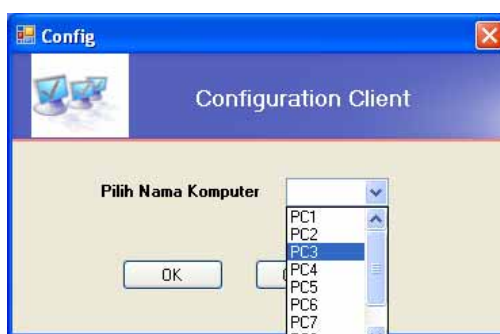
Jumlah maksimal client untuk Billing Internet Berbasis Client Server Menggunakan Visual basic 2005 adalah sebanyak 10 client, sehingga range IP Address untuk *client* adalah dari 192.168.0.2 - 192.0.0.11 dengan menggunakan nomor port 9900.

Program *client* akan mengirim laporan login/logout ke server dan pada waktu yang bersamaan program *server* akan menerima data dari client dan menyimpannya ke database, proses tersebut akan dijelaskan pada sub bab 4.3.4.

Konfigurasi yang dilakukan saat menjalankan aplikasi client pertama adalah mengganti nama komputer *client* dengan mengklik tombol config pada bawah kanan pada aplikasi client dan mengisi password dengan username sebagai admin maupun operator. Restart komputer client setelah konfigurasi dilakukan. Dapat di tunjukan pada gambar 4.2 dan 4.3.



Gambar 4.2 Konfirmasi login pada *client*



Gambar 4.3 Ganti nama komputer *client*

Aplikasi *client* akan menyimpan data nomor pc dan alamat ip ke dalam *database*. Pencarian alamat ip di lakukan secara otomatis pada saat program client berjalan.

Pada aplikasi *server* tidak ada konfigurasi khusus dalam melakukan koneksi *client server*.

4.3 Implementasi Proses

Dalam membuat Aplikasi untuk melakukan komunikasi antar komputer berbeda penulis menggunakan *library socket namespace* dengan memanfaatkan *system.net* dan *system.net.socket*. Pada pengolahan database penulis menggunakan teknologi MySqlADO.Net data provider yaitu teknologi yang dikembangkan untuk mengakses database *mysql* dengan menggunakan Visual basic.net. Untuk mengakses *query* dari basis data *mysql* seperti menampilkan data maupun manipulasi data pada aplikasi billing internet berbasis *client server*, penulis menggunakan komponen *mysql connector.net*.

4.3.1 Proses Koneksi Basis Data

Berikut ini adalah proses koneksi basis data yang ditunjukkan pada gambar

4.4

```
Imports MySql.Data.MySqlClient
Public Class ClassConnect

    Public CnnStr As String = _
        "server=localhost;uid=root;pwd=jasmine;database=dwarnet;
Port=3306;" & _
        "Protocol=TCP;Compress=false;"
    Public Shared Cnn As MySqlConnection
    Public Sub CnnOpen()
        Cnn = New MySqlConnection(CnnStr)
        Cnn.Open()
    End Sub
    Public Sub CnnClose()
        Cnn = New MySqlConnection(CnnStr)
        Cnn.Close()
    End Sub
```

Gambar 4.4 Source Code Class yang berobjek Classconnect

Pada gambar 4.4, Classconnect merupakan nama *class* yang bertujuan untuk mengkoneksikan database mysql, variabel Public Cnnstr(*connection string*) bertujuan untuk memberikan koneksi berupa *string* yang nantinya akan dibaca oleh *MysqlConnetion* pada aplikasi ini bervariasi *Public Cnn*. Perintah *Cnn.open* pada aplikasi ini adalah untuk membuka atau mengeksekusi database setelah Cnn sudah di inisialisasi.

4.3.2 Proses Penghitungan Harga dan Waktu

Penggalan kode yang bertujuan untuk membaca data harga dalam *database* dan digunakan untuk penghitungan waktu pemakaian pada billing warung internet seperti ditunjukkan pada gambar 4.5

```

Private Sub Getharga()
    Dim param As MySqlParameter
    Cmd.Connection = objcon.Cn
    Cmd.CommandType = CommandType.StoredProcedure
    Cmd.CommandText = "sp_harga"
    param = New MySqlParameter("plevel",
lblanggota.Text)
    Cmd.Parameters.Add(param)
    If lblanggota.Text = "Anggota" Then
        Try
            objcon.CNOpen()
            DR = Cmd.ExecuteReader
            While DR.Read
                PKoneksi =
DR("hrgperkoneksi").ToString
                PMenit =
DR("hrgperlapmenit").ToString
                pLapMenit = DR("lapmenit").ToString
            Cmd.Dispose()
            End While
        Catch ex As Exception
            MsgBox("ERR : " & ex.Message)
        Finally
            DR.Close()
            objcon.Cnclose()
        End Try
    Else

```

Gambar 4.5 Source Code membaca data harga dalam database

Pada pernyataan :

```
Cmd.CommandText = "sp_harga"
```

Yang ditunjukkan pada gambar 4.5 yaitu memanggil *store procedure* pada mysql bernama *sp_harga*. *Query Sp_harga* ditunjukkan pada gambar 4.6

```
DROP PROCEDURE IF EXISTS `dwarnet`.`sp_harga`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_harga`(in plevel
  varchar(12))
BEGIN
  select
    t_harga.hrgperkoneksi,t_harga.hrgperlapmenit,t_harga.lapmenit
  from t_harga
  where t_harga.level=plevel;
  END$$

DELIMITER ;
```

Gambar 4.6 *Query store procedure sp_harga*

Gambar 4.6 menjelaskan menampilkan seluruh field pada tabel *t_harga* dimana *level = plevel*. *Plevel* adalah parameter masukan yang digunakan untuk membedakan status non anggota dan anggota. *Plevel* juga sebagai pembeda harga pada non anggota dan anggota.

```
param = New MySqlParameter("plevel", lblanggota.Text)
```

Variabel *param* diinisialisasi sebagai parameter mysql dan dimasukan data dengan data bertipe *string* yang terdapat pada *lblanggota.text*.

Pada gambar 4.7 adalah kode yang bertujuan untuk melakukan proses penghitungan waktu pemakaian pada billing warung internet :

```

Private Sub lap()
    If sw.IsRunning Then
        paused = True
    End If
    Total = Total + PMenit
    LblHarga.Text = Total
    message.msgharga.Text = "Total Biaya : Rp." &
Me.LblHarga.Text & ".-"
    objcon.CNOpen()
    UpdateLap()
End Sub
Private Sub permenit()
    Timer3.Interval = pLapMenit * 60000
End Sub

```

Gambar 4.7 Source Code Class waktu pemakaian billing

Penghitungan waktu dengan memanfaatkan fungsi *stopwatch*, private sub *lap()* akan *running* satu kali per *lapmenit* sesuai dengan set pada *plapmenit* pada program *server*. Pada saat private sub *lap()* *running* maka akan menghitung $total = total + pmenit$. *Pmenit* adalah harga perlapmenit.

4.3.3 Proses Koneksi Client Server

Dalam melakukan komunikasi antar komputer atau device lainnya dibutuhkan *socket programming*. Penggalan code program untuk melakukan koneksi pada aplikasi *client* dapat ditunjukkan oleh gambar 4.8.

```

Public Sub Connect()
    Try
        ipHostinfo = Dns.Resolve(FrmDepan.IPCLIENT)
        ipAdd = ipHostinfo.AddressList(0)
        remoteEP = New IPEndPoint(ipAdd, 9900)
        sender.Connect(remoteEP)
    Catch ex As Exception
        MessageBox.Show("Connection ERR : " & ex.Message)
        Exit Sub
    End Try
End Sub

```

Gambar 4.8 Source Code koneksi pada client

Penggunaan *socket* untuk jaringan *tcp/ip* pada .net framework dapat menggunakan class *tcp.listener* berikut ini code untuk membuat *socket client* :

```
remoteEP = New IPEndPoint(ipAdd, 9900)
```

Dimana *ipadd* adalah *ip server* dengan format dot(.) contoh “192.168.0.1” dan 9900 adalah port *server* yang digunakan oleh *server* untuk proses *listen*.

```
Sender.Connect(remoteEP)
```

Adalah proses untuk melakukan komunikasi dengan *server*. Penggalan kode untuk melakukan koneksi pada aplikasi *server* dapat ditunjukkan pada gambar 4.9

```
Public Sub Connect()
    Try
        ipHostinfo = Dns.Resolve(Dns.GetHostName())
        ipAdd = ipHostinfo.AddressList(0)
        localEndPoint = New IPEndPoint(ipAdd, 9900)
        listner.Bind(localEndPoint)
        listner.Listen(10)
        th = New System.Threading.Thread(AddressOf Acceptstart)
        th.Start()

    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
```

Gambar 4.9 Source Code koneksi pada server

Aplikasi *server* berusaha untuk berkomunikasi dengan 10 komputer *client* menggunakan multi treading :

```
listner.Listen(10)
```

Untuk memulai proses listener dapat menggunakan method start().

4.3.4 Proses Pengiriman data

Terdapat dua proses pengiriman data pada *clinet server* yaitu :

1. Pengiriman dan penerimaan pesan pemakaian *client server*

Mengirimkan pesan dari komputer *client* ke *server* berupa nomor pc,.

Pada gambar 4.10 menjelaskan proses pengiriman data pada aplikasi *client*.

```

Public MSG As Byte()
. . .
Private Sub SendDurasi()
    Try
        MSG = Encoding.ASCII.GetBytes(FrmDepan.DPC)
        Me.sender.Send(MSG)
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub

```

Gambar 4.10 *Source Code* pengiriman data pemakaian pada *client*

Pada gambar 4.10 menjelaskan proses transfer data dari *client* ke *server*. Pada proses transfer data, semua data menggunakan dalam format byte. Jika data berupa string maka harus mengubah kebentuk byte dengan menggunakan method *System.Text.Encoding.ASCII.GetBytes*. berikut adalah contoh penggunaan :

```
MSG = Encoding.ASCII.GetBytes(FrmDepan.DPC)
```

Jika *FrmDepan.DPC* = "PC1" dan *i* = 1 (pada gambar 4.11) maka client mengirimkan pesan *pc1* telah berjalan untuk menghitung durasi waktu, secara *realtime server* yang dalam keadaan menunggu (*listen*) segera menjalankan proses penghitungan waktu. Dan sebaliknya jika *FrmDepan.DPC* = "PC1" dan *i* = 0 (pada gambar 4.11) maka client akan berhenti dan server menjalankan proses penghentian waktu serta menyimpan data pengguna.

Pengambilan pesan pada server dapat ditunjukkan pada gambar 4.11.


```

If (str = "PC1") Then
    n = 0
    If i1 = 0 Then
        Tmr1.Enabled = True
        Sw.Start()
        NamaPc = "PC1"
        PCRUN()
        i1 = 1
    Else
        i1 = 0
        tselesai(n).Text =DateTime.Now.ToString("HH:mm:ss")
        Tdurasi = TDurasil.Text
        NamaPc = "PC1"
        Objcon.CNOpen()
        PCSelesai()
        Sw.Stop()
        Tmr1.Enabled = False
    End If
End If

```

Gambar 4.11 Source Code Penerimaan pemakaian data pada server

2. Pengiriman dan penerimaan pesan untuk *remote* komputer *client*

Aplikasi server dapat melakukan *shutdown*, *restart* dan *logoff* pada komputer *client* berdasarkan nomor *pc client*. Kode pengiriman pesan untuk *remote* komputer dapat ditunjukkan pada gambar 4.12

```

Dim tcpCli As TcpClient = tcpList.AcceptTcpClient()

Try
    ' Send data to the client.
    Dim sw As New StreamWriter(ns)
    If rbShutdown.Checked = True Then
        sw.WriteLine("#Shutdown#")
    End If
    If rbReboot.Checked = True Then
        sw.WriteLine("#Reboot#")
    End If
    If rbLogOff.Checked = True Then
        sw.WriteLine("#Logoff#")
    End If
    sw.Flush()
Catch ex As Exception
    MsgBox(ex.Message)
End Try

```

Gambar 4.12 Source Code fungsi pengiriman data pada server

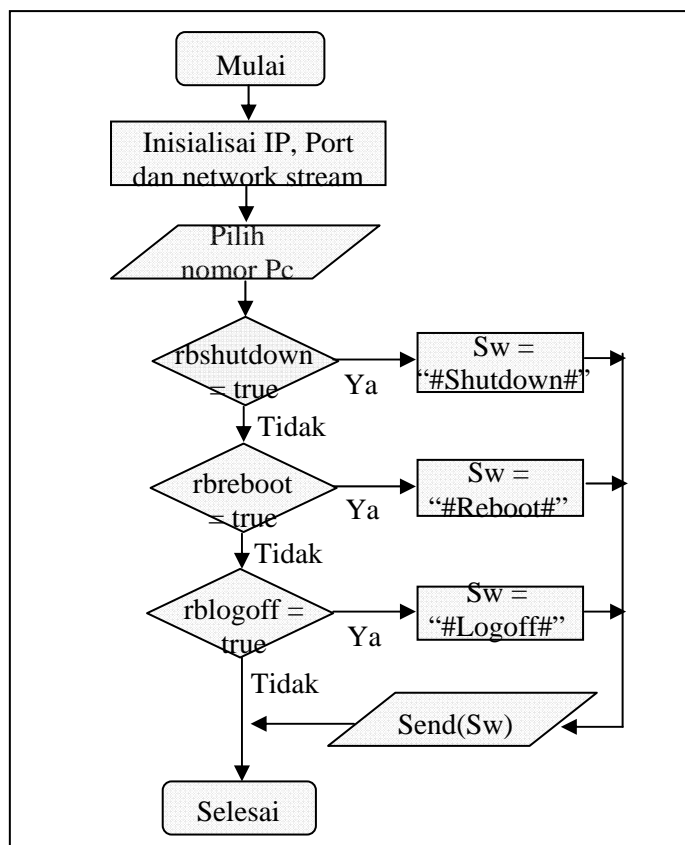
Aplikasi server dapat mengirim tiga pesan pilihan yaitu “#Shutdown#”, “#Reboot#” dan “#Logoff#” untuk meremot komputer *client*. Pengiriman pesan menggunakan jalur tcp/ip.

```
Dim tcpCli As TcpClient = tcpList.AcceptTcpClient()
Dim sw As New StreamWriter(ns)
```

Menginisialisakain sw untuk menulis pesan yang akan dikirim ke client.

`sw.Flush()` di gunakan untuk melakukan proses pengiriman pesan.

Flowchart dari proses pengiriman pesan untuk *remote* komputer pada client dapat ditunjukan oleh gambar 4.13



Gambar 4.13 *Flowchart* pengiriman data pada *server*

Rbshutdown, rbreboot dan rblogoff adalah tools radiobutton pada form remote computer client.

Gambar 4.14 menjelaskan kode untuk fungsi penerimaan pada komputer *client*.

```

Dim tcpCli As TcpClient = tcpList.AcceptTcpClient()
Dim ns As NetworkStream = tcpCli.GetStream
Dim sr As New StreamReader(ns)
    ` mengambil data dari server
Dim receivedData As String = sr.ReadLine()
If receivedData = "#Shutdown#" Then
    trShutdown = New Thread(AddressOf shutdown)trShutdown.Start()
End If
If receivedData = "#Reboot#" Then
    trReboot = New Thread(AddressOf reboot)trReboot.Start()
End If
If receivedData = "#Logoff#" Then
    trLogOff = New Thread(AddressOf logoff)trLogOff.Start()
End If

```

Gambar 4.14 *Source Code* fungsi penerimaan pada *client(remote)*

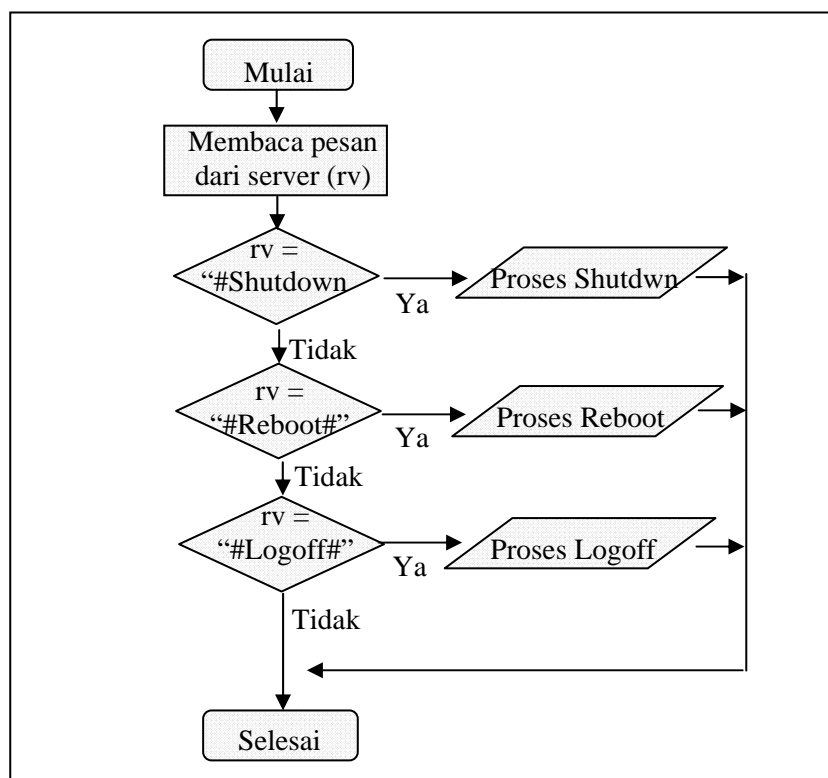
Komputer client menunggu untuk penerimaan data dari server, Dim `receivedData As String = sr.ReadLine()` Digunakan oleh *client* untuk menunggu pesan yang nantinya siap dibaca oleh client. Terdapat tiga pesan yang kemungkinan diterima oleh client yaitu “#Shutdown#”, “#Reboot#” dan “#Logoff#”.

Pernyataan kode :

```
trShutdown = NewThread(AddressOfshutdown)trShutdown.Start()
```

Adalah kode untuk proses shutdown komputer secara langsung.

Flowchart dari proses penerimaan pesan untuk *remote* komputer *client* pada *server* dapat ditunjukkan oleh gambar 4.15



Gambar 4.15 Flowchart pengiriman data pada client(remote)

4.4 Sistem interface

Pada saat users sebagai operator dan administrator menjalankan aplikasi akan tampil form login, tampilan pembuka dari program aplikasi adalah Login yang ditunjukkan pada gambar 4.16.

Gambar 4.16 Tampilan Login (server)



Gambar 4.17 Tampilan Login (*client*)

Pada gambar 4.17, pengguna dapat memilih durasi yang telah ditentukan, jika status pengguna adalah anggota, pengguna dapat melakukan ceklist pada status anggota dan mengisi password. Tombol start untuk memanggil ke form proses dan tombol config untuk menset nomor Pc *computer client*. Gambar 4.18 menunjukkan proses penghitungan aplikasi *client*.

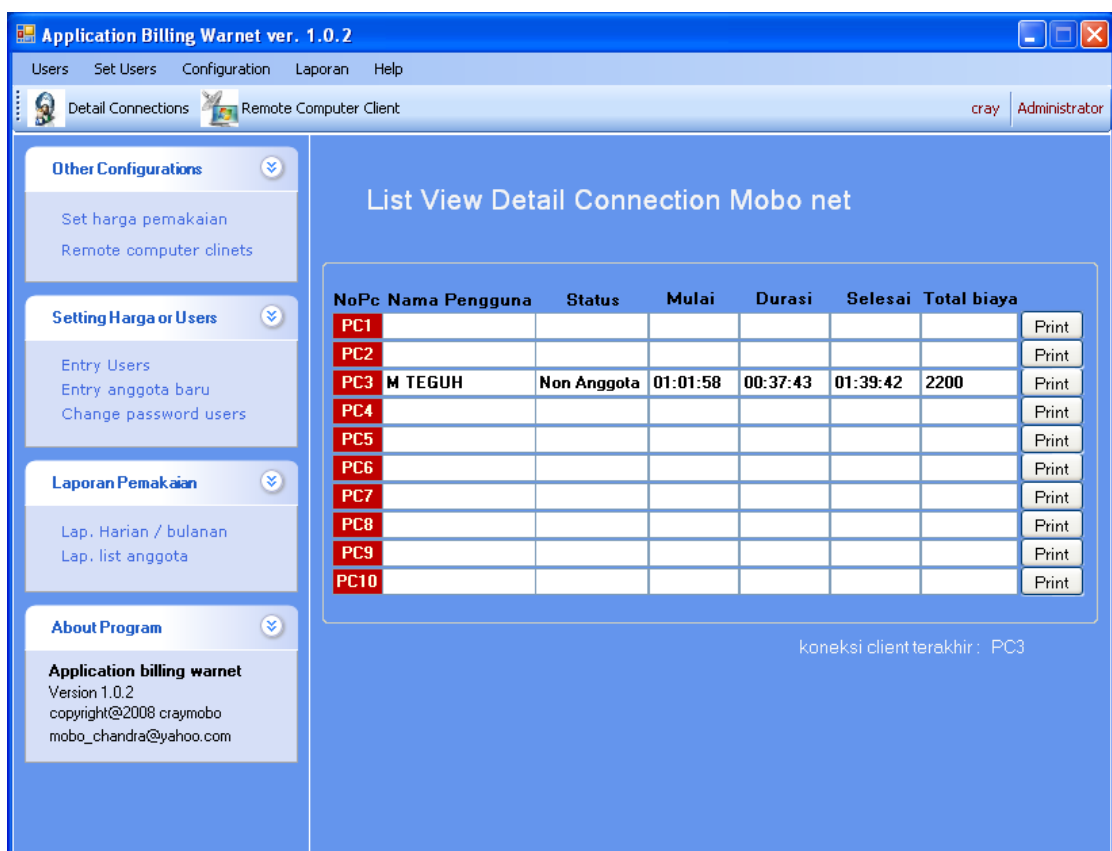


Gambar 4.18 Tampilan Login (*client*)

Pada aplikasi *server* memperbolehkan dua user yang boleh masuk ke sistem, pertama login sebagai ‘Operator’ dan login kedua sebagai ‘Administrator’.

Pada aplikasi *server* jika user memasukkan login sebagai Operator, maka menu utama akan aktif dengan tidak mengaktifkan submenu Set Harga Pengguna pada

menu konfigurasi dan submenu *entry user* pada menu *set users*, hal ini dikarenakan untuk membatasi otoritas login sebagai Operator, sebab didesain bahwa Operator dapat entri anggota, remote komputer *client* dan print laporan, Sedangkan jika user memasukan login sebagai Adminisrator dan memasukan password secara benar, Form menu utama akan aktif dengan mengaktifkan seluruh menu-menu yang ada tanpa membatasi akses. Gambar 4.19 memperlihatkan tampilan menu utama.



Gambar 4.19 Tampilan Menu Utama pada aplikasi dengan konsep *MDI (server)*

Aplikasi menu utama menggunakan konsep aplikasi *MDI (multiple document Interface)* bila aplikasi dijalankan, maka akan terlihat sebuah jendela utama aplikasi seperti terlihat pada gambar 4.19

Pada jendela utama aplikasi terdapat 5 menu utama dan 4 buah *panel box* utama yang bisa diakses oleh pengguna. Kelima menu dan empat *panel box* utama

tersebut pada kegunaannya sama hanya saja untuk melakukan keleluasaan user dalam mengakses atau memilih menu dalam menggunakan aplikasi tersebut. Menu utama tersebut adalah Users, Setusers, configuration, Laporan dan Help. Penjelasan penjelasan dari ke 5 buah menu utama adalah sebagai berikut :

1. File

Pada menu ini terdapat submenu Switch users dan Exit. Submenu Switch users digunakan untuk mengganti pengguna aplikasi dengan memasukan data *user name* dan password.

Pada submenu exit digunakan untuk menutup atau mengakhiri program jika user mengklik submenu exit maka form menu utama akan tertutup dan aplikasi akan berakhir secara keseluruhan.

2. Set users

Ketika pengguna memilih menu Set users maka akan tampil submenu “Entry users”, “Entri anggota baru” dan “change password”. Fungsi-fungsi dari form tersebut adalah sebagai berikut ini:

a. Entry users

Pada submenu entry users hanya administrator yang diperbolehkan untuk mengaksesnya, berfungsi untuk menambahkan, menghapus atau mengubah data pengguna aplikasi *server*.

b. Entry anggota baru

Berfungsi melakukan penyimpanan data anggota baru. Program akan memberikan 5-6 digit password secara *random* kepada anggota baru dalam bentuk *print out* cetak.

c. Change password

Submenu ganti password digunakan untuk mengganti password pengguna.

3. Konfigurasi

Ketika pengguna memilih menu Konfigurasi maka akan tampil submenu “Set Harga Pengguna” dan “Remote Computer *Clients*”. Fungsi-fungsi dari form tersebut adalah sebagai berikut ini:

a. Set harga pengguna

Pada form ini hanya users sebagai administrator yang memperbolehkan mengaksesnya. Berfungsi untuk pengaturan data harga billing internet dan harga anggota per bulannya. Form set harga pengguna dapat ditunjukkan pada gambar 4.20.

Level	Harga Kone...	Lap Me...	Hrg per L...
Anggota	1000	10	300
Non Anggota	1000	10	400
Operator	0	10	0

Gambar 4.20 Tampilan from set harga pengguna

b. Remote computer clients

Berfungsi untuk meremot komputer *client* dan melakukan sesuai dengan pilihan *options* yang ada berdasarkan nomor pc. Form ini dapat diakses oleh users sebagai operator maupun administrator.

4. Laporan

Berfungsi untuk melakukan pencetakan data pemakaian pengguna. Pada form laporan terdapat dua jenis laporan yang dilakukan yaitu laporan harian dan laporan bulanan pemakaian pengguna. Pada form laporan terdapat fasilitas Zoom, Refresh dan Export. Report yang dapat di simpan dalam word, excel dan berekstensi pdf. Tampilan Laporan harian dapat dilihat pada Gambar 4.21

No	Status	NoPc	Nama	Mulai	Selesai	Durasi	TotalBiaya	Tanggal	operator
1	Non Anggota	PC1	M TEGUH	01:27:57	01:45:57	00:17:58	1400	11-Jan-2009	cray
2	Non Anggota	PC1	JASMINE	01:50:20	02:24:48	00:34:26	2200	11-Jan-2009	cray
3	Non Anggota	PC5	CHANDRA	23:04:00	23:46:22	00:42:21	2600	11-Jan-2009	cray
4	Anggota	PC5	ANDRA	02:59:11	03:19:58	00:20:45	1700	11-Jan-2009	cray
5	Anggota	PC5	SETYO	03:37:07	04:02:35	00:25:26	1700	11-Jan-2009	oper
6	Non Anggota	PC10	Melatie	04:37:21	05:22:27	00:45:05	2600	11-Jan-2009	oper

Gambar 4.21 Tampilan Laporan data pemakaian harian.

5. Help

Berfungsi untuk mengetahui asal atau versi dari aplikasi tersebut dan juga berfungsi untuk menghubungi email dari pembuat aplikasi.

4.5 Pengujian Sistem

Pengujian ini dilakukan untuk menilai kehandalan dan kesesuaian aplikasi dengan tujuan penelitian.

4.5.1 Lingkup pengujian

Pengujian yang dilakukan meliputi

1. modul-modul perangkat lunak

Aplikasi *server* meliputi System File, Set users, Configurasi, Laporan yang di tunjukan pada tabel 4.3

Tabel 4.3 Modul perangkat lunak pada *server*

Modul	Sub Modul
File	1. Swtch users 2. Exit
Set Users	1. Enry users 2. Set anggota baru
Configurasi	1. Set harga pengguna 2. <i>Remote computer client</i>
Laporan	3. Laporan harian 4. Laporan bulanan

Aplikasi pada *client* meliputi Login dan Harga pemakaian yang dapat ditunjukan pada tabel 4.4

Tabel 4.4 Modul perangkat lunak pada *client*

Modul	Sub Modul
Login	1. Anggota 2. Non anggota
Harga Pemakaian	1. Anggota 2. Non anggota

Metode Pengujian ini meliputi :

1. Kebenaran sematik dengan parameter pengujian terdapat kesesuaian tujuan definistif modul dengan hasil eksekusi. (*Black Box Testing*)
2. Reliabilitas Perangkat lunak

Pengujian reliabilitas perangkat lunak meliputi hal-hal berikut yang ditunjukan pada tabel 4.5.

Tabel 4.5 Pengujian reliabilitas perangkat lunak

Bahasan	Keterangan
Koneksi <i>client</i> dengan <i>server</i>	Program akan di uji apakah program pada <i>client</i> dapat terkoneksi dengan program pada <i>server</i> .
Koneksi database	Program akan di uji apakah program dapat terkoneksi dengan baik atau tidak dengan database <i>mysql</i> .
Terterkaitan antara formulir yang satu dengan yang lain	Program akan di uji apakah keterkaitan antara formulir yang satu dengan yang lainnya telah terpenuhi
Autentifikasi user/login.	Program akan di uji apakah autentifikasi user pada formulir login data dapat digunakan sesuai keinginan pembuat.
Input Data	Program akan di uji apakah form Insert Data dapat digunakan untuk menyimpan data, mencari data, pengeditan data bahkan menghapus data sesuai keinginan user.
<i>Remote komputer client</i>	Program akan diuji apakah remote komputer <i>client</i> dapat digunakan untuk merestart, <i>turn off</i> dan <i>log off</i> komputer <i>clients</i>
Pencetakan laporan	Program akan di uji apakah form laporan akan menampilkan data sesuai yang diinginkan dan apakah dapat melakukan print laporan.
Penghitungan biaya	Program akan di uji apakah penghitungan biaya berjalan dengan baik sesuai dengan banyaknya waktu yang digunakan pada komputer <i>client</i> dan dengan besarnya harga yang di set pada komputer <i>server</i> .

4.5.2 Data Hasil Pengujian

Berikut ini Akan dijelaskan Tentang data hasil pengujian modul-modul dan pengujian realibilitas perangkat lunak yang digambarkan dalam bentuk tabel. Tabel 4.6 memperlihatkan pengujian modul-modul.

Tabel 4.6 Pengujian modul-modul.

Modul	Sub Modul	Kebenaran Sintak	Kebenaran Semantik	Keterangan
File	Swicth user	No error	Sesuai Tujuan	Ketka sub Menu Switch user dipanggil maka program akan menampilkan form switch login.
	Exit	No error	Sesuai Tujuan	Ketika sub Menu Exit. dipanggil maka dipanggil maka program akan menampilkan form konfirmasi login.
Set users	Entry users	No error	Sesuai Tujuan	Ketika sub Menu Entry users dipanggil maka program akan menampilkan form Entry users
	Entry anggota baru	No error	Sesuai Tujuan	Ketika sub menu Entry anggota baru dipanggil maka program akan menampilkan form Entry anggota baru
	Ganti password	No error	Sesuai Tujuan	Ketika sub menu Ganti password dipanggil maka program akan menampilkan form ganti password
configurasi	Set harga pengguna	No error	Sesuai Tujuan	Ketika sub menu Set harga pengguna dipanggil maka program akan menampilkan form set harga pengguna
	Remote computer clients	No error	Sesuai Tujuan	Ketika sub Menu <i>Remote computer clients</i> dipanggil maka program akan menampilkan <i>Remote computer clients</i>
Laporan	Laporan harian	No error	Sesuai Tujuan	Ketika sub menu Laporan harian dipanggil maka program akan menampilkan form

				Laporan harian
	Laporan bulanan	No error	Sesuai Tujuan	Ketika sub menu Laporan bulanan dipanggil maka program akan menampilkan form Laporan bulanan
Help	About	No error	Sesuai Tujuan	Ketika sub menu About dipanggil maka program akan menampilkan menu about aplikasi

Dibawah ini akan dijelaskan reliabilitas perangkat lunak yang digambarkan dalam bentuk tabel yang dapat dilihat pada tabel 4.7

Tabel 4.7 Reliabilitas perangkat lunak

Bahasan	Parameter Keberhasilan	Keterangan
1. Koneksi ke komputer <i>server</i>	Koneksi terpenuhi	Program atau komputer <i>client</i> akan terhubung ke komputer <i>server</i>
2. Koneksi Ke database	Koneksi Terpenuhi	Jika Program memanggil database Mysql maka program akan langsung terkoneksi dengan database
3. Keterkaitan antara form yang satu dengan form yang lain	Keterkaitan terpenuhi	Jika program memanggil salah satu form maka form yang dipanggil tersebut akan ditampilkan oleh program
4. Autentifikasi User/login	Autentifikasi terpenuhi	Program akan melakukan pemilihan yang jelas antara per user yang masuk kedalam system.
5. Insert data dan Update data	Data berhasil disimpan, diedit dan dihapus	Program akan menyimpan data, melakukan pencarian data, mengubah data ataupun menghapus data secara otomatis jika user mengklik salah satu tombol eksekusi pada form input data.
6. remote kompuer <i>client</i>	Remote computer terpenuhi	Program akan melakukan remote komputer <i>client</i> sesuai pilihan yang ada 'restart', 'turn off' dan 'log off' berdasarkan no pc.
7. Pencetakan laporan	Data yang di inginkan berhasil dicetak	Program akan melakukan pencetakan terhadap data yang pengguna inginkan.
8. Penghitungan biaya	Penghitungan biaya dapat diproses dengan baik	Program <i>client</i> akan melakukan penghitungan biaya pada saat di klik tombol start.

4.6 Analisis Hasil Pengujian

Dari hasil pengujian di atas dapat didapatkan beberapa kesimpulan.

1. Koneksi komputer *client* dengan komputer *server* terpenuhi, program pada *client* melakukan pengiriman data yang dibutuhkan ke server berupa tampilan layar pada *server*.
2. Koneksi program dengan database telah terpenuhi, dimana jika program memanggil database program langsung terkoneksi dengan database.
3. Keterkaitan antara form yang satu dengan form yang lain telah terpenuhi, jika user memanggil salah satu form yang terdapat pada program sebagai contoh form Insert data maka yang akan tampil dilayar adalah form Insert data, ini membuktikan keterkaitan antara form yang satu dengan form yang lain telah terpenuhi.
4. Autentifikasi user/login terpenuhi, program akan menampilkan kotak isian login sebagai autentifikasi user. Contohnya jika user login sebagai Operator maka form Menu Utama tidak akan mengaktifkan submenu set harga pengguna, sedangkan jika user login sebagai Administrator user dapat melakukan tanpa batasan apapun.
5. Form Insert data dan Update data dapat digunakan sesuai tujuan program, dimana form Insert data dan Update data berfungsi untuk melakukan penyimpanan data ke dalam database. Selain itu, dapat juga digunakan untuk melakukan pencarian data, mengubah data maupaun proses penghapusan.
6. Remote komputer *clients* berjalan dengan baik dimana ketika pengguna melakukan remote komputer *clients* berdasarkan no pc maka program akan melakukan *remote client*.

7. Pencetakan Laporan harian dan bulanan pemakaian warung internet berhasil dicetak, dimana program akan menampilkan dan melakukan pencetakan laporan terhadap seluruh record yang diinginkan oleh user berdasarkan tanggal dan bulan.
8. Penghitungan biaya berjalan dengan baik sesuai dengan pengaturan harga pada *server*, jika harga di set :
 - Lap per-menit = 10 menit
 - Harga perkoneksi = 1000 (Rp)
 - Harga per lap menit = 350 (Rp)

Hasil dari penghitungan biaya dapat ditunjukkan pada tabel 4.3

Tabel 4.3 Pengujian penghitungan biaya

No	Lama pemakaian dengan per-10 menit(scenario)	Total biaya di hitung secara manual	Total biaya dihitug pada program.
1	Lebih kecil dari (>)10	1000	1000
2	10	1350	1350
3	20	1700	1700
4	30	2050	2050
5	40	2400	2400
6	50	2700	2700

BAB V

PENUTUP

Berdasarkan analisis, perancangan, dan implementasi serta pengujian yang dilakukan terhadap aplikasi ini, dapat diperoleh beberapa hal seperti pada sub bab berikut ini.

5.1 Kesimpulan

Dalam pembuatan Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 dapat diambil beberapa kesimpulan, antara lain :

1. Aplikasi dapat dipergunakan untuk mengelola data-data anggota, operator dan administrator, menghitung dan mencatat jumlah biaya pemakaian *internet* oleh *user* serta membuat laporan untuk data-data yang terdapat pada *database server*.
2. Dalam membangun aplikasi ini penulis menggunakan program basis data Mysql 5.0 yang membutuhkan 7 tabel dan 8 *stored procedure*, pada program Visual Studio 2005 terdapat 5 *forms* pada aplikasi *client* dan 16 *forms* pada aplikasi *server*, dan untuk laporan menggunakan Crystal Reports 10.
3. Administrator dapat mengatur harga pemakaian internet untuk *user* yang sudah terdaftar menjadi anggota ataupun untuk *user* yang tidak terdaftar menjadi anggota, dengan mengkonfigurasi biaya pemakaian yang tersimpan pada *database* yang terdapat di komputer *server*.

4. Administrator atau operator dapat melakukan *shutdown*, *restart*, *logoff* pada komputer client secara mudah dan cepat, langsung dari komputer *server*.
5. *User* dapat mengkonfigurasi sendiri jumlah waktu yang diperlukan dalam menggunakan *internet*, yang bisa disesuaikan dengan kemampuan *user* berdasarkan waktu yang dibutuhkan maupun berdasarkan jumlah dana yang dimiliki oleh *user*, dan *user* dapat memonitor biaya pemakaian tersebut secara *real time*.
6. Sebelum sistem perangkat lunak ke pengguna, penulis melakukan pengujian dengan sistem detail terhadap aplikasi, yang meliputi pengujian kebenaran sintaks dan kebenaran semantik. Pengujian tersebut terdiri pengujian modul-modul dan pengujian reliabilitas perangkat lunak yang penulis gambarkan dalam bentuk tabel-table pengujian.

5.2 Saran

Untuk pengembangan Billing Internet Berbasis *Client Server* Menggunakan Visual Basic 2005 ini, perlu diperhatikan beberapa hal sebagai berikut :

1. Billing Internet Berbasis Client Server Menggunakan Visual Studio 2005. dengan menambahkan beberapa fungsi lainnya, seperti memonitor data hasil pemakaian internet menggunakan web, melakukan modifikasi data dari komputer client atau menggunakan web.
2. Aplikasi dapat dilengkapi dengan menu help atau bantuan. Yang berfungsi dalam menampilkan informasi sistem dan tatacara pengguna sistem.
3. Untuk lebih optimalnya tampilan Billing Internet Berbasis Client Server Menggunakan Visual Basic 2005 sebaiknya dijalankan pada resolusi 1024 x 768.

4. Untuk melakukan penyimpanan data password ke dalam database Mysql dapat dilakukan enkrip data sehingga nama password tidak ada yang mengetahui selain user itu sendiri.
5. Pada Aplikasi Billing Warnet berbasis *client server* dapat ditambahkan *event* pada saat komputer client mati secara tiba-tiba.
6. Aplikasi Billing Warnet dapat ditambahkan penghitungan penggunaan dengan fasilitas internet maupun penggunaan tanpa fasilitas internet

Daftar Pustaka

- Suharli, Suryanto. Membangun Aplikasi Berbasis Windows, Jakarta : PT. Elek Media Komputindo. 2005
- Suryo. Ario. Pemograman Visual Basic 2005, Jakarta : PT. Elek Media Komputindo. 2005
- Alexander, Hengky. Membuat Aplikasi Sistem Inventory dengan Visual Basic.net 2005 dan Sql Server 2005. Jakarta : PT. Elex Media Komputindo. 2005
- Harianto, Kristanto. Konsep Database, Jakarta : PT. Elek Media Komputindo. 2004
- Haris. N. Mengenal Jaringan Komputer. <http://simkesugm07.wordpress.com>
[2008, Mei 25]
- Sigit. Port Number dan Cara Melacaknya, <http://neotekno.com> [2008. Mei 25]
- Butt, Faraz. Socket Programming using Visual Basic.net 2003. [online], available:
<http://www.devasp.net/net/articles/display/192.html>. [2008, Mei 25]
- Fiade, Andrew. Indy sebagai Socket Programming pada Delphi 2006 [online],
available: <http://mercubuana-it.org> [2008, Mei 25]
- [online], available: <http://codeproject.com> [2008,april 28]
- [online], available: <http://vb-bego.com> [2008,april 28]
- [online], available: <http://vbdotnetheavent.com> [2008,april 20]

LAMPIRAN I

QUERY & STORED PROSEDURE MYSQL

SQLyog Enterprise - MySQL GUI v7.02
MySQL - 5.1.30-community : Database - dwarnet

```
CREATE DATABASE dwarnet;
USE dwarnet;
Struktur tabel untuk tabel anggota :
CREATE TABLE `t_anggota` (
  `ID` int(6) NOT NULL
  AUTO_INCREMENT,
  `level` varchar(15) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `password` varchar(20) NOT NULL,
  `tgldaftar` date NOT NULL,
  `tglakhir` date NOT NULL,
  `status` varchar(10) NOT NULL,
  `bulan` int(2) NOT NULL,
  `harga` int(10) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `level` (`level`),
  CONSTRAINT `t_anggota_ibfk_1`
  FOREIGN KEY (`level`) REFERENCES
  `t_hrgblnanggota` (`level`)
) ENGINE=InnoDB AUTO_INCREMENT=7
DEFAULT CHARSET=latin1;
Struktur tabel untuk tabel biaya
pemakaian :
CREATE TABLE `t_biaya_pemakaian` (
  `ID` int(6) NOT NULL
  AUTO_INCREMENT,
  `level` varchar(15) NOT NULL,
  `nopc` varchar(8) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `mulai` time NOT NULL,
  `selesai` time NOT NULL,
  `durasi` time NOT NULL,
  `totalbiaya` int(6) NOT NULL,
  `tanggal` date NOT NULL,
  `operator` varchar(30) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `level` (`level`),
  CONSTRAINT
  `t_biaya_pemakaian_ibfk_1` FOREIGN
  KEY (`level`) REFERENCES `t_harga`
  (`level`)
) ENGINE=InnoDB;
Struktur tabel untuk tabel harga :
DROP TABLE IF EXISTS `t_harga`;
CREATE TABLE `t_harga` (
  `level` varchar(15) NOT NULL,
  `hrgerkoneksi` int(11) NOT NULL,
  `lapmenit` int(11) NOT NULL,
  `hrgerlapmenit` int(11) NOT NULL,
  PRIMARY KEY (`level`)
) ENGINE=InnoDB;
```

Struktur tabel untuk tabel harga bulan anggota :

```
CREATE TABLE `t_hrgblnanggota` (
  `level` varchar(15) NOT NULL,
  `harga` int(5) NOT NULL,
  PRIMARY KEY (`level`),
  CONSTRAINT
  `t_hrgblnanggota_ibfk_1` FOREIGN KEY
  (`level`) REFERENCES `t_harga`
  (`level`)
) ENGINE=InnoDB;
```

Struktur tabel untuk tabel IP :

```
DROP TABLE IF EXISTS `t_ip`;
CREATE TABLE `t_ip` (
  `PC` varchar(8) NOT NULL,
  `IP` varchar(16) DEFAULT NULL,
  `STATUS` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`PC`)
) ENGINE=InnoDB;
```

Struktur Tabel untuk tabel users :

```
CREATE TABLE `t_user` (
  `ID` int(11) NOT NULL
  AUTO_INCREMENT,
  `level` varchar(15) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `password` varchar(20) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `level` (`level`),
  CONSTRAINT `t_user_ibfk_1` FOREIGN
  KEY (`level`) REFERENCES `t_harga`
  (`level`)
) ENGINE=InnoDB;
```

Struktur tabel untuk tabel temp :

```
DROP TABLE IF EXISTS `temp`;
CREATE TABLE `temp` (
  `temp` int(6) DEFAULT NULL
) ENGINE=InnoDB;
```

Stored prosedur sp_akhir :

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `sp_akhir`(ppc
  varchar(6),poperator
  varchar(30),pselesai time )
BEGIN
  delete temp from temp ;
  insert into temp select ID
  from t_biaya_pemakaian where nopc =
  ppc order by id desc limit 1;
  update t_biaya_pemakaian,temp
  set operator = poperator, selesai =
  pselesai where id = temp.temp ;
```

```

        END */$$
DELIMITER ;
Stored prosedur harga :
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `sp_harga`(in plevel
varchar(12))
BEGIN
        select
t_harga.hrgperkoneksi,t_harga.hrgper
lapmenit,t_harga.lapmenit from
t_harga
        where t_harga.level=plevel;
        END */$$
DELIMITER ;
Stroed Prosedur harga anggota :
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `sp_hrganggota`(in pharga
int)
BEGIN
        update t_hrgblnanggota set
        harga = pharga
        where level='Anggota';
        END */$$
DELIMITER ;
Stored Prosedur sp_insertanggota :
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `sp_insertanggota`(in
pnama varchar(30),in ppass
varchar(20), in pbln int)
BEGIN
insert into
t_anggota(nama,t_anggota.password,bu
lan,tgldaftar,tglakhir,t_anggota.sta
tus,t_anggota.harga,t_anggota.level)
values(pnama,ppass,pbln,CURDATE(),DA
TE_ADD(tgldaftar, Interval pbln
month),if(tglakhir>=CURDATE(),'Aktif
','Non Aktif'),(SELECT harga from
t_hrgblnanggota),'Anggota');
        END */$$
DELIMITER ;
Stored Prosedure transaksi anggota :
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `sp_transanggota`()

```

```

BEGIN
        END */$$
DELIMITER ;
Stored Prosedur update harga :
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `sp_uharga`(in plevel
varchar(12),in pkonek int(11),in
plap int(11),in phrglap int(11))
BEGIN
        update t_harga set
        hrgperkoneksi=pkonek,
        lapmenit=plap,
        hrgperlapmenit
=phrglap
        where t_harga.level=plevel;
        END */$$
DELIMITER ;
Stored Prosedur Update data per lap
menit :
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `sp_updatelap`(ppc
varchar(6),ptotal int(6))
BEGIN
        delete temp from temp ;
        insert into temp select ID
from t_biaya_pemakaian where nopc =
ppc order by id desc limit 1;
        update t_biaya_pemakaian,temp
set totalbiaya = ptotal where id =
temp.temp ;
        END */$$
DELIMITER ;
Stored Prosedure View Harga :
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `vw_harga`()
BEGIN
        select
t_harga.hrgperkoneksi,t_harga.hrgper
lapmenit,t_harga.level,t_harga.lapme
nit
        from t_harga ;
        END */$$
DELIMITER ;

```

LAMPIRAN II LISTING PROGRAM CLIENT

```

                                End If
                                End Sub
                                Private Sub Login()
                                    level = ""
                                    If CAnggota.Checked = True
                                        Then
                                            Cmd.Connection =
                                                Objcon.Cn
                                            Sql = "select
COUNT(t_anggota.ID),t_anggota.nama,t_
anggota.password,t_anggota.status,t_a
nggota.tglakhir from t_anggota " & _
" where t_anggota.nama
=" & TxtUname.Text & "' and
t_anggota.password =" & txtpass.Text
& "' group by t_anggota.status "
                                                Objcon.CNOpen()
                                                Cmd = New
MySQLCommand(Sql, Objcon.Cn)
                                                Try
                                                    Objcon.CNOpen()
                                                    DR =
Cmd.ExecuteReader
                                                    While DR.Read
                                                        myid =
DR(0).ToString
                                                        nama =
DR("nama").ToString
                                                        pwd =
DR("password").ToString
                                                        level =
DR("status").ToString
                                                        tanggal =
DR("tglakhir").ToString
                                                        Cmd.Dispose()
                                                    End While
                                                    If myid = 0 Then
                                                        Call showmsgok()

message.msgharga.Text = ""

message.lblmsg.Text = "Ada Kesalahan
pada User Name / password !! "
                                                    Else
                                                        If level =
"Aktif" Then
                                                            Call
getform2()
                                                        Else
                                                            Call
showmsgok()

message.msgharga.Text = "Maaf, Access
Ditolak !!"

message.lblmsg.Text = "Masa Aktif
Keanggotaan Anda berakhir tanggal: "
& tanggal

```

Class koneksi database pada client :

```

Imports MySql.Data.MySqlClient
Public Class connect
    Public Cn As New MySqlConnection
    Public StrCon As String =
"Host=localhost;uid=root;pwd=jasmine;
" & _
"Database=dwarnet;Port=3306;" & _
"Protocol=TCP;Compress=false;" & _
"Min Pool Size=0;Max Pool
Size=100;" & _
"Compress=false;" & _
"Connection Lifetime=0"
    Public Sub CNOpen()
        Cn = New
MySQLConnection(StrCon)
        Try
            If Cn.State =
ConnectionState.Closed Then
                Cn.Open()
            End If
            Catch ex As Exception
                MsgBox("Connection Err: "
& ex.Message)
            End Try
        End Sub
        Public Sub Cnclose()
            Cn.Close()
        End Sub
    End Class

Kode pada form login :
Imports MySql.Data.MySqlClient
Imports System.Net.Dns
Public Class FrmDepan
    Private Objcon As New connect
    Private Cmd As New MySqlCommand
    Private DR As MySqlDataReader
    Dim Sql As String
    Dim param1 As MySqlParameter
    Dim param2 As MySqlParameter
    Dim myid, nama, level, pwd,
tanggal, MyPC, Savenama As String
    'Public SaveID As String
    Public DPC, IPCLIENT As String
    Private Sub
CAnggota_CheckedChanged(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
CAnggota.CheckedChanged
        If CAnggota.Checked = True
            Then
                txtpass.Enabled = True
            Else
                txtpass.Enabled = False

```

```

                End If
            End If
        Catch ex As Exception
            MsgBox("LGON ERR : " &
ex.Message)
        Finally
            DR.Close()
            myid = 0
        End Try
    Else
        Call getform2()
    End If
End Sub
Private Sub SimpanData()
    Objcon.CNOpen()
    Dim Status, waktu, tanggal As
String
    waktu =
DateTime.Now.ToString("HH:mm:ss")
    tanggal =
DateTime.Now.ToString("yyyy-MM-dd")
    If Me.CAnggota.Checked = True
Then
        Status = "Anggota"
    Else
        status = "Non Anggota"
    End If
    Sql = "insert into
t_biaya_pemakaian
(level,nopc,nama,mulai,selesai,durasi
,totalbiaya,tanggal,operator)" & _
" values('" & Status & "','" &
Lblhostname.Text & "','" & Savenama &
',' & waktu & "','" & '00:00:00
','00:00:00',(select hrperkoneksi
from t_harga where level = '" &
Status & "','" & tanggal & "','" &
Cmd = New MySqlCommand(Sql,
Objcon.Cn)
    Try
        Cmd.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox("Simpan Data Error
: " & ex.Message)

    End Try
    Objcon.CnClose()
End Sub
Private Sub btnstart_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btnstart.Click
    If TxtUname.Text = "" Then
        MsgBox("User Name tidak
boleh kosong !!",
MsgBoxStyle.Exclamation, "Masukan
Nama")
    Else
        Savenama = TxtUname.Text
        Objcon.CNOpen()

                Call SimpanData()
                Call Login()
                Call clear()
            End If
        End Sub
    Private Sub FrmDepan_Load(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        Objcon.CNOpen()
        autoIP()
        Call Panggil_IP()
    End Sub
    Private Sub showmsgok()
        message.Show()
        message.Focus()
        message.BtnOK.Visible = True
        message.BtnYes.Visible =
False
        message.Btnno.Visible = False
    End Sub
    Private Sub clear()
        Me.txtpass.Clear()
        Me.TxtUname.Clear()
    End Sub
    Private Sub getform2()
        Me.Hide()
        flogin.Hide()
        message.Hide()
        Form2.Show()
    End Sub
    Private Sub btnchat_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btnchat.Click
        flogin.Show()
        flogin.Focus()
        TxtUname.Text = ""
        txtpass.Text = ""
    End Sub
    Private Sub autoIP()
        Dim shostname As String
        ' Dim Sql As String
        shostname =
System.Net.Dns.GetHostName
        Console.WriteLine("Your
Machine Name = " & shostname)
        'Call Get IPAddress
        Console.WriteLine("Your IP =
" & GetIPAddress())
        Lblhostname.Text =
Environment.MachineName.ToString
        TextBox2.Text = "" &
GetIPAddress()
    End Sub
    Shared Function GetIPAddress() As
String
        Dim oAddr As
System.Net.IPAddress
        Dim sAddr As String

```

```

        With
System.Net.Dns.GetHostByName(System.N
et.Dns.GetHostName())
        oAddr = New
System.Net.IPAddress(.AddressList(0).
Address)
        sAddr = oAddr.ToString
        End With
        GetIPAddress = sAddr
    End Function
    Private Sub Panggil_IP()
        Dim IP, Status As String
        Dim sql2 As String
        DPC = ""
        IPCLIENT = ""
        With Me.Lblhostname
            If (.Text = "PC1") Or
(.Text = "PC2") Or (.Text = "PC3") Or
(.Text = "PC4") Or (.Text = "PC5") Or
(.Text = "PC6") Or (.Text = "PC7") Or
(.Text = "PC8") Or (.Text = "PC9") Or
(.Text = "PC10") Then
                Sql = " Update t_ip
set ip='" & TextBox2.Text & "',
STATUS='STOP' where PC='" &
Lblhostname.Text & "' and
STATUS='Stop' "
                sql2 = "select
COUNT(t_ip.pc),t_ip.ip,t_ip.status
from t_ip" & _
                " where t_ip.pc
='" & Lblhostname.Text & "' and
t_ip.status='Stop' group by pc "
                Objcon.CNOpen()
                Try
                    Cmd = New
MySQLCommand(sql2, Objcon.Cn)
                    DR =
Cmd.ExecuteReader
                    While DR.Read
                        MyPC =
DR(0).ToString
                        IP =
DR("ip").ToString
                        Status =
DR("status").ToString
                        Cmd.Dispose()
                    End While
                    DR.Close()
                    If MyPC = 0 Then
                        MsgBox("Nomor
PC sudah terpakai atau dalam keadaan
running !!, klik config untuk
mengganti. ")
                    End Sub
                Else
Objcon.CNOpen()
                    Cmd = New
MySQLCommand(Sql, Objcon.Cn)

```

```

        DPC =
Lblhostname.Text
        IPCLIENT =
TextBox2.Text
        Try
Cmd.ExecuteNonQuery()
        Catch ex As
Exception
MsgBox("Error : pada database ->
tabel t_ip : " & ex.Message)
        End Try
Objcon.Cnclose()
        End If
        Catch ex As Exception
            MsgBox("Error
pada database -> tabel t_ip : " &
ex.Message)
        End Try
        Else
            MsgBox(" ERR : mohon
untuk mengganti computer name : Klik
Configurasi Client !!")
            'Me.Close()
            btnstart.Enabled =
False
        End If
    End With
    End Sub
    Private Sub Timer1_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        If ProgressBar1.Value < 100
Then
            ProgressBar1.Value += 5
        Else
            Timer1.Enabled = False
            ProgressBar1.Value = 0
            ProgressBar1.Visible =
False
            Call getform2()
        End If
    End Sub
End Class
Kode pada form login :
Imports MySql.Data.MySqlClient
Imports System.Globalization
Imports System
Imports System.Net
Imports System.Net.Sockets
Imports System.Text
Imports System.Threading
Imports System.IO
Imports
System.Runtime.InteropServices
Imports System.Text.Encoding
Public Class Form2
    Private objcon As New connect

```



```

Private Cmd As New MySqlCommand
Private DR As MySqlDataReader
Dim PKoneksi, PMenit, pLapMenit
As Integer
Dim Total As Integer = 0
Dim sw As New Stopwatch
Dim paused As Boolean = False
Dim Jenis As String
Public MSG As Byte()
Public MSGLap As Byte()
Public th As Thread
Public cap As Thread
Public Data As String
Dim Sql As String
Dim i As Integer
Delegate Sub SetDisplay(ByVal
[Text] As String)
#Region "connect server"
Public Sub Connect()
Try
    ipHostinfo =
Dns.Resolve(FrmDepan.IPCLIENT)
    ipAdd =
ipHostinfo.AddressList(0)
    remoteEP = New
IPEndPoint(ipAdd, 11000)
    sender.Connect(remoteEP)
    'th = New
System.Threading.Thread(AddressOf
Receive1)
    'th.Start()
    ' i = 1
    'MessageBox.Show("Client
Connected With Server-> ")
Catch ex As Exception

MessageBox.Show("Connection ERR : " &
ex.Message)
Exit Sub
End Try
End Sub
Private Sub SendDurasi()
Try
    MSG =
Encoding.ASCII.GetBytes(FrmDepan.DPC)
    Me.sender.Send(MSG)
    'LblDurasi.Text = ""
Catch ex As Exception

'MessageBox.Show(ex.Message)
'Me.Close()
'FrmDepan.Show()
End Try
End Sub
Private Sub SendLAP()
Try
    MSGLap =
Encoding.ASCII.GetBytes(FrmDepan.DPC
& "L")

```

```

'status =
Encoding.ASCII.GetBytes("Aktif")
Me.sender.Send(MSGLap)
'lbluser.Text = ""
'LblDurasi.Text = ""
Catch ex As Exception
    ' MessageBox.Show("Send
Data Error Periksa Connection Server
dan Clinet")
    Me.Close()
End Try
End Sub
#End Region
Private Sub Form2_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.sender = New
System.Net.Sockets.Socket(Net.Sockets
.AddressFamily.InterNetwork,
Net.Sockets.SocketType.Stream,
Net.Sockets.ProtocolType.Tcp)
    Connect()
    SendDurasi()
    objcon.CNOpen()
    UserInfo()
    Getharga()
    permenit()
    InitializeTimer()
    If sw.IsRunning Then
        sw.Stop()
    Else
        sw.Start()
    End If
    Total = PKoneksi
    LblHarga.Text = PKoneksi
End Sub
Private Sub
Form2_FormClosed(ByVal sender As
Object, ByVal e As
System.Windows.Forms.FormClosedEventA
rgs) Handles Me.FormClosed
    SendDurasi()
    ' If i = 1 Then
    ' th.Abort()
Try
    Me.sender.Shutdown(SocketShutdown.Bot
h)
    Me.sender.Close()
    ' End If
    Dim p1 As Process
    Dim p As Process() =
Process.GetProcessesByName("TesWarnet
.exe")
    For Each p1 In p
        p1.Kill()
    Next
Catch ex As Exception

End Try
End Try

```

```

End Sub
Private Sub Timer3_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick
lap()
SendLAP()
End Sub
Private Sub UpdateLap()
Dim param1, param2 As
MySQLParameter
Cmd.Connection = objcon.Cn
Cmd.CommandType =
CommandType.StoredProcedure
Cmd.CommandText =
"sp_updatelap"
param1 = New
MySQLParameter("ppc",
FrmDepan.Lblhostname.Text)
param2 = New
MySQLParameter("ptotal",
LblHarga.Text)
Cmd.Parameters.Add(param1)
Cmd.Parameters.Add(param2)
Try
objcon.CNOpen()
Cmd.ExecuteNonQuery()
Catch ex As Exception
MsgBox("ERR : " &
ex.Message)
Finally
objcon.CnClose()
End Try
End Sub
Private Sub Button1_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs)
Me.Close()
FrmDepan.Close()
End Sub
Private Sub Button3_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs)
Me.Visible = False
End Sub
Private Sub
NotifyIcon1_MouseDoubleClick(ByVal
sender As System.Object, ByVal e As
System.Windows.Forms.MouseEventArgs)
Handles NotifyIcon1.MouseDoubleClick
Me.Visible = True
End Sub
Private Sub BtnStop_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
BtnStop.Click
Call showmsgyes()
message.msgharga.Text =
"Total Biaya : Rp." &
Me.LblHarga.Text & ".-"

```

```

message.lblmsg.Text = "Click
Yes untuk selesai, atau No untuk
melanjutkan. "
message.Label1.Text = "Terima
Kasih Telah Mengunjungi Mobo Net"
End Sub
Private Sub
OpenToolStripMenuItem_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
OpenToolStripMenuItem.Click
Me.Visible = True
End Sub
Private Sub
StopToolStripMenuItem_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
StopToolStripMenuItem.Click
Call showmsgyes()
message.msgharga.Text =
"Total Biaya : Rp." &
Me.LblHarga.Text & ".-"
message.lblmsg.Text = "Click
Yes untuk selesai, atau No untuk
melanjutkan. "
message.Label1.Text = "Terima
Kasih Telah Mengunjungi Mobo Net"
End Sub

Private Sub InitializeTimer()
' Run this procedure in an
appropriate event.
' Set to 1 second.
Timer1.Interval = 1000
' Enable timer.
Timer1.Enabled = True
End Sub
Private Sub Timer1_Tick(ByVal
Sender As Object, ByVal e As
EventArgs) Handles Timer1.Tick
' Set the caption to the
current time.
lbldate.Text = DateTime.Now
End Sub
Private Sub Button1_Click_1(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Button1.Click
Me.Visible = False
End Sub
Private Sub Timer2_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
If sw.IsRunning Then
Dim ts As TimeSpan =
TimeSpan.FromSeconds(sw.Elapsed.Total
Seconds)
LblDurasi.Text =
String.Format(CultureInfo.CurrentCult

```

```

ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
    If paused Then
        lbllap.Text =
LblDurasi.Text
        paused = False
    End If
End If
End Sub
Private Sub lap()
    If sw.IsRunning Then
        paused = True
    End If
    Total = Total + PMenit
    LblHarga.Text = Total
    message.msgharga.Text =
"Total Biaya : Rp." &
Me.LblHarga.Text & ".-"
    objcon.CNOpen()
    UpdateLap()
End Sub
Private Sub UserInfo()
    lbluser.Text =
FrmDepan.TxtUname.Text
    If
FrmDepan.RBNLimited.Checked Then
        LblLama.Text = "No
Limited."
    ElseIf
FrmDepan.RB1jam.Checked Then
        LblLama.Text = "1 Jam."
    ElseIf
FrmDepan.RB2jam.Checked Then
        LblLama.Text = "2 Jam."
    Else
        LblLama.Text = "30
Minutes."
    End If
    If FrmDepan.CAnggota.Checked
= True Then
        lblanggota.Text =
"Anggota"
    Else
        lblanggota.Text = "Non
Anggota"
    End If
End Sub
Private Sub permenit()
    Timer3.Interval = pLapMenit *
60000
End Sub

Private Sub Getharga()
    Dim param As MySqlParameter
    Cmd.Connection = objcon.Cn
    Cmd.CommandType =
CommandType.StoredProcedure
    Cmd.CommandText = "sp_harga"

```

```

        param = New
MySqlParameter("plevel",
lblanggota.Text)
        Cmd.Parameters.Add(param)
        If lblanggota.Text =
"Anggota" Then
            Try
                objcon.CNOpen()
                DR =
Cmd.ExecuteReader
                While DR.Read
                    PKoneksi =
DR("hrgperkoneksi").ToString
                    PMenit =
DR("hrgperlapmenit").ToString
                    pLapMenit =
DR("lapmenit").ToString
                    Cmd.Dispose()
                End While
                Catch ex As Exception
                    MsgBox("ERR : " &
ex.Message)
            Finally
                DR.Close()
                objcon.Cnclose()
            End Try
        Else
            Try
                objcon.CNOpen()
                DR =
Cmd.ExecuteReader
                While DR.Read
                    PKoneksi =
DR("hrgperkoneksi").ToString
                    PMenit =
DR("hrgperlapmenit").ToString
                    pLapMenit =
DR("lapmenit").ToString
                    Cmd.Dispose()
                End While
                Catch ex As Exception
                    MsgBox("ERR : " &
ex.Message)
            Finally
                DR.Close()
                objcon.Cnclose()
            End Try
        End If
    End Sub
Private Sub showmsgyes()
    message.Show()
    message.Focus()
    message.msgharga.Text = ""
    message.BtnOK.Visible = False
    message.BtnYes.Visible = True
    message.Btnno.Visible = True
End Sub
End Class

```

Kode Pada Form Konfigurasi :

```

Public Class Config
    Declare Function SetComputerName
Lib "kernel32" _
    Alias "SetComputerNameA" (ByVal
lpComputerName As String) As Long
    Public Function
ChangeComputerName(ByVal
sNewComputerName As String) As
Boolean
        On Error Resume Next
        Dim nReturn As Long
        nReturn =
SetComputerName(sNewComputerName)
        If Err.Number = 0 Then
            ChangeComputerName =
nReturn <> 0
        End If

```

```

    End Function
    Private Sub Button1_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Button1.Click
        SetComputerName("192.168.0.2")
ChangeComputerName(CMBPC.Text)
        MsgBox("RESTART KOMPUTER
UNTUK CONFIGURASI BARU !!")
    End Sub
    Private Sub Config_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    End Sub
End Class

```

LAMPIRAN III LISTING PROGRAM SERVER

Class koneksi database pada client :

```
Imports MySql.Data.MySqlClient
Public Class connect
    Public Cn As New MySqlConnection
    Public StrCon As String =
"Host=localhost;uid=root;pwd=jasmine;
" & _
"Database=dwarnet;Port=3306;" & _
"Protocol=TCP;Compress=false;" & _
-
"Min Pool Size=0;Max Pool
Size=100;" & _
"Compress=false;" & _
"Connection Lifetime=0"
    Public Sub CNOpen()
        Cn = New
MySqlConnection(StrCon)
        Try
            If Cn.State =
ConnectionState.Closed Then
                Cn.Open()
            End If
            Catch ex As Exception
                MsgBox("Connection Err: "
& ex.Message)
            End Try
        End Sub
    Public Sub Cnclose()
        Cn.Close()
    End Sub
End Class
```

Kode pada Form Login :

```
Imports MySql.Data.MySqlClient
Imports System.net.dns
Public Class Login2
    Private Objcon As New connect
    Private ObjAtur As New AturNum
    Dim Cmd As New MySqlCommand
    Dim DR As MySqlDataReader
    Dim myid, user, access, pwd As
String
    Private Sub Button2_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Button2.Click
        Me.Hide()
    End Sub
    Private Sub btnLogin_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btnLogin.Click
        Call getlogin()
        ' MDI.Show()
        ' Me.Hide()
    End Sub
```

```
Private Sub login_Load(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    autoIP()
End Sub
Private Sub getlogin()
    Dim Sql As String
    Sql = "select
COUNT(t_user.nama),t_user.* from
t_user where nama =' " & txtnama.Text
& "' and password =' " & txtpass.Text
& "' group by level "
    Objcon.CNOpen()
    Cmd = New MySqlCommand(Sql,
Objcon.Cn)
    Try
        'Objcon.CNOpen()
        DR = Cmd.ExecuteReader
        While DR.Read()
            myid = DR(0).ToString
            user =
DR("nama").ToString
            access =
DR("level").ToString
            pwd =
DR("password").ToString
            Cmd.Dispose()
        End While
        If myid = 0 Then
            message.lblmsg.Text =
"Ada Kesalahan Pada UserName /
Password !!"
            txtnama.Focus()
            Call showmsgok()
        Else
            If access =
"Administrator" Then
                MDI.TSL1.Text =
"Administrator"
            Else
                MDI.TSL1.Text =
"Operator"
            End If
            MDI.TSL2.Text =
Me.txtnama.Text
            MDI.Savepass = pwd
            MDI.SaveUser = user
            myid = 0
            access = 0
            user = 0
            pwd = 0
            Me.txtpass.Clear()
            Me.txtnama.Clear()
            Me.Hide()
            message.Hide()
            MDI.Show()
        End If
    End Try
End Sub
```

```

        MDI.Focus()
        ' MDI.Show()
        ' MDI.Focus()
    End If
Catch ex As Exception
    MsgBox(ex.Message)
Finally
    DR.Close()
    Objcon.Cnclose()
End Try
End Sub
Private Sub showmsgok()
    message.Show()
    message.Focus()
    message.BtnOK.Visible = True
    message.BtnYes.Visible =
False
    message.Btnno.Visible = False
End Sub
Private Sub autoIP()
    Dim shostname As String
    Dim Sql As String
    shostname =
System.Net.Dns.GetHostName
    Console.WriteLine("Your
Machine Name = " & shostname)
    'Call Get IPAddress
    Console.WriteLine("Your IP =
" & GetIPAddress())
    ' TextBox1.Text = shostname
    TextBox2.Text = "" &
GetIPAddress()
    Sql = " Update t_ip set ip="
& TextBox2.Text & "', status
='Runing' where PC='SERVER'"
    Objcon.CNOpen()
    Cmd = New MySqlCommand(Sql,
Objcon.Cn)
    Try
        Cmd.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox("Error : pada
database -> tabel t_ip")
    End Try
    Objcon.Cnclose()
End Sub
Shared Function GetIPAddress() As
String
    Dim oAddr As
System.Net.IPAddress
    Dim sAddr As String
    With
System.Net.Dns.GetHostByName(System.N
et.Dns.GetHostName())
        oAddr = New
System.Net.IPAddress(.AddressList(0).
Address)
        sAddr = oAddr.ToString
    End With
    GetIPAddress = sAddr

```

```

    End Function
End Class
Kode Pada Form MDI :
Imports MySql.Data.MySqlClient
Imports System.GlobalizatiOn
Imports System.Net
Imports System.Net.Sockets
Imports System.Text
Imports System.Threading
Imports System.Drawing
Imports System.IO
Public Class MDI
    Public Savepass As String
    Public SaveUser As String
    Dim Objcon As New connect
    Dim Cmd As New MySqlCommand
    Private Adap As New
MySqlDataAdapter
    Dim DR As MySqlDataReader
    Dim handler As Socket
    Public th As Thread
    Public th1 As Thread
    Public Data As String
    Public fname As String
    Dim i As Integer
    Dim n As Integer
    Dim paused As Boolean = False
    Public msg, nama, status As
Byte()
    Dim apppath As String
    Dim NamaPc As String
    Dim Sw As New Stopwatch
    Dim cap As Integer
    Public f As Integer
    Dim StrXmlName As String
    Dim Tdurasi As String = ""
    Dim Sql, Sql2 As String
    Delegate Sub SetDisplay(ByVal
[Text] As String)
#Region "Deklarasi-Inisialisasi"
    Dim TStatus(9) As TextBox
    Dim TNama(9) As TextBox
    Dim Ttotal(9) As TextBox
    Dim Tmulai(9) As TextBox
    Dim tselesai(9) As TextBox
    Dim ID(9) As String
    Dim i1 As Integer = 0
    Dim i2 As Integer = 0
    Dim i3 As Integer = 0
    Dim i4 As Integer = 0
    Dim i5 As Integer = 0
    Dim i6 As Integer = 0
    Dim i7 As Integer = 0
    Dim i8 As Integer = 0
    Dim i9 As Integer = 0
    Dim i10 As Integer = 0
    Private Sub Inisial()
        TStatus(0) = TStatus1
        TStatus(1) = TStatus2
        TStatus(2) = TStatus3

```

```

TStatus(3) = TStatus4
TStatus(4) = TStatus5
TStatus(5) = TStatus6
TStatus(6) = TStatus7
TStatus(7) = TStatus8
TStatus(8) = TStatus9
TStatus(9) = TStatus10
TNama(0) = TNama1
TNama(1) = TNama2
TNama(2) = TNama3
TNama(3) = TNama4
TNama(4) = TNama5
TNama(5) = TNama6
TNama(6) = TNama7
TNama(7) = TNama8
TNama(8) = TNama9
TNama(9) = TNama10
Ttotal(0) = Ttotal1
Ttotal(1) = Ttotal2
Ttotal(2) = Ttotal3
Ttotal(3) = Ttotal4
Ttotal(4) = Ttotal5
Ttotal(5) = Ttotal6
Ttotal(6) = Ttotal7
Ttotal(7) = Ttotal8
Ttotal(8) = Ttotal9
Ttotal(9) = Ttotal10
Tmulai(0) = Tmulai1
Tmulai(1) = Tmulai2
Tmulai(2) = Tmulai3
Tmulai(3) = Tmulai4
Tmulai(4) = Tmulai5
Tmulai(5) = Tmulai6
Tmulai(6) = Tmulai7
Tmulai(7) = Tmulai8
Tmulai(8) = Tmulai9
Tmulai(9) = Tmulai10
tselesai(0) = Tselesai1
tselesai(1) = Tselesai2
tselesai(2) = Tselesai3
tselesai(3) = Tselesai4
tselesai(4) = Tselesai5
tselesai(5) = Tselesai6
tselesai(6) = Tselesai7
tselesai(7) = Tselesai8
tselesai(8) = Tselesai9
tselesai(9) = Tselesai10
End Sub
#End Region
#Region "Connect Clients"
Public Sub Connect()
Try
ipHostinfo =
Dns.Resolve(Dns.GetHostName())
ipAdd =
ipHostinfo.AddressList(0)
localEndPoint = New
IPEndPoint(ipAdd, 11000)
listner.Bind(localEndPoint)

```

```

listner.Listen(10)
th = New
System.Threading.Thread(AddressOf
Acceptstart)
th.Start()
i = 1
Catch ex As Exception
MessageBox.Show(ex.Message)
End Try

End Sub
Public Sub Acceptstart()
handler = listner.Accept()
th1 = New
System.Threading.Thread(AddressOf
Receive)
th1.Start()
i = 2
'MessageBox.Show("Connected")
End Sub
Public Sub Processdata(ByVal str
As String)
If Me.LBLstr.InvokeRequired
Then
Dim d As New
SetDisplay(AddressOf Processdata)
Me.Invoke(d, New Object()
{str})
Else
Me.LBLstr.Text = str
If (str = "PC1") Then
n = 0
If i1 = 0 Then
Tmr1.Enabled =
True
Sw.Start()
NamaPc = "PC1"
PCRUN()
i1 = 1
Else
i1 = 0
tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
Tdurasi =
NamaPc = "PC1"
Objcon.CNOpen()
PCSelesai()
Sw.Stop()
Tmr1.Enabled =
False
End If
End If
If (str = "PC2") Then
n = 1
If i2 = 0 Then
Tmr2.Enabled =
True
Sw.Start()

```

```

NamaPc = "PC2"
PCRUN()
i2 = 1
Else
i2 = 0
tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
TDurasi2.Text
NamaPc = "PC2"
Objcon.CNOpen()
PCSelesai()
Sw.Stop()
Tmr2.Enabled =
False
End If
End If
If (str = "PC3") Then
n = 2
If i3 = 0 Then
Tmr3.Enabled =
True
Sw.Start()
NamaPc = "PC3"
PCRUN()
i3 = 1
Else
i3 = 0
tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
TDurasi3.Text
NamaPc = "PC3"
Objcon.CNOpen()
PCSelesai()
Sw.Stop()
Tmr3.Enabled =
False
End If
End If
If (str = "PC4") Then
n = 3
If i3 = 0 Then
Tmr4.Enabled =
True
Sw.Start()
NamaPc = "PC4"
PCRUN()
i4 = 1
Else
i4 = 0
tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
TDurasi4.Text
NamaPc = "PC4"
Objcon.CNOpen()
PCSelesai()
Sw.Stop()
False
Tmr4.Enabled =
End If
End If
If (str = "PC5") Then
n = 4
If i5 = 0 Then
Tmr5.Enabled =
True
Sw.Start()
NamaPc = "PC5"
PCRUN()
i5 = 1
Else
i5 = 0
tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
TDurasi5.Text
NamaPc = "PC5"
Objcon.CNOpen()
PCSelesai()
Sw.Stop()
Tmr5.Enabled =
False
End If
End If
If (str = "PC6") Then
n = 5
If i3 = 0 Then
Tmr6.Enabled =
True
Sw.Start()
NamaPc = "PC6"
PCRUN()
i6 = 1
Else
i6 = 0
tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
TDurasi6.Text
NamaPc = "PC6"
Objcon.CNOpen()
PCSelesai()
Sw.Stop()
Tmr6.Enabled =
False
End If
End If
If (str = "PC7") Then
n = 6
If i7 = 0 Then
Tmr7.Enabled =
True
Sw.Start()
NamaPc = "PC7"
PCRUN()
i7 = 1
Else

```



```

        i7 = 0
        tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
        Tdurasi =
TDurasi7.Text
        NamaPc = "PC7"
        Objcon.CNOpen()
        PCSelesai()
        Sw.Stop()
        Tmr7.Enabled =
False
        End If
    End If
    If (str = "PC8") Then
        n = 7
        If i8 = 0 Then
            Tmr8.Enabled =
True
                Sw.Start()
                NamaPc = "PC8"
                PCRUN()
                i8 = 1
            Else
                i8 = 0
                tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
                Tdurasi =
TDurasi8.Text
                NamaPc = "PC8"
                Objcon.CNOpen()
                PCSelesai()
                Sw.Stop()
                Tmr8.Enabled =
False
                    End If
                End If
                If (str = "PC9") Then
                    n = 8
                    If i9 = 0 Then
                        Tmr9.Enabled =
True
                            Sw.Start()
                            NamaPc = "PC9"
                            PCRUN()
                            i9 = 1
                        Else
                            i9 = 0
                            tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
                            Tdurasi =
TDurasi9.Text
                            NamaPc = "PC9"
                            Objcon.CNOpen()
                            PCSelesai()
                            Sw.Stop()
                            Tmr9.Enabled =
False
                                End If
                            End If
                            If (str = "PC10") Then
                                n = 9
                                If i10 = 0 Then
                                    Tmr10.Enabled =
True
                                        Sw.Start()
                                        NamaPc = "PC10"
                                        PCRUN()
                                        i10 = 1
                                    Else
                                        i10 = 0
                                        tselesai(n).Text
= DateTime.Now.ToString("HH:mm:ss")
                                        Tdurasi =
TDurasi10.Text
                                        NamaPc = "PC10"
                                        Objcon.CNOpen()
                                        PCSelesai()
                                        Sw.Stop()
                                        Tmr10.Enabled =
False
                                            End If
                                        End If
                                        If (str = "PC1L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC2L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC3L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC4L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC5L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC6L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC7L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC8L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC9L") Then
                                            PCLAP()
                                        End If
                                        If (str = "PC10L") Then
                                            PCLAP()
                                        End If
                                    End If
                                End Sub
                                Public Sub Receive()
                                    Try
                                        Dim bytes(100000) As Byte
                                        Dim bytesRec As Integer
                                        bytes = New Byte(100000)
                                    {}
                                End Sub
                            End Sub
                    End Sub
                End Sub
            End Sub
        End Sub
    End Sub
End Sub

```

```

A:          While True
            bytesRec =
handler.Receive(bytes)
            If bytesRec > 0 Then
                Data =
Encoding.ASCII.GetString(bytes, 0,
bytesRec)
                Exit While
            End If
        End While
        Processdata(Data)

'Processdata(bytesRec.ToString())
    GoTo A
    Catch ex As Exception
        MessageBox.Show("Server
Problem:" + ex.Message.ToString())
    End Try
End Sub
#End Region
#Region "SET CLIENT"
    Private Sub PCRUN()
        Sql = "select
ID,nopc,nama,level,totalbiaya,mulai
from t_biaya_pemakaian where nopc='
& NamaPc & "' order by id desc Limit
1 "
        Objcon.CNOpen()
        Cmd = New MySqlCommand(Sql,
Objcon.Cn)
        ' Try
        Cmd.ExecuteNonQuery()
        DR = Cmd.ExecuteReader
        DR.Read()
        TStatus(n).Text =
DR("level").ToString
        TNama(n).Text =
DR("nama").ToString
        Ttotal(n).Text =
DR("totalbiaya").ToString
        Tmulai(n).Text =
DR("mulai").ToString
        ID(n) = DR("ID").ToString
        ' Catch ex As Exception
        ' MsgBox(ex.Message)
        ' End Try
        DR.Close()
        Objcon.Cnclose()
    End Sub
    Private Sub PCLAP()
        Sql2 = "select ID,totalbiaya
from t_biaya_pemakaian where nopc='
& NamaPc & "' order by id desc Limit
1 "
        Objcon.CNOpen()
        Cmd = New MySqlCommand(Sql2,
Objcon.Cn)
        ' Try
        Cmd.ExecuteNonQuery()
        DR = Cmd.ExecuteReader
        DR.Read()
        Ttotal(n).Text =
DR("totalbiaya").ToString
        ' Catch ex As Exception
        ' MsgBox(ex.Message)
        ' End Try
        DR.Close()
        Objcon.Cnclose()
    End Sub
    Private Sub PCSelesai()
        Dim param1, param2, param3,
param4 As MySqlParameter
        Cmd.Connection = Objcon.Cn
        Cmd.CommandType =
CommandType.StoredProcedure
        Cmd.CommandText = "sp_akhir"
        param1 = New
MySqlParameter("ppc", NamaPc)
        param2 = New
MySqlParameter("pselesai",
DateTime.Now.ToString("HH:mm:ss"))
        param3 = New
MySqlParameter("poperator",
TSL2.Text)
        param4 = New
MySqlParameter("pdurasi", Tdurasi)
        Cmd.Parameters.Add(param1)
        Cmd.Parameters.Add(param2)
        Cmd.Parameters.Add(param3)
        Cmd.Parameters.Add(param4)
        Try
            Objcon.CNOpen()
            Cmd.ExecuteNonQuery()
            Catch ex As Exception
                MsgBox("ERR : " &
ex.Message)
            Finally
                Objcon.Cnclose()
            End Try
        End Sub
#End Region
#Region "Remote Clients"
#End Region
#Region "Details Clients"
#End Region
#Region "Icon Clients"
#End Region
#Region "Atur warna Fonts"
    Private Sub Lbl1_MouseEnter(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl1.MouseEnter
        Me.Lbl1.LinkColor =
System.Drawing.SystemColors.GradientA
ctiveCaption
    End Sub
    Private Sub Lbl2_MouseEnter(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl2.MouseEnter

```

```

        Me.Lbl2.LinkColor =
System.Drawing.SystemColors.GradientA
ctiveCaption
    End Sub
    Private Sub Lbl3_MouseEnter(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl3.MouseEnter
        Me.Lbl3.LinkColor =
System.Drawing.SystemColors.GradientA
ctiveCaption
    End Sub
    Private Sub Lbl11_MouseLeave(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl11.MouseLeave
        Me.Lbl11.LinkColor =
System.Drawing.Color.RoyalBlue
    End Sub
    Private Sub Lbl2_MouseLeave(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl2.MouseLeave
        Me.Lbl2.LinkColor =
System.Drawing.Color.RoyalBlue
    End Sub
    Private Sub Lbl3_MouseLeave(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl3.MouseLeave
        Me.Lbl3.LinkColor =
System.Drawing.Color.RoyalBlue
    End Sub

    Private Sub Lbl5_MouseEnter(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl5.MouseEnter
        Me.Lbl5.LinkColor =
System.Drawing.SystemColors.GradientA
ctiveCaption
    End Sub
    Private Sub Lbl5_MouseLeave(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl5.MouseLeave
        Me.Lbl5.LinkColor =
System.Drawing.Color.RoyalBlue
    End Sub
    Private Sub Lbl8_MouseEnter(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl8.MouseEnter
        Me.Lbl8.LinkColor =
System.Drawing.SystemColors.GradientA
ctiveCaption
    End Sub
    Private Sub Lbl8_MouseLeave(ByVal
sender As Object, ByVal e As

```

```

System.EventArgs) Handles
Lbl8.MouseLeave
        Me.Lbl8.LinkColor =
System.Drawing.Color.RoyalBlue
    End Sub
    Private Sub Lbl9_MouseEnter(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl9.MouseEnter
        Me.Lbl9.LinkColor =
System.Drawing.SystemColors.GradientA
ctiveCaption
    End Sub
    Private Sub Lbl9_MouseLeave(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles
Lbl9.MouseLeave
        Me.Lbl9.LinkColor =
System.Drawing.Color.RoyalBlue
    End Sub
    Private Sub
Lbl10_MouseEnter(ByVal sender As
Object, ByVal e As System.EventArgs)
Handles Lbl10.MouseEnter
        Me.Lbl10.LinkColor =
System.Drawing.SystemColors.GradientA
ctiveCaption
    End Sub
    Private Sub
Lbl10_MouseLeave(ByVal sender As
Object, ByVal e As System.EventArgs)
Handles Lbl10.MouseLeave
        Me.Lbl10.LinkColor =
System.Drawing.Color.RoyalBlue
    End Sub
#End Region
#Region "ButtonPrint"
    Private Sub Rept()
        Dim DS As New DataSet
        Dim Sql2 As String
        StrXmlName = "finish"
        Dim strXmlPath As String =
Application.StartupPath & "\" & _
StrXmlName & ".xml"
        If TNama(n).Text <> "" Then
            Sql2 = "select
id,nopc,nama,TIME_FORMAT(mulai,'%H:%i
:%s')as
Mulai,TIME_FORMAT(durasi,'%H:%i:%s')a
s
Durasi,TIME_FORMAT(selesai,'%H:%i:%s'
)as
Selesai,concat('RP.',totalbiaya)as
totalbiaya,DATE_FORMAT(tanggal,'%d-
%b-%Y')as Tanggal from
t_biaya_pemakaian where nopc='" &
NamaPc & "' and nama ='" &
TNama(n).Text & "' order by id desc
limit 1 "
Objcon.CNOpen()

```

```

        Try
            Adap = New
            MySqlDataAdapter(Sql2, Objcon.Cn)
            Adap.Fill(DS)
            'Menulis Skema Ke XML
        DS.WriteXml(strXmlPath)
            FrmRptFinish.Show()
            cleartxt()
            'DR.Close()
        Catch ex As Exception
            MsgBox("ERR :" &
ex.Message)
        End Try
        Objcon.Cnclose()
    End If
End Sub
Private Sub Rept2()
    Dim DS As New DataSet
    Dim Sql1 As String
    StrXmlName = "LAnggota"
    Dim strXmlPath As String =
Application.StartupPath & "\" & _
StrXmlName & ".xml"
    Sql1 = "select
ID,level,nama,password,DATE_FORMAT(tg
ldaftar,'%d-%b-%Y')as
TglDaftar,DATE_FORMAT(tglakhir,'%d-
%b-%Y')as TglAkhir,status,bulan,harga
from t_anggota "
    Objcon.CNOpen()
    Try
        Adap = New
        MySqlDataAdapter(Sql1, Objcon.Cn)
        Adap.Fill(DS)
        'Menulis Skema Ke XML
        DS.WriteXml(strXmlPath)
        FrmLanggota.Show()
        cleartxt()
        'DR.Close()
    Catch ex As Exception
        MsgBox("ERR :" &
ex.Message)
    End Try
    Objcon.Cnclose()
End Sub
Private Sub Print3_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print3.Click
    n = 2
    NamaPc = "PC3"
    Rept()
    TDurasi3.Clear()
End Sub
Private Sub Print1_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print1.Click
    n = 0
    NamaPc = "PC1"
    Rept()
    TDurasi1.Clear()
End Sub
Private Sub Print2_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print2.Click
    n = 1
    NamaPc = "PC2"
    Rept()
    TDurasi2.Clear()
End Sub
Private Sub Print4_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print4.Click
    n = 3
    NamaPc = "PC4"
    Rept()
    TDurasi4.Clear()
End Sub
Private Sub Print5_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print5.Click
    n = 4
    NamaPc = "PC5"
    Rept()
    TDurasi5.Clear()
End Sub
Private Sub Print6_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print6.Click
    n = 5
    NamaPc = "PC6"
    Rept()
    TDurasi6.Clear()
End Sub
Private Sub Print7_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print7.Click
    n = 6
    NamaPc = "PC7"
    Rept()
    TDurasi7.Clear()
End Sub
Private Sub Print8_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print8.Click
    n = 7
    NamaPc = "PC8"
    Rept()
    TDurasi8.Clear()
End Sub
Private Sub Print9_Click(ByVal
sender As System.Object, ByVal e As

```

```

System.EventArgs) Handles
Print9.Click
    n = 8
    NamaPc = "PC9"
    Rept()
    TDurasi9.Clear()
End Sub
Private Sub Print10_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Print10.Click
    n = 9
    NamaPc = "PC10"
    Rept()
    TDurasi10.Clear()
End Sub
#End Region
Private Sub
Lbl7_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkCli
ckedEventArgs)
    changepass.Show()
End Sub
Private Sub
Lbl6_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkCli
ckedEventArgs)
    Remote.Show()
End Sub
Private Sub MDI_FormClosed(ByVal
sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventA
rgs) Handles Me.FormClosed
    Login2.Show()
End Sub
Private Sub MDI_Load(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.listner = New
System.Net.Sockets.Socket(Net.Sockets
.AddressFamily.InterNetwork,
Net.Sockets.SocketType.Stream,
Net.Sockets.ProtocolType.Tcp)
    Connect()
    inisial()
End Sub
Private Sub Tmr1_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr1.Tick
    ' If Sw.IsRunning Then
    Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
    TDurasi1.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
    If paused Then
        ' lbllap.Text =
LblDurasi.Text
        paused = False
    End If
    ' End If
End Sub
Private Sub Tmr3_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr3.Tick
    ' If Sw.IsRunning Then
    Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
    TDurasi3.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
    If paused Then
        ' lbllap.Text =
LblDurasi.Text
        paused = False
    End If
    ' End If
End Sub
Private Sub Tmr2_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr2.Tick
    ' If Sw.IsRunning Then
    Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
    TDurasi2.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
    If paused Then
        ' lbllap.Text =
LblDurasi.Text
        paused = False
    End If
    ' End If
End Sub
Private Sub cleartxt()
    TNama(n).Clear()
    tselesai(n).Clear()
    TStatus(n).Clear()
    Tmulai(n).Clear()
    Tmulai(n).Clear()
    Ttotal(n).Clear()
End Sub
Private Sub
DaftarAnggotaBaruToolStripMenuItem_Cl
ick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles
DaftarAnggotaBaruToolStripMenuItem.Cl
ick
    EAdmin.Show()
End Sub
Private Sub
EntryAnggotaBaruToolStripMenuItem_Cli

```

```

ck(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles
EntryAnggotaBaruToolStripMenuItem.Cli
ck
    EUser.Show()
End Sub
Private Sub
ReportHarianToolStripMenuItem_Click(B
yVal sender As System.Object, ByVal e
As System.EventArgs) Handles
ReportHarianToolStripMenuItem.Click
    FrmPrintLaporan.Show()
End Sub
Private Sub
Lbl8_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkCli
ckedEventArgs) Handles
Lbl8.LinkClicked
    FrmPrintLaporan.Show()
End Sub
Private Sub
Lbl9_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkCli
ckedEventArgs) Handles
Lbl9.LinkClicked
    Rept2()
End Sub
Private Sub
LaporanAnggotaToolStripMenuItem_Click
(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles
LaporanAnggotaToolStripMenuItem.Click
    Rept2()
End Sub
Private Sub
AboutApplicationToolStripMenuItem_Cli
ck(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles
AboutApplicationToolStripMenuItem.Cli
ck
    About.Show()
End Sub
Private Sub
SwitchUserToolStripMenuItem_Click(ByV
al sender As System.Object, ByVal e
As System.EventArgs) Handles
SwitchUserToolStripMenuItem.Click
    Login2.Show()
End Sub
Private Sub
SetHargaPenggunaToolStripMenuItem_Cli
ck(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles
SetHargaPenggunaToolStripMenuItem.Cli
ck
    EHarga.Show()
End Sub

```

```

Private Sub
Lbl5_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkCli
ckedEventArgs)
    EHarga.Show()
End Sub

Private Sub Tmr4_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr4.Tick
    ' If Sw.IsRunning Then
    Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
    TDurasi4.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
    If paused Then
        ' lbllap.Text =
LblDurasi.Text
        paused = False
    End If
    ' End If
End Sub
Private Sub Tmr5_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr5.Tick
    Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
    TDurasi5.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
    If paused Then
        ' lbllap.Text =
LblDurasi.Text
        paused = False
    End If
    End Sub
Private Sub Tmr6_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr6.Tick
    Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
    TDurasi6.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
    If paused Then
        ' lbllap.Text =
LblDurasi.Text
        paused = False
    End If
End Sub

```

```

Private Sub Tmr7_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr7.Tick
Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
TDurasi7.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
If paused Then
' lbllap.Text =
LblDurasi.Text
paused = False
End If
End Sub
Private Sub Tmr8_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr8.Tick
Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
TDurasi8.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
If paused Then
' lbllap.Text =
LblDurasi.Text
paused = False
End If
End Sub
Private Sub Tmr9_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr9.Tick
Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
TDurasi9.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
If paused Then
' lbllap.Text =
LblDurasi.Text
paused = False
End If
End Sub
Private Sub Tmr10_Tick(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Tmr10.Tick
Dim ts As TimeSpan =
TimeSpan.FromSeconds(Sw.Elapsed.Total
Seconds)
TDurasi10.Text =
String.Format(CultureInfo.CurrentCult
ure, "{0:00}:{1:00}:{2:00}",
ts.Hours, ts.Minutes, ts.Seconds)
If paused Then

```

```

' lbllap.Text =
LblDurasi.Text
paused = False
End If
End Sub
End Class
Kode pada Form Set data Harga :
Imports MySql.Data.MySqlClient
Public Class EHarga
Private Objcon As New connect
Private ObjAtur As New AturNum
Private Cmd As New MySqlCommand
Private DR As MySqlDataReader
Public Shared Item As New
ListViewItem
Dim param1, param2, param3,
param4 As MySqlParameter
Private Sub List()
Objcon.CNOpen()
Cmd.Connection = Objcon.Cn
Cmd.CommandType =
CommandType.StoredProcedure
Cmd.CommandText = "vw_harga"
Try
ListView1.Items.Clear()
Objcon.CNOpen()
DR = Cmd.ExecuteReader
Do While DR.Read
Item =
ListView1.Items.Add(DR("level".ToStri
ng))
Item.SubItems.Add(DR("hrgperkoneksi")
.ToString)
Item.SubItems.Add(DR("lapmenit").ToSt
ring)
Item.SubItems.Add(DR("hrgperlapmenit"
).ToString)
Loop
Catch ex As Exception
MsgBox(ex.Message)
Finally
DR.Close()
End Try
End Sub
Private Sub List2()
Dim Sql As String
Sql = "select harga from
t_hrgblnanggota "
Objcon.CNOpen()
Cmd = New MySqlCommand(Sql,
Objcon.Cn)
Try
ListView2.Items.Clear()
Objcon.CNOpen()
DR = Cmd.ExecuteReader
Do While DR.Read

```

```

        Item =
ListView2.Items.Add(DR("harga".ToString))
    Loop
    Catch ex As Exception
        MsgBox("List2" &
ex.Message)
    Finally
        DR.Close()
    End Try
End Sub
Private Sub EHarga_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Objcon.CNOpen()
    Call List()
    Call List2()
End Sub
Private Sub
ListView1_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ListView1.SelectedIndexChanged
    CmbJenis.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(0).Text
    TxtHperkonek.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(1).Text
    CmbPerlap.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(2).Text
    TxtHperLap.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(3).Text
End Sub
Private Sub
ListView2_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ListView2.SelectedIndexChanged
    txtharga.Text =
ListView2.Items(ListView2.FocusedItem
.Index).SubItems(0).Text
End Sub
Private Sub Btnupdate_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Btnupdate.Click
    Call Uharga()
    Call List()
End Sub
Private Sub
BtnUpdate2_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
BtnUpdate2.Click
    Call Uhargabulan()
    Call List2()
End Sub

Private Sub Uhargabulan()
    Dim param1 As MySqlParameter
    Cmd.Connection = Objcon.Cn
    Cmd.CommandType =
CommandType.StoredProcedure
    Cmd.CommandText =
"sp_hrganggota"
    param1 = New
MySqlParameter("pharga",
Me.txtharga.Text)
    Cmd.Parameters.Add(param1)
    Try
        Objcon.CNOpen()
        Cmd.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox(ex.Message)
    Finally
        Objcon.Cnclose()
    End Try
End Sub
Private Sub Uharga()
    Cmd.Connection = Objcon.Cn
    Cmd.CommandType =
CommandType.StoredProcedure
    Cmd.CommandText = "sp_uharga"
    param1 = New
MySqlParameter("plevel",
CmbJenis.Text)
    param2 = New
MySqlParameter("pkonek",
TxtHperkonek.Text)
    param3 = New
MySqlParameter("plap",
CmbPerlap.Text)
    param4 = New
MySqlParameter("phrglap",
TxtHperLap.Text)
    Cmd.Parameters.Add(param1)
    Cmd.Parameters.Add(param2)
    Cmd.Parameters.Add(param3)
    Cmd.Parameters.Add(param4)
    Try
        Objcon.CNOpen()
        Cmd.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox(ex.Message)
    Finally
        Objcon.Cnclose()
    End Try
End Sub
Private Sub Btncancel_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Btncancel.Click
    CmbJenis.Text = ""
    TxtHperkonek.Text = ""
    CmbPerlap.Text = ""
    TxtHperLap.Text = ""
End Sub

```



```

        Private Sub
        TxtHperkonek_KeyDown(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventArgs)
Handles TxtHperkonek.KeyDown
        ObjAtur.OnlyNumber(e)
        End Sub
        Private Sub
        TxtHperLap_KeyDown(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventArgs)
Handles TxtHperLap.KeyDown
        ObjAtur.OnlyNumber(e)
        End Sub
        Private Sub
        txtharga_KeyDown(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventArgs)
Handles txtharga.KeyDown
        ObjAtur.OnlyNumber(e)
        End Sub
    End Class
Form Entri Anggota Baru :
Imports MySql.Data.MySqlClient
Public Class EUser
    Dim Objcon As New connect
    Dim Cmd As New MySqlCommand
    Dim Adap As New MySqlDataAdapter
    Dim DR As MySqlDataReader
    Public Shared Item As
ListViewItem
    Dim StrXmlName As String
    Private objAtur As New AturNum
    Private Sub List()
        Dim SQL As String
        SQL = "select
ID,nama,password,tgldaftar,tglakhir,s
tatus,bulan,harga,harga * bulan as
total from t_anggota order by
tgldaftar"
        Objcon.CNOpen()
        Cmd = New MySqlCommand(SQL,
Objcon.Cn)
        ListView1.Items.Clear()
        Objcon.CNOpen()
        DR = Cmd.ExecuteReader
        Do While DR.Read
            Item =
ListView1.Items.Add(DR("ID").ToString
)
            Item.SubItems.Add(DR("nama").ToString
)
            Item.SubItems.Add(DR("password").ToSt
ring)
            Item.SubItems.Add(DR("tgldaftar").ToSt
ring))
            Item.SubItems.Add(DR("tglakhir").ToStr
ing))
            Item.SubItems.Add(DR("status").ToStri
ng)
            Item.SubItems.Add(DR("bulan").ToStrin
g)
            Item.SubItems.Add(DR("harga").ToStrin
g)
            Item.SubItems.Add(DR("total").ToStrin
g)
                'Cmd.Dispose()
                Loop
                'Catch ex As Exception
                'MsgBox("ERR : " &
ex.Message)
                ' Finally
                DR.Close()
                Objcon.Cnclose()
                ' End Try
        End Sub
        Private Sub
        ListView1_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
        ListView1.SelectedIndexChanged
            txtID.Text =
        ListView1.Items(ListView1.FocusedItem
.Index).SubItems(0).Text
            Txtnama.Text =
        ListView1.Items(ListView1.FocusedItem
.Index).SubItems(1).Text
            txtpass.Text =
        ListView1.Items(ListView1.FocusedItem
.Index).SubItems(2).Text
            cmbbln.Text =
        ListView1.Items(ListView1.FocusedItem
.Index).SubItems(6).Text
        End Sub
        Private Sub EUser_Load(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
            Objcon.CNOpen()
            Call List()
            Randomize()
        End Sub
        Private Sub Btnsave_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
        Btnsave.Click
            If Txtnama.Text = "" Or
cmbbln.Text = "" Then
                MsgBox("Data Kosong, Isi
data bulan dan Nama !!")
            Else
                random()
            End If
        End Sub
    End Class

```

```

        Call Simpan()
        Call List()
        Rept()
    End If

End Sub
Private Sub Rept()
    Dim DS As New DataSet
    Dim Sql2 As String
    StrXmlName = "anggota"
    Dim strXmlPath As String =
Application.StartupPath & "\" & _
StrXmlName & ".xml"
    If Txtnama.Text <> "" Then
        Sql2 = "select
ID,nama,password,DATE_FORMAT(tgldaftar, '%d-%b-%Y')as
TglDaftar,DATE_FORMAT(tglakhir, '%d-
%b-%Y')as TglAkhir,bulan as
LamaBulan,concat('Rp.',harga) as
Harga,concat('Rp.', bulan * harga) as
TotalHarga from t_anggota where nama
=' & Txtnama.Text & ' ' order by id
desc limit 1"
        Objcon.CNOpen()
        Try
            Adap = New
MySQLDataAdapter(Sql2, Objcon.Cn)
            Adap.Fill(DS)

DS.WriteXml(strXmlPath)
            FrmRptAnggota.Show()
        Catch ex As Exception
            MsgBox("ERR : " &
ex.Message)
        End Try
        Objcon.Cnclose()
    End If
End Sub
Private Sub Simpan()
    Dim param1, param2, param3 As
MySQLParameter
    Objcon.CNOpen()
    Cmd.Connection = Objcon.Cn
    Cmd.CommandType =
CommandType.StoredProcedure
    Cmd.CommandText =
"sp_insertanggota"
    param1 = New
MySQLParameter("pnama", Txtnama.Text)
    param2 = New
MySQLParameter("ppass", txtpass.Text)
    param3 = New
MySQLParameter("pbln", cmbbln.Text)
    Cmd.Parameters.Add(param1)
    Cmd.Parameters.Add(param2)
    Cmd.Parameters.Add(param3)
    Try
        Cmd.ExecuteNonQuery()
    Catch ex As Exception

```

```

        MsgBox("SAVE ERR: " &
ex.Message)
    Finally
        Objcon.Cnclose()
    End Try
End Sub
Private Sub random()
    txtpass.Text = CStr(Int(Rnd()
* 1000000))
End Sub
Private Sub btndelete_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btndelete.Click
    Call DEL()
    Call List()
End Sub
Private Sub DEL()
    Dim Sql As String
    Sql = "delete from t_anggota
where ID = " & txtID.Text & "'
    Objcon.CNOpen()
    Cmd = New MySqlCommand(Sql,
Objcon.Cn)
    Try
        Cmd.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox("ERR DELETE : " &
ex.Message)
    End Try
    Objcon.Cnclose()
End Sub
Private Sub Btncancel_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Btncancel.Click
    Txtnama.Clear()
    txtpass.Clear()
    cmbbln.Text = ""
    txtID.Clear()
    Txtnama.Focus()
End Sub
Private Sub Btnupdate_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Btnupdate.Click
    random()
    Call Ubah()
    Call List()
End Sub
Private Sub Ubah()
    Dim Sql As String
    Sql = "update t_anggota set
nama=' & Txtnama.Text & ',' & _
"password=' &
txtpass.Text & ' ' where ID = " &
txtID.Text & "'
    Objcon.CNOpen()
    Cmd = New MySqlCommand(Sql,
Objcon.Cn)

```

```

Try
    Cmd.ExecuteNonQuery()
Catch ex As Exception
    MsgBox("ERR UPDATE : " &
ex.Message)
End Try
Objcon.Cnclose()
End Sub
Private Sub cari()
    Dim SQL As String
    SQL = "select
ID,nama,password,tgldaftar,tglakhir,s
tatus,bulan,harga,harga * bulan as
total from t_anggota where nama like
'%" & Txtnama.Text & "%' order by
nama"

    Objcon.CNOpen()
    Cmd = New MySqlCommand(SQL,
Objcon.Cn)
    ListView1.Items.Clear()
    Objcon.CNOpen()
    DR = Cmd.ExecuteReader
    Do While DR.Read
        Item =
ListView1.Items.Add(DR("ID").ToString)
)

Item.SubItems.Add(DR("nama").ToString)
)

Item.SubItems.Add(DR("password").ToSt
ring)

Item.SubItems.Add(DR("tgldaftar".ToSt
ring))

Item.SubItems.Add(DR("tglakhir".ToStr
ing))

Item.SubItems.Add(DR("status").ToStri
ng)

Item.SubItems.Add(DR("bulan").ToStrin
g)

Item.SubItems.Add(DR("harga").ToStrin
g)

Item.SubItems.Add(DR("total").ToStrin
g)

'Cmd.Dispose()
Loop
End Sub
Private Sub Btnsearch_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Btnsearch.Click
    cari()
End Sub

```

```

End Class
Form Entri User(Admin Atau Operator)
:
Imports MySql.Data.MySqlClient
Public Class EAdmin
    Dim Objcon As New connect
    Dim Cmd As New MySqlCommand
    Dim DR As MySqlDataReader
    Public Shared Item As
ListViewItem
    Private objAtur As New AturNum
    Private Sub List()
        Dim Sql As String
        Sql = "SELECT
ID,level,nama,password from t_user "
        Cmd = New MySqlCommand(Sql,
Objcon.Cn)
        ListView1.Items.Clear()
        Objcon.CNOpen()
        'try
        DR = Cmd.ExecuteReader
        Do While DR.Read
            Item =
ListView1.Items.Add(DR("ID").ToString)
)

Item.SubItems.Add(DR("nama").ToString)
)

Item.SubItems.Add(DR("level").ToStrin
g)

Item.SubItems.Add(DR("password".ToStr
ing))

'Cmd.Dispose()
Loop
'Catch ex As Exception
'MsgBox("ERR : " &
ex.Message)
' Finally
DR.Close()
Objcon.Cnclose()
' End Try
End Sub
Private Sub
ListView1_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ListView1.SelectedIndexChanged
    txtID.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(0).Text
    Txtnama.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(1).Text
    Txtpass.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(3).Text

```

```

        Cmblevel.Text =
ListView1.Items(ListView1.FocusedItem
.Index).SubItems(2).Text
    End Sub
    Private Sub EUser_Load(ByVal
sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        Objcon.CNOpen()
        Call List()
    End Sub
    Private Sub Btncancel_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Btncancel.Click
        Call clear()
    End Sub
    Private Sub btndelete_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btndelete.Click
        Call Del()
        Objcon.CNOpen()
        Call List()
    End Sub
    Private Sub Del()
        Dim Sql As String

```

```

        Sql = " delete from t_user
where ID ='" & txtID.Text & "'"
        Objcon.CNOpen()
        Cmd = New MySqlCommand(Sql,
Objcon.Cn)
        Try
            Cmd.ExecuteNonQuery()
        Catch ex As Exception
            MsgBox("ERR DELETE : " &
ex.Message)
        End Try
        Objcon.Cnclose()
    End Sub
    Private Sub Btnsearch_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
Btnsearch.Click
        Dim sql As String
        Objcon.CNOpen()
        sql = "select * from t_user
where t_user.nama like '%" &
Txtnama.Text & "%'"
        Cmd = New MySqlCommand(sql,
Objcon.Cn)
        DR = Cmd.ExecuteReader
        ListView1.Items.Clear()
        Do While DR.Read
            Item =
ListView1.Items.Add(DR("id").ToString
)
            Item.SubItems.Add(DR("nama".ToString)
)
            Item.SubItems.Add(DR("level".ToString)
)
            Item.SubItems.Add(DR("password".ToStr
ing))
        Loop
        DR.Close()
        Objcon.Cnclose()
    End Sub
    Private Sub btnShow_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btnShow.Click
        Objcon.CNOpen()
        Call List()
    End Sub
End Class
Code Form Remote :
Imports System.Net
Imports System.Net.Sockets
Imports System.Threading
Imports System.IO
Public Class Remote
    Dim ipAddress As IPAddress =
Dns.Resolve(Dns.GetHostName()).Address
sList(0)

```

```

Dim trSendMessage As Thread
Dim MYIP As String
Sub SendMessage()
    Dim host As String =
    TxtClientIP.Text
    Dim port As Integer = 63000
    Try
        Dim tcpCli As New
    TcpClient(host, port)
        Dim ns As NetworkStream =
    tcpCli.GetStream
        ' Send data to the
    server.
        Dim sw As New
    StreamWriter(ns)
        If rbShutdown.Checked =
    True Then
            sw.WriteLine("###SHUTDOWN###")
            End If
            If rbReboot.Checked =
    True Then
                sw.WriteLine("###REBOOT###")
                End If
                If rbLogOff.Checked =
    True Then
                    sw.WriteLine("###LOGOFF###")
                    End If
                    If rbNothing.Checked =
    True Then
                        sw.WriteLine("to")
                        End If
                        sw.Flush()
                        ' Receive and display
                    data.
                    Dim sr As New
                StreamReader(ns)
                    Dim result As String =
                sr.ReadLine()
                    If result = "###OK###"
                Then
                    MsgBox("Operation
                Performed!!!",
                MsgBoxStyle.Information, "Accepted by
                client")
                    End If
                    'MsgBox(result)
                    sr.Close()
                    sw.Close()
                    ns.Close()
                Catch ex As Exception
                    MsgBox(ex.Message)
                End Try
            End Sub
            Private Sub BtnStart_Click(ByVal
            sender As System.Object, ByVal e As
            System.EventArgs) Handles
            BtnStart.Click

```

```

        If Trim(TxtClientIP.Text) =
    "" Then
            MsgBox("Please enter
            Client IP Address!!!",
            MsgBoxStyle.Information, "IP
            Address???" )
            TxtClientIP.Focus()
            Exit Sub
        End If
        If rbNothing.Checked = False
    Then
            trSendMessage = New
            Thread(AddressOf SendMessage)
            trSendMessage.Start()
            End If
        End Sub
    End Class
Kode Form Ganti Password :
Imports MySql.Data.MySqlClient
Public Class changepass
    Public Savepass As String
    Public SaveUser As String
    Dim Objcon As New connect
    Dim Cmd As New MySqlCommand
    Dim DR As MySqlDataReader
    ' FIX SELESAI
    Sub _changePass()
        Dim sql As String
        sql = "update t_user set
    password=' ' & txtnpass2.Text & "' &
    -
        "where nama = ' " &
    SaveUser & "' "
        If MsgBox("Anda yakin ingin
    merubah password ? ",
    MsgBoxStyle.YesNo, "Change Password")
    = MsgBoxResult.Yes Then
            Objcon.CNOpen()
            Cmd = New
            MySqlCommand(sql, Objcon.Cn)
            Objcon.CNOpen()
            Try
                Cmd.ExecuteNonQuery()
            PASSkosong()
            Catch ex As Exception
                MsgBox("Change
            Password ERROR" & ex.Message,
            MsgBoxStyle.Exclamation, "ERROR")
            End Try
            Objcon.Cnclose()
        End If
    End Sub
    Private Sub BtnCPass_Click(ByVal
    sender As System.Object, ByVal e As
    System.EventArgs) Handles
    BtnCPass.Click
        If Txtpass.Text = "" Or
    Txtnpass1.Text = "" Or txtnpass2.Text
    = "" Then

```

```

        MsgBox("Data belum
Lengkap! ", MsgBoxStyle.Information,
"Information")
    Else
        If Txtypass.Text =
Savepass Then
            If Txtnpass1.Text =
txtnpass2.Text Then
                Call
_changePass()
            Else
                MsgBox("Password
yang diketik tidak sama, mohon ketik
ulang password.",
MsgBoxStyle.Information,
"Information")
            End If
        Else
            MsgBox("Password lama
Anda Salah !!",
MsgBoxStyle.Information,
"Information")
        End If
    End If
End Sub
Private Sub BtnCancel_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
BtnCancel.Click
    PASSkosong()
End Sub
Sub PASSkosong()
    Txtnpass1.Text = ""
    txtnpass2.Text = ""
    Txtypass.Text = ""
End Sub
End Sub
End Class

```

Kode Form Laporan Bulanan dan harian
:

```

Imports MySql.Data.MySqlClient
Imports System.Windows.Forms

Public Class FrmPrintLaporan
    Private Objcon As New connect
    Private Cmd As New MySqlCommand
    Private Adap As New
MySqlDataAdapter
    Private DR As MySqlDataReader
    Private StrXmlName As String
    Private Sub
FrmPrintLaporan_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        CmbBulan.Text = "Januari"
        Txttahun.Text = Date.Now.Year
    End Sub
    Private Sub Rept1()
        Dim DS As New DataSet
        Dim Sql1 As String

```

```

        StrXmlName = "LapHarian"
        Dim strXmlPath As String =
Application.StartupPath & "\" & _
StrXmlName & ".xml"
        If DTPicker.Text <> "" Then
            Sql1 = "select
ID,level,nopc,nama,TIME_FORMAT(mulai,
'%H:%i:%s')as
mulai,TIME_FORMAT(selesai,'%H:%i:%s')
as
selesai,TIME_FORMAT(durasi,'%H:%i:%s'
)as
durasi,totalbiaya,DATE_FORMAT(tanggal
,'%d-%b-%Y')as Tanggal,operator from
t_biaya_pemakaian where tanggal ='" &
DTPicker.Text & "'"
            Objcon.CNOpen()
            Try
                Adap = New
MySqlDataAdapter(Sql1, Objcon.Cn)
                Adap.Fill(DS)
                'Menulis Skema Ke XML
DS.WriteXml(strXmlPath)
                ' Lapl.MdiParent =
MDI
                FrmRptHarian1.Show()
                ' Cleartxt1()
                ' DR.Close()
                Catch ex As Exception
                    MsgBox("ERR : " &
ex.Message)
                End Try
            Objcon.CNclose()
        End If
    End Sub
    Private Sub btnprintd_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btnprintd.Click
        Call bulan()
        Dim DS As New DataSet
        Dim Sql2 As String
        StrXmlName = "LapBulan"
        Dim strXmlPath As String =
Application.StartupPath & "\" & _
StrXmlName & ".xml"
        If CmbBulan.Text <> "" Or
Txttahun.Text <> "" Then
            Sql2 = "select
ID,level,nopc,nama,TIME_FORMAT(mulai,
'%H:%i:%s')as
mulai,TIME_FORMAT(selesai,'%H:%i:%s')
as
selesai,TIME_FORMAT(durasi,'%H:%i:%s'
)as
durasi,totalbiaya,DATE_FORMAT(tanggal
,'%d-%b-%Y')as Tanggal,operator from
t_biaya_pemakaian where
MONTH(tanggal) = '" & CmbBulan.Text &

```

```

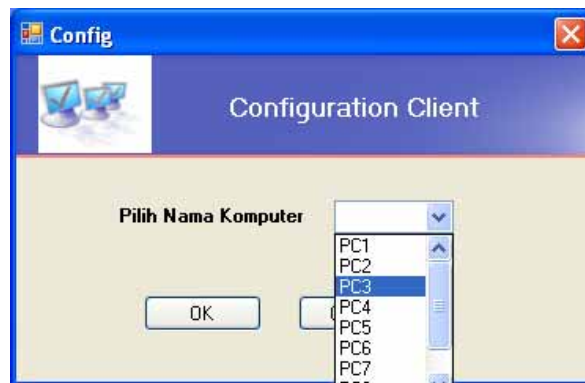
'' and year(tanggal) = '' &
Txttahun.Text & ""
    Objcon.CNOpen()
    Try
        Adap = New
        MySqlDataAdapter(Sql2, Objcon.Cn)
        Adap.Fill(DS)
        'Menulis Skema Ke XML
        DS.WriteXml(strXmlPath)
        FrmRptBulan.Show()
        ' DR.Close()
    Catch ex As Exception
        MsgBox("ERR :" &
ex.Message)
    End Try
    Objcon.CNclose()
    Cleartxt2()
End If
End Sub
Public Sub SetDate()
    DTPicker.Format =
DateTimePickerFormat.Custom
    DTPicker.CustomFormat =
"yyyy-MM-dd"
End Sub 'SetMyCustomForm
Public Sub SetDate2()
    DTPicker.Format =
DateTimePickerFormat.Custom
    DTPicker.CustomFormat =
"dd,MMMM yyyy"
End Sub
Sub Cleartxt2()
    CmbBulan.Text = "Januari"
    Txttahun.Text = Date.Now.Year
End Sub
Private Sub bulan()
    Select Case CmbBulan.Text
        Case "Januari"
            CmbBulan.Text = "1"
        Case "Februari"
            CmbBulan.Text = "2"
        Case "Maret"
            CmbBulan.Text = "3"
        Case "April"
            CmbBulan.Text = "4"
        Case "Mei"
            CmbBulan.Text = "5"
        Case "Juni"
            CmbBulan.Text = "6"
        Case "Juli"
            CmbBulan.Text = "7"
        Case "Agustus"
            CmbBulan.Text = "8"
        Case "September"
            CmbBulan.Text = "9"
        Case "Oktober"
            CmbBulan.Text = "10"
        Case "November"
            CmbBulan.Text = "11"
        Case "Desember"
            CmbBulan.Text = "12"
    End Select
End Sub
Private Sub btnprint_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btnprint.Click
    SetDate()
    Rept1()
    SetDate2()
End Sub
End Class

```

LAMPIRAN IV INTERFACE CLIENT



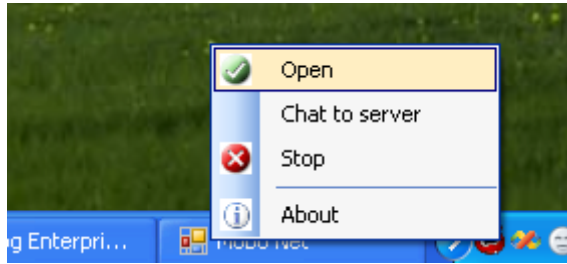
Tampilan Form Login



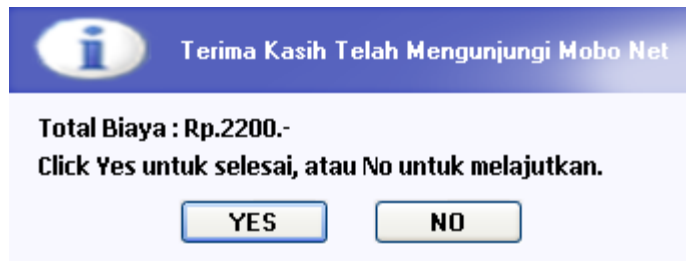
Tampilan Form Konfigurasi



Tampilan Form Proses



System Tray pada Taksbar



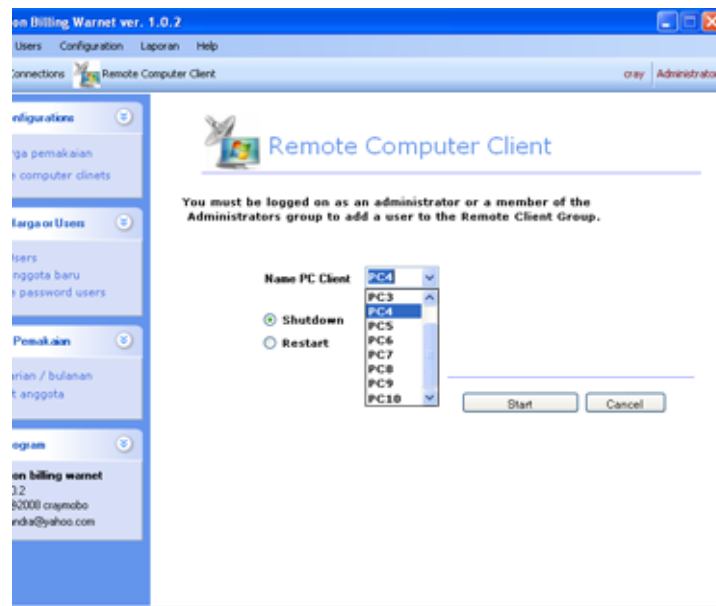
Konfirmasi Pesan pengguna

LAMPIRAN V INTERFACE SERVER

Tampilan Form Login

Tampilan Form Menu Utama

Tampilan Form Set Harga Pemakaian



Tampilan Form Remote client




Tampilan Form Entri Anggota



Tampilan Form Print hari dan bulan

Laporan Harian Billing warnet

Main Report

 **Laporan Harian Billing internet Mobo NET**

Tanggal : 11/01/2009


No	Status	NoPc	Nama	Mulai	Selesai	Durasi	TotalBiaya	Tanggal	operator
1	Non Anggota	PC1	M TEGUH	01:27:57	01:45:57	00:17:58	1400	11-Jan-2009	cray
2	Non Anggota	PC1	JASMINE	01:50:20	02:24:48	00:34:26	2200	11-Jan-2009	cray
3	Non Anggota	PC5	CHANDRA	23:04:00	23:46:22	00:42:21	2600	11-Jan-2009	cray
4	Anggota	PC5	ANDRA	02:59:11	03:19:58	00:20:45	1700	11-Jan-2009	cray
5	Anggota	PC5	SETYO	03:37:07	04:02:35	00:25:26	1700	11-Jan-2009	oper
6	Non Anggota	PC10	Melatie	04:37:21	05:22:27	00:45:05	2600	11-Jan-2009	oper

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

Tampilan Form Laporan Harian

Laporan Bulanan

Main Report

 **LAPORAN BULANAN BILING WARNET MOBO NET**

Laporan Tanggal: 17-Jan-2009 Tanggal Print: 1/19/2009

No	Status	NoPc	Nama	Mulai	Selesai	Durasi	Total	Tanggal	Operator
1	Non Anggota	PC4	CRAYMOBO	03:19:24	03:24:40	00:05:15	1900	17-Jan-2009	User name
2	Non Anggota	PC3	CRAYON	05:41:24	05:41:29	00:00:04	1000	17-Jan-2009	User name
3	Non Anggota	PC3	CHANDRA	05:42:01	05:42:15	00:00:14	1000	17-Jan-2009	User name
4	Non Anggota	PC3	crayon	05:25:33	05:25:49	00:00:17	1000	17-Jan-2009	cray
5	Non Anggota	PC3	TEST	06:26:17	06:26:26	00:00:10	1000	17-Jan-2009	cray
6	Non Anggota	PC3	test	06:26:34	06:26:43	00:00:10	1000	17-Jan-2009	cray
7	Non Anggota	PC3	testif	06:26:45	06:26:52	00:00:10	2000	17-Jan-2009	cray

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

Tampilan Form Laporan Bulanan

 Universitas Mercu Buana
Teknik Informatika

Application Billing Internet Clinet Server

Version 1.0.2 Build 01 - Build On Nov 12 2008

About Application **CRAY MOBO**

Contact : email mobo_chandra@hotmail.com | ym : andra_mobo

Tampilan Form About aplikasi