

## IMPLEMENTASI TCP/IP UNTUK MEMBUAT SERVER DATABASE ACCESS

**Yones Raven, Teady Matius Surya Mulyana**  
yonesraven@gmail.com, tmulyana@bundamulia.ac.id  
Program Studi Teknik Informatika, Universitas Bunda Mulia

### ABSTRAK

Database Access adalah salah satu perangkat lunak yang didesain untuk membantu memelihara dan memanfaatkan kumpulan data yang cukup besar. Tetapi ada suatu kekurangan pada perangkat lunak ini, yaitu database access tidak dapat menjadi *database server*. Tujuan penelitian ini adalah untuk membuktikan database access dapat menjadi *server* database access dengan cara mengimplementasikan TCP/IP.

*Socket* adalah suatu abstraksi yang mana aplikasi dapat mengirim dan menerima data sama halnya dengan membuka suatu file untuk dibaca dan ditulis pada tempat penyimpanan file. *Connection-oriented Protocol* adalah protokol yang dapat mengirim data stream setelah terjadi koneksi dengan target. TCP/IP merupakan salah satu protokol yang memanfaatkan connection-oriented.

Dengan mengimplementasikan TCP/IP, database access dapat menjadi database *server* dimana aplikasi *server* dapat menerima permintaan dari aplikasi *client* yang akan dieksekusi ke database access dan mengirim hasil proses tersebut ke aplikasi *client*. Hasil penelitian ini adalah aplikasi yang dibuat dengan mengimplementasikan TCP/IP mampu membuat database access menjadi *server* database access.

Berdasarkan pengujian, maka dapat disimpulkan bahwa TCP/IP layak sebagai sarana pengiriman data ke aplikasi lain. Berdasarkan penelitian yang telah dilakukan, maka disarankan dalam melakukan penelitian untuk TCP/IP dengan mengirim data, secara dasar data yang diterima berupa tipe data string sehingga dibutuhkan suatu method yang digunakan untuk mengkonversi tipe data yang diinginkan.

**Kata Kunci :** *Database Access, aplikasi client, aplikasi server, TCP/IP, socket, database server*

---

### PENDAHULUAN

Teknologi komputer saat ini berkembang sangat cepat, baik dari sisi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang ada. Karena begitu cepatnya perkembangan teknologi ini, manusia dapat mengakses segala jenis informasi dengan mudah dan cepat meskipun jarak yang berjauhan sehingga dapat meningkatkan efisiensi dan efektifitas dalam melakukan pekerjaan.

Dari banyaknya teknologi yang telah ditemukan dan dikembangkan, Database Access adalah salah satu perangkat lunak yang didesain untuk membantu memelihara dan memanfaatkan kumpulan

data yang cukup besar. Tetapi ada suatu kekurangan pada perangkat lunak ini, yaitu database access tidak dapat menjadi *database server*.

Berdasarkan latar belakang tersebut, penulis mengangkat kasus ini menjadi skripsi dengan mengambil judul: *Implementasi TCP/IP Untuk Membuat Server Database Access*.

Tujuan dari penelitian dibangunnya "Implementasi TCP/IP untuk Membuat *Server Database Access*" adalah membuktikan database access dapat menjadi *server* database access dengan cara mengimplementasikan TCP/IP.

## KERANGKA TEORI

Landasan teori ini menggunakan teori *socket*, *Connection-Oriented Protocol* dan Implementasi *Socket* Dengan Java.

### *Socket*

Kurniawan (Kurniawan, 2011) menjelaskan, *socket* adalah suatu abstraksi yang mana aplikasi dapat mengirim dan menerima data sama halnya dengan membuka suatu file untuk dibaca dan ditulis pada tempat penyimpanan file. *Socket* memungkinkan untuk masuk ke dalam jaringan dan berkomunikasi dengan aplikasi lain yang juga masuk ke dalam jaringan yang sama. Informasi yang ditulis ke dalam *socket* pada suatu aplikasi pada suatu mesin dapat dibaca oleh aplikasi lain pada mesin yang berbeda dan sebaliknya.

### Connection-Oriented Protocol

Kurniawan (Kurniawan, 2011) menjelaskan, jaringan protocol berbasis *connection-oriented* adalah protokol yang dapat mengirim data stream setelah terjadi koneksi dengan target. Bentuknya dapat berupa *circuit switched connection* atau *virtual circuit connection* pada suatu paket *switched network*. Pada protokol ini kepastian paket data diterima oleh *client* dapat diandalkan.

Berikut ini beberapa contoh protokol yang memanfaatkan *connection-oriented*:

- TCP
- DCCP
- *Connection-oriented Ethernet*
- *Phone Call*
- ATM
- *Frame Relay*
- TIPC
- SCTP
- IPX/SPX

Protokol ini banyak digunakan pada aplikasi yang memang membutuhkan kepastian bahwa paket data yang dikirim dapat diterima dengan sukses pada target yang diinginkan. Pada umumnya digunakan protokol TCP/IP. Contoh aplikasi yang memanfaatkan TCP/IP adalah browser dan aplikasi chatting seperti Yahoo Messenger dan MSN.

## Implementasi *Socket* Dengan Java

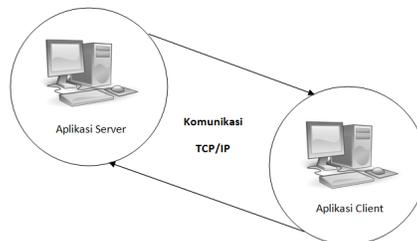
Kurniawan (Kurniawan, 2011) menjelaskan, realisasi *socket* pada Java, yaitu dengan memanfaatkan *package* *java.net*. Semua objek yang berhubungan jaringan dapat memanfaatkan ini. Khusus untuk *socket* dapat menggunakan kelas

- *Socket*
- *ServerSocket*
- *SocketAddress*

Kelas *Socket* digunakan untuk *socket client* dan *ServerSocket* untuk *socket server* sedangkan *SocketAddress* untuk informasi *socket* yang akan digunakan sebagai target *socket*. Aplikasi *server* akan menunggu koneksi dari aplikasi *client*. Ketika koneksi antara *server* dan *client* telah terhubung, aplikasi *server* dan *client* saling mengirim data.

## PERANCANGAN SISTEM

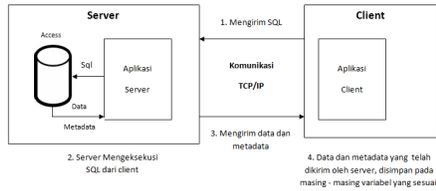
Tahap perancangan sistem merupakan kelanjutan dari kerangka teori yang dijelaskan, sehingga data yang akan dibuat dapat disusun dengan mudah, dan tepat pada sasaran yang telah ditetapkan sebelum menyusun suatu aplikasi. Langkah yang harus dilakukan adalah dengan membuat rancangan aplikasi terhadap permasalahan yang dibahas dengan harapan agar pembuatan aplikasi tidak meluas dari pokok permasalahan.



Gambar 1 Skenario TCP/IP

Skenario TCP/IP pada gambar 1 dibuat sesuai dengan skenario Kurniawan (Kurniawan, 2011) yang menjelaskan aplikasi *server* akan menunggu koneksi dari aplikasi *client*. Ketika koneksi antara *server* dan *client* telah terhubung, aplikasi *server* dan *client* saling mengirim data.

Skenario dari Gambar 1 dapat dikembangkan menjadi perancangan TCP/IP untuk membuat *server* database access, dapat dilihat pada gambar 2.



**Gambar 2 Perancangan TCP/IP Untuk Membuat Server Database Access**

Pada Gambar 2 menggambarkan aplikasi *server* menunggu permintaan dari aplikasi *client*, hal ini dilaksanakan dengan menggunakan objek *Socket*. Ketika koneksi antara *server* dan *client* telah terhubung, aplikasi *client* dapat mengirimkan perintah sql ke aplikasi *server*.

Aplikasi *server* merupakan aplikasi yang berjalan pada komputer yang memberikan layanan aplikasi. *Server* bersifat pasif, artinya bila tidak ada permintaan layanan dari *client*, maka program *server* menunggu tanpa memerlukan sumber daya CPU Cycle.

Aplikasi *client* merupakan aplikasi yang berjalan pada komputer yang meminta layanan kepada *server*. Permintaan ini disampaikan kepada *server* dalam bentuk pesan sql.

Ketika aplikasi *client* mengirim perintah sql, aplikasi *server* menerima permintaan dari *client* dan mengeksekusi sql dari *client* yang terhubung dengan database access. Setelah tersekusi sesuai perintah sql dari *client*, database access memberikan data serta metadata ke aplikasi *server* dan aplikasi *server* mengirim metadata dan data tersebut kepada *client* sesuai permintaan *client*.

Server
- server x; - Connection koneksi; - Statement sql; - ResultSet hasil; - ResultSetMetaData rsMeta - ServerSocket s; - Socket cs; - InetAddress ia; - String cli; - BufferedReader in; - PrintWriter out;
+ void dbInsert(String query, PrintWriter out) + void dbSelect(String query, PrintWriter out)

**Gambar 3 Class Server**

Selanjutnya *client* menerima data dan metadata (informasi dari sebuah data)

yang telah diolah oleh aplikasi *server* dan menyimpan data serta metadata tersebut pada masing – masing variabel yang dapat diakses oleh *client*.

Gambar 3 dan gambar 4 merupakan class diagram yang menggambarkan sisi *server* dan sisi *client*, sebagai berikut:

Atribut – atribut dari class *server* terdiri dari :

- *server x* : Digunakan untuk memanggil fungsi pada class *server* pada java.
- Connection koneksi : Digunakan untuk membuat koneksi pada database access.
- Statement sql : Digunakan untuk menampung query yang akan dieksekusi ke database access.
- ResultSet hasil : Digunakan untuk menampung data dari hasil eksekusi query dalam pengambilan data.
- ResultSetMetaData rsMeta : Digunakan untuk menampung metadata dari hasil eksekusi query dalam pengambilan data.
- *ServerSocket s*: Digunakan untuk membuat *socket server* sebagai aplikasi *server*.
- *Socket cs*: Digunakan untuk masuk ke dalam jaringan dan berkomunikasi dengan aplikasi *client* yang juga masuk ke dalam jaringan yang sama.
- *InetAddress ia*: Digunakan untuk mendapatkan ip client yang terkoneksi pada *server*.
- *String cli*: Digunakan untuk menampung ip address *client*.
- *BufferedReader in*: Digunakan untuk menerima permintaan dari client.
- *PrintWriter Out*: Digunakan untuk mengirim data sesuai perintah yang telah dikirim oleh *client*.

Method – method pada class *server* terdiri dari:

- void dbinsert(String query, PrintWriter out) : Method yang digunakan untuk mengeksekusi query saat tambah, ubah dan hapus data pada database access.
- void dbselect(String query, PrintWriter out): Method yang digunakan untuk mengeksekusi query saat pengambilan data dari database access.

Proses yang terjadi pada sisi *server*, dimulai dari membuat objek *ServerSocket* dan menentukan port *server* yang digunakan

untuk membuat *socket server* sebagai aplikasi *server*. Setelah itu membuat objek *socket* dan *inetaddress* untuk dapat menerima permintaan dari *client* serta mengetahui alamat *client*.

Selanjutnya membuat input dan output stream untuk membaca dan membalas respon dari *client*. Setelah itu aplikasi *server* menunggu permintaan dari *client*. Ketika *client* mengirim permintaan ke *server*, maka aplikasi *server* membuat koneksi pada database access dan memvalidasi perintah sql, apakah sql tersebut sql pengambilan data, jika ya maka menjalankan method *dbselect*, jika tidak maka akan menjalankan method *dbinsert*.

Client
-String Server; -String [][] data; -String [] ColumnName; -Integer[] ColumnType; -int jumlahkolom; -int jumlahbaris -int posisiField; -int posisiColumn; -Socket con; -BufferedReader in; -PrintWriter out;
+void executeQuery(String input) +boolean next() +String getString(String NamaField) +Integer getInteger(String NamaField) +Double getDouble(String NamaField) +String getColumnName(int x) +Integer getColumnType(int x) +Integer getColumnCount() +Integer getColumnRow()

**Gambar 4 Class Client**

Method *dbselect* digunakan untuk mengesekusi sql pengambilan data, *dbselect* mengesekusi query untuk pengambilan data yang telah diterima dari *client*. Selanjutnya menampung data dari query yang telah dieksekusi pada variabel yang disediakan. Setelah data telah diperoleh, maka dilanjutkan mengirim semua data yang dibutuhkan ke *client* dengan mengimplementasikan TCP/IP seperti jumlah baris, jumlah kolom, nama kolom, tipe data, dan isi record data.

Sedangkan method *dbinsert* digunakan untuk mengesekusi sql manipulasi data, proses. Method *dbinsert* mengesekusi query untuk memanipulasi data yang telah diterima dari *client*, dilanjutkan dengan memberi respon ke *client* apakah query berhasil dieksekusi atau gagal dieksekusi.

Atribut – atribut dari class *client* terdiri dari :

- String *server*: Digunakan untuk menampung ip address *server*.
- String [][] data: Digunakan untuk menampung record data yang dikirim oleh *server*.
- String [] ColumnName: Digunakan untuk menampung nama kolom sesuai data yang telah diolah dan dikirim oleh *server*.
- Integer[] ColumnType: Digunakan untuk menampung tipe data nama kolom sesuai data yang telah diolah dan dikirim oleh *server*.
- int jumlahKolom: Digunakan untuk menampung jumlah kolom yang dikirimkan oleh *server*.
- int jumlahBaris: Digunakan untuk menampung jumlah baris yang dikirimkan oleh *server*.
- int posisiField: Digunakan untuk mengetahui posisi baris data.
- int posisiColumn: Digunakan untuk mengetahui posisi kolom data berada.
- Socket con: Digunakan untuk masuk ke dalam jaringan dan berkomunikasi dengan aplikasi *server* yang juga masuk ke dalam jaringan yang sama.
- BufferedReader in: Digunakan untuk menerima data yang dikirim oleh *server*.
- PrintWriter Out: Digunakan untuk mengirim permintaan kepada *server*.

Method – method pada class *client* terdiri dari:

- void executeQuery(String input): Method yang digunakan untuk mengirim perintah ke *server* dengan mengimplementasikan TCP/IP.
- boolean next(): Method yang digunakan mengecek apakah masih ada record berikutnya.
- String getString(String NamaField): Method yang digunakan untuk mengambil data berupa string.
- Integer getInteger(String NamaField): Method yang digunakan untuk mengambil data berupa integer.
- Double getDouble(String NamaField): Method yang digunakan untuk mengambil data berupa double.
- String getColumnName(int x): Method yang digunakan untuk mengambil nama kolom.

- String `getColumnType(int x)`: Method yang digunakan untuk mengambil tipe data kolom.
- Integer `getColumnCount()`: Method yang digunakan untuk mendapat jumlah kolom data.
- Integer `getColumnRow()`: Method yang digunakan untuk mendapat jumlah baris data.

Proses yang terjadi pada sisi *client*, dimulai dari menentukan ip address dan port *server* serta nama database yang ingin diakses dan membuat objek *socket* yang digunakan untuk dapat berkomunikasi kepada *server*. Membuat variabel – variabel penampung, seperti jumlah baris, jumlah kolom, nama kolom, tipe data, dan isi record data.

Dilanjutkan dengan menginput perintah sql dan perintah sql tersebut dapat dikirim ke *server* dengan menggunakan method `executeQuery` yang telah mengimplementasikan TCP/IP pada method tersebut. Method `executeQuery` menerima ip address dan port *server* serta nama database yang telah ditentukan serta membuat *socket* dari ip address *server* dan port yang telah ditentukan dan membuat input dan output stream untuk mengirim perintah ke *server* dan menerima data dari *server*. Setelah *client* telah terkoneksi *server*, *client* dapat mengirim nama database dan perintah sql ke *server* dengan implementasi TCP/IP.

Pada method `executeQuery` terdapat validasi perintah sql, apakah perintah tersebut sql pengambilan data, jika ya maka menerima dan menampung data serta metadata yang telah direspon oleh *server* ke variabel yang sesuai seperti jumlah baris, jumlah kolom, nama kolom, tipe data, dan isi record data, jika tidak maka *client* hanya menerima pesan berhasil atau gagal saat eksekusi sql manipulasi data yang direspon dari *server*. Data yang telah ditampung dapat diakses dengan menggunakan method yang tersedia seperti method `next`, `getString`, `getInteger`, `getDouble`, `getColumnName`, `getColumnType`, `getColumnCount`, `getRowCount`.

Method `next` digunakan untuk mengecek apakah masih ada record berikutnya. Dimulai dari membuat objek validasi untuk mengembalikan nilai true atau

false. Dilanjutkan nilai pada variabel posisi Field ditambah dengan 1 yang digunakan untuk berpindah posisi field pada record.

Selanjutnya memvalidasi apakah masih ada record setelah berpindah posisi. Jika ya, maka mengembalikan nilai validasi true dan jika tidak, maka mengembalikan nilai validasi false serta membuat nilai `posisiField` menjadi 0.

Method `getColumnName` yang digunakan untuk mendapatkan nama kolom. Dimana method `getColumnName` menerima inputan posisi kolom untuk mendapatkan nama kolom sesuai posisi kolom tersebut dan mengembalikan nama kolom sesuai posisi kolom yang telah diinputkan.

method `getColumnType` yang digunakan untuk mendapatkan tipe data kolom. Dimana method `getColumnType` menerima inputan posisi kolom untuk mendapatkan jenis tipe data kolom sesuai posisi kolom tersebut dan mengembalikan jenis tipe data sesuai posisi kolom yang telah diinputkan.

Method `getColumnCount`, dapat mengembalikan berupa jumlah kolom data. Method `getRowCount`, dimana method tersebut dapat mengembalikan berupa jumlah baris data.

## HASIL DAN PEMBAHASAN

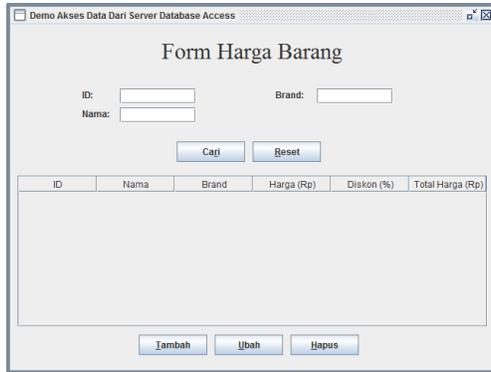
Pada implementasi TCP/IP akan ditampilkan setiap tampilan dasar dari aplikasi yang telah dibuat.

### Tampilan Aplikasi *Client*

Pada Gambar 4.1 menampilkan sebuah tampilan dasar pada aplikasi *client* yang menggunakan class *client* yang telah dibuat untuk mengakses data dari *server* dengan mengimplementasikan TCP/IP. Pada Gambar 1 terdapat textfield dan button.

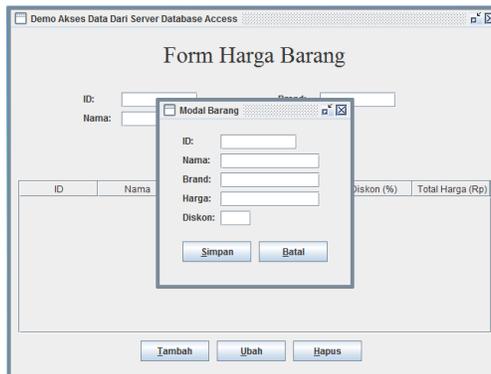
Pada Gambar 5 yang merupakan tampilan aplikasi *client* yang menggunakan class *client*, terdapat textfield pencarian data barang dimana user dapat mencari berdasarkan id, nama, dan brand. Button cari digunakan untuk mengesekusi query pada *server* database access berdasarkan pencarian yang telah diinput oleh user pada textfield id, nama, maupun brand dan menampilkan data yang telah dieksekusi pada tabel yang telah disediakan. Jika textfield id, nama, maupun brand tidak diisi

maka secara dasar akan menampilkan semua data yang tersedia pada tabel.



**Gambar 5 Tampilan Aplikasi Client Yang Menggunakan Class Client**

Tombol reset digunakan untuk menghapus semua inputan user pada texfield id, nama dan brand serta membersihkan tampilan pada gridview sehingga tidak ada data yang ditampilkan pada tabel.



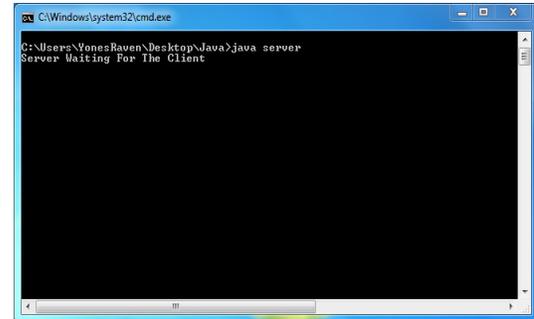
**Gambar 6 Tampilan Form Modal Barang**

Tombol tambah dan tombol ubah pada form harga barang akan menampilkan form modal barang yang dapat dilihat pada Gambar 6 yang digunakan untuk menambah atau mengubah data barang sesuai inputan user pada textfield yang tersedia pada form modal barang.

#### Tampilan Aplikasi Server

Pada gambar 7 menampilkan sebuah tampilan dasar pada aplikasi server yang akan menerima dan merespon permintaan client dengan mengimplementasikan TCP/IP, yang digunakan untuk menampilkan informasi alamat client

yang terkoneksi ke server dan perintah sql dari client.



**Gambar 7 Tampilan Aplikasi Server**

#### Pengujian Sistem

Pengujian sistem dilakukan terhadap aplikasi pada dua sisi, yaitu pada sisi server dan sisi client. Tabel 1 merupakan informasi pengujian yang telah dilakukan.

**Tabel 1. Hasil Pengujian**

Pengujian	Ya	Tidak	Berapa kali uji	Sukses	Gagal
Koneksi Aplikasi client dengan aplikasi server	✓		5	5	0
Aplikasi client mengirimkan perintah sql ke server	✓		5	5	0
Aplikasi server mengolah perintah sql pengambilan data dari client	✓		2	2	0
Aplikasi server mengirim data sesuai perintah sql dari client	✓		2	2	0
Aplikasi client menerima data dari aplikasi server	✓		2	2	0
data yang diterima aplikasi client dapat diakses	✓		2	2	0
Aplikasi server mengolah perintah sql manipulasi data dari client ke database access?	✓		3	3	0

Sasaran pengujian dilakukan terhadap tujuh fungsi dari sisi client dan sisi server, yaitu:

- Koneksi Aplikasi client dengan aplikasi server, pengujian dilakukan terhadap sisi client dan sisi server.
- Aplikasi client mengirimkan perintah sql ke server. pengujian dilakukan terhadap sisi client, dan berhasil mengirimkan perintah sql ke server.
- Aplikasi server mengolah perintah sql pengambilan data dari client. Pengujian dilakukan terhadap sisi server.

- Aplikasi *server* mengirim data sesuai perintah sql dari *client*. Pengujian dilakukan pada sisi *client* dengan memperhatikan data yang diterima sisi *client* apakah sesuai dengan perintah yang dikirim atau tidak..
- Aplikasi *client* menerima data dari aplikasi *server*. Pengujian dilakukan pada sisi *client* dengan memperhatikan data yang diterima sisi *client* apakah sesuai dengan perintah yang dikirim atau tidak.
- data yang diterima aplikasi *client* dapat diakses. Pengujian dilakukan pada sisi *client* dan dicoba dengan mengakses data set.
- Aplikasi *server* mengolah perintah sql manipulasi data dari *client* ke database access.

Semua pengujian berhasil dan tidak ada yang gagal.

## SIMPULAN

### Kesimpulan

Berdasarkan penelitian TCP/IP untuk membuat *server* database access, dapat disimpulkan:

- Aplikasi yang dibuat dengan memanfaatkan TCP/IP mampu membuat database access menjadi *server* database access.
- Dari hasil pengujian memberikan kesimpulan bahwa TCP/IP layak sebagai sarana pengiriman data ke aplikasi lain.

## Saran

Berdasarkan penelitian yang telah dilakukan, maka disarankan ;

- Dengan fitur dan aplikasi yang sama, dalam melakukan penelitian untuk TCP/IP dengan mengirim data, secara dasar data yang diterima berupa tipe data string sehingga dibutuhkan suatu method yang digunakan untuk mengkonversi tipe data yang diinginkan

## DAFTAR PUSTAKA

- [1] Hariyanto, Bambang (2010), *Esensi – Esensi Bahasa Pemrograman Java*, Informatika Bandung, Bandung.
- [2] Kurniawan, Agus (2011), *Pemrograman Jaringan dengan JAVA*, ANDI, Yogyakarta.
- [3] Kusriani (2007), *Strategi Perancangan dan Pengelolaan Basis Data*, ANDI, Yogyakarta.
- [4] Microsoft. 2014. *Cara Berbagi Database Desktop Access*.  
<http://office.microsoft.com/id-id/access-help/cara-berbagi-database-desktop-access-HA102749577.aspx>. 25 Oktober 2014(10:00).
- [5] Sadeli, Muhammad (2011), *7 Jam Belajar Interaktif Access 2010 untuk Orang Awam*, Maxikom, Palembang.
- [6] WartaWarga. 2012. *Model Database Client-Server*.  
<http://wartawarga.gunadarma.ac.id/2010/02/model-database-client-server/>. 26 Oktober 2014(14:00)..