

Aplikasi Transaksi Bengkel

PROYEK AKHIR 1

Oleh :

Eko Irawan / 3311211035

Khoiri Syarifudin / 3311211042

Rezha Pahlevi / 3311211048

Yanis Anwarani / 3311211068



PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2013

DAFTAR ISI

BAB I	Error! Bookmark not defined.
PENDAHULUAN	Error! Bookmark not defined.
1.1 Latar Belakang.....	Error! Bookmark not defined.
1.2 Rumusan Masalah.....	Error! Bookmark not defined.
1.3 Batasan Masalah	Error! Bookmark not defined.
1.4 Tujuan.....	Error! Bookmark not defined.
1.5 Sistematika Penulisan	Error! Bookmark not defined.
BAB II	Error! Bookmark not defined.
LANDASAN TEORI.....	Error! Bookmark not defined.
2.1 POINT OF SALES (POS)	Error! Bookmark not defined.
2.2 Java Language	Error! Bookmark not defined.
2.3 NetBeans IDE 7.0.2	Error! Bookmark not defined.
2.4 Database MySQL	Error! Bookmark not defined.
2.5 <i>iReport</i>	Error! Bookmark not defined.
BAB III	Error! Bookmark not defined.
ANALISIS DAN PERANCANGAN.....	Error! Bookmark not defined.
3.1 DISKRIPSI UMUM KEBUTUHAN	Error! Bookmark not defined.
3.2 Use Case Diagram	Error! Bookmark not defined.
3.3 Skenario Use Case	Error! Bookmark not defined.
3.4 Sequence Diagram.....	Error! Bookmark not defined.
BAB IV	Error! Bookmark not defined.
IMPLEMENTASI DAN HASIL	Error! Bookmark not defined.
4.1 Implementasi Program.....	26
4.2 Metode Implementasi	26
4.3 Proses implementasi	27

4.3.1	Fitur login	27
4.3.4	Fitur <i>Delete</i>	30
4.3.5	Fitur <i>Update</i>	31
4.3.6	Fitur cari data	32
BAB V		33
KESIMPULAN DAN SARAN.....		33
5.1	Kesimpulan.....	33
5.2	Saran	34
BAB VI		Error! Bookmark not defined.
DAFTAR PUSTAKA		34

DAFTAR GAMBAR

- Use Case Diagram

- Sequence Diagram

- * Sequence diagram login
- * Sequence diagram data barang
- * Sequence diagram data karyawan
- * Sequence diagram data customer
- * Sequence diagram data transaksi

DAFTAR TABEL

Skenario use case

BAB I

Pendahuluan

Latar Belakang

Melihat kondisi lapangan (objek tujuan) yang masih menggunakan metode manual dalam sistem transaksi maupun mengelola keuangan, stok barang, dll di era modern saat ini tentu tidak efektif. Akan memerlukan banyak waktu dan tenaga. Hal tersebut diatas selain dapat menimbulkan efek negatif misalnya kesalahan menghitung, lupa menentukan harga barang pada konsumen yang berdampak pada kerugian.

Untuk sedikit membantu agar pekerjaan mengelola bengkel menjadi lebih mudah, maka kami membuat aplikasi transaksi bengkel.

Rumusan Masalah

Rumusan masalah yang dapat diambil adalah :

- Bagaimana cara mengetahui dengan cepat daftar spare part bulan kemarin dengan cepat?
- Bagaimana cara mengetahui semua transaksi hari ini dengan cepat?
- Bagaimana cara mengetahui harga jual dan beli barang serta quantitynya secara bersamaan dengan cepat?
- Bagaimana cara mengetahui semua daftar customer dengan cepat?

Batasan Masalah

- Tidak bisa mengeprint data transaksi kemarin.
- Tidak bisa mencari data customer berdasarkan nomer teleponnya.
- Tidak bisa diakses secara on-line.
- Tidak bisa diakses di kompuer yang tidak terinstal *NetBeans*

Tujuan

- Untuk mengetahui semua daftar spare part dengan cepat.
- Untuk mengetahui semua data transaksi hari ini dengan cepat.
- Untuk mengetahui harga jual dan beli serta quantity barang secara bersamaan dengan cepat.
- Untuk mengetahui semua data customer dengan cepat.

Sistematika Penulisan

Bab I Pendahuluan

- Latar belakang, yang berisi tentang alasan mendasar pemilihan topik berupa paparan teoritis dan paparan praktis.
- Rumusan masalah, berisi tentang masalah yang dihadapi.
- Batasan masalah, berisi tentang kendala-kendala apa saja yang akan timbul setelah aplikasi ini selesai dibuat.
- Tujuan, berisi sasaran yang ingin dicapai dalam pembuatan proyek ini.

BAB II TINJAUAN PUSTAKA

Bab ini memuat tentang deskripsi singkat perangkat lunak.

BAB III ANALISIS DAN PERANCANGAN

Bab ini berisi tentang software yang terlibat dalam pembuatan aplikasi serta rancangan tentang aplikasi yang akan dibuat.

BAB IV IMPLEMENTASI

Bab ini berisi tentang Implementasi dan Pengujian

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang didapat dari hasil laporan yang dikerjakan serta saran dari pengembangan aplikasi yang dibuat.

BAB II

Tinjauan Pustaka

1.1 Point Of Sales

Perkembangan dunia komputer semakin lama semakin berkembang dengan pesat, sehingga mendorong setiap individu maupun kelompok harus menerapkannya dalam segala bidang. Perkembangan tersebut dapat menimbulkan hal yang positif dan juga hal yang negatif (Darwin Sitompul, 1994).

Dampak-dampak positif dari perkembangan dunia komputer salah satunya adalah dengan ditemukannya/dimodifikasinya mesin *Cash Register* menjadi mesin POS (*Point of sales*) yang digunakan dalam bidang penjualan, khususnya *retail* toko dan sebagainya. Yang terbukti mempermudah proses transaksi penjualan.

Menurut para ahli *Point of sales* atau PoS memiliki banyak pengertian diantaranya adalah :

- Secara umum *Point of Sales* adalah sebuah sistem yang memungkinkan diadakannya proses transaksi.
- Secara spesifik *Point of sales* adalah *hardware* atau *software* yang digunakan untuk transaksi. Yang dilengkapi dengan sistem pelaporan manajemen yang terintegrasi dan biasanya digunakan di restoran, toko, supermarket dan tempat-tempat lain.

Point Of Sales adalah sistem kasir, yaitu aktivitas yang berorientasi pada penjualan yang terjadi pada bidang usaha retail.

Dari banyak pengertian tersebut dapat disimpulkan bahwa *Point of sales* adalah Sebuah aplikasi/program komputer yang terdapat pada mesin kasir digunakan untuk proses transaksi yang telah dilengkapi dengan sistem

pelaporan manajemen yang terintegrasi dan biasanya berorientasi pada penjualan.

Ada 2 komponen penting dalam sistem *Point of sales* yaitu :

POS software adalah program yang membantu fungsi seluruh sistem dan bertanggung jawab untuk mengirimkan dan menerima informasi atau data dalam sistem.

POS hardware adalah layar *display* pelanggan, laci kas, perangkat menggesekkan (untuk kartu kredit), printer, komputer dan pembaca *barcode*.

2.2 NetBeans IDE 7.03

NetBeans adalah *platform* pengembangan kerangka kerja dan *open source IDE* (Integrated Development Environment) untuk *Java*, *JavaScript*, *PHP*, dll. Yang dikembangkan oleh *Oracle Corporation* dan bebas biaya.

IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun *Graphic User Interface* (GUI), suatu *text* atau kode editor, suatu *compiler* atau *interpreter* dan suatu *debugger*.

NetBeans IDE digunakan untuk menulis, mengompilasi, mencari kesalahan dan menyebarkan program. *Netbeans IDE* ditulis dalam *Java*, namun dapat mendukung bahasa pemrograman lain. Selain itu *NetBeans IDE* merupakan produk bebas dan tanpa batasan apapun.

NetBeans IDE sangat cocok jika digunakan dalam proyek ini karena ditulis dalam bahasa *java*, bersifat bebas serta berbasis GUI. Sehingga mempermudah proses penelitian dan pembuatan proyek ini.

Fitur-fitur yang terdapat dalam netbeans antara lain :

- *Smart Code Completion*: untuk mengusulkan nama *variabel* dari suatu tipe, melengkapi *keyword* dan mengusulkan tipe parameter dari sebuah *method*.
- *Bookmarking*: fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
- *Go to commands*: fitur yang digunakan untuk *jump* ke deklarasi variabel, *source code* atau file yang ada pada *project* yang sama.
- *Code generator*: jika kita menggunakan fitur ini kita dapat meng-*generate* *constructor*, *setter* and *getter method* dan yang lainnya.
- *Error stripe*: fitur yang akan menandai baris yang *error* dengan memberi *highlight* merah.

Netbeans IDE dalam pembuatan suatu program juga di dukung oleh beberapa jenis *Database*, antara lain adalah :

- *JDBC* merupakan spesifikasi standar dari *JavaSoft API (Application Programming Interface)* yang memungkinkan program *Java* untuk mengakses sistem database manajemen. *JDBC API* terdiri dari satu set *interface* dan kelas yang ditulis dalam bahasa pemrograman *Java*. *JDBC API* menggunakan *interface* standar dan kelas, *programmer* dapat menulis aplikasi yang terhubung ke database, mengirimkan pertanyaan ditulis *SQL (Structured Query Language)*, dan memproses hasilnya.
- *ORACLE* merupakan sebuah aplikasi basisdata yang didukung oleh *NetBeans*, *Oracle* secara umum hampir sama dengan *MySQL* namun yang membedakan adalah *oracle* dapat digunakan dan dihubungkan dengan *NetBeans* dan harus menggunakan *drivers* untuk menyimpan data-data yang telah dibuat. *Oracle* merupakan salah satu dari beberapa aplikasi basisdata yang sering digunakan untuk koneksi ke basisdata pada *NetBeans* karena relatif mudah dan cepat.
- *MySQL* merupakan sebuah aplikasi basisdata yang didukung oleh *NetBeans*, *MySQL* memiliki Fitur Yang ada sudah lumayan lengkap, dari input, update, delete serta Search. Sebuah antarmuka *ODBC* memanggil *MyODBC* yang memungkinkan setiap bahasa pemrograman yang

mendukung ODBC untuk berkomunikasi dengan basis data MySQL. Kebanyakan kode sumber MySQL dalam ANSI C.

- **DataBase Access** (*Microsoft Access*) merupakan sebuah aplikasi basis data yang didukung oleh *NetBeans*, data dapat disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database*, atau semua *kontainer* basis data yang mendukung standar ODBC. *Access* juga mendukung teknik-teknik pemrograman berorientasi objek.

2.3. Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana Anda membuatnya dengan bahasa seperti Pascal atau C++. Yang lebih menarik, Java juga mendukung sumber daya internet yang saat ini populer, yaitu World Wide Web atau yang sering disebut Web saja. Java juga mendukung aplikasi client/server, baik dalam jaringan lokal (LAN) maupun jaringan berskala luas (WAN).

- **Edisi Java**

Java adalah bahasa yang dapat dijalankan di sembarang platform, di beragam lingkungan, misalnya internet, consumer electronic product, dan computer applications. Menurut Hariyanto (2010 : 3), The Java 2 Platform tersedia dalam tiga

edisi untuk keperluan berbeda sebagai berikut :

1. Java 2 Standard Edition (J2SE)

The Java 2 Platform, Standard Edition (J2SE) menyediakan lingkungan

pengembangan yang kaya fitur, stabil, aman, dan cross-platform. Edisi ini mendukung konektivitas basis data, rancangan antarmuka pemakai, masukan/keluaran, dan pemrograman jaringan dan termasuk sebagai paket-paket dasar bahasa Java.

2. Java 2 Enterprise Edition (J2EE)

The Java 2, Enterprise Edition (J2EE) menyediakan kakas untuk membangun dan menjalankan multitier enterprise applications. J2EE berisi paket-paket di J2SE ditambah paket-paket untuk mendukung pengembangan Enterprise JavaBeans, Java Servlets, Java Server Page, XML, dan kendali transaksi yang fleksibel.

3. Java 2 Micro Edition (J2ME)

The Java 2, Micro Edition (J2ME) untuk beragam consumer electronic product, seperti pager, smart card, cell phone, handheld PDA, dan set-top box. J2ME sembari menyediakan bahasa Java yang sama, unggul dalam portabilitas –kemampuan dijalankan di mana pun dan safe network delivery seperti J2SE dan J2EE. J2ME menggunakan sekumpulan paket lebih kecil. J2ME berisi subset paket J2SE ditambah paket spesifik Micro Edition javax.microedition.io. Aplikasi J2ME dapat diskala agar juga dapat bekerja dengan J2SE dan J2EE.

- **Fitur Bahasa Java**

Menurut Hariyanto (2010 : 9), fitur penting bahasa java adalah bahasa ditujukan untuk membuat beragam jenis aplikasi secara seragam, yaitu :

1. Program di lingkungan web browser

Yaitu Applet, program ini dieksekusi di web browser dari halaman web yang memuat Java applet. Web browser kemudian menugaskan Java interpreter (JRE – Java Runtime Environment) untuk mengeksekusi Java applet yang diterima. Java applet membuat langkah besar, memungkinkan web menjadi sarana/media interaktif di mana halaman web menjadi dapat bereaksi terhadap masukan atau pemakai.

- **Program di lingkungan web server**

a. Java Server Pages, sebagai web scripting serupa ASP, PHP dan sebagainya. Program ditempelkan di halaman html. Html tidak langsung dikirim ke web browser tapi diolah dulu oleh web server dan hasilnya berupa dokumen html dikirim web server ke web browser. Pemrograman JSP mengakomodasi pemrograman gaya tempelan seperti PHP, gaya tagging seperti Cold Fusion dan orientasi komponen JavaBeans.

b. Java Servlet, komponen ini adalah semacam modul di web server. JSP akan diterjemahkan menjadi servlet agar mempercepat proses eksekusi.

3. Program mandiri (atau bisa disebut stand-alone application). Java merupakan pilihan bagus untuk membuat applet, sekaligus sebagai bahasa bermaksud

umum (general-purpose language) untuk mengembangkan semua jenis program yang dapat dijalankan di computer, sistem operasi apa pun asalkan terdapat Java interpreter di platform itu. Untungnya hampir seluruh platform menyediakan Java interpreter karena spesifikasi Java bytecode disebarluaskan secara terbuka. Filosofi program Java dapat dijalankan di platform computer dan sistem operasi apapun dikenal sebagai “WORA – write once, run anywhere”.

4. Program mandiri sebagai pustaka komponen untuk pengembangan aplikasi.

Java menyediakan Bean untuk mendukung RAD (rapid application development) yang berbasis visual seperti Visual Basic atau Borland Delphi.

5. Bahasa untuk pengembangan aplikasi objek-objek tersebar skala enterprise.

Terdapat teknologi Java untuk mengembangkan komponen yaitu EJB (Enterprise JavaBeans) yang berjalan di application server. Application server adalah middleware yang bertugas menjadi intermediasi beragam server di aplikasi tersebar berskala enterprise.

• 2.3.3. Kelebihan dan Kekurangan Java

Kelebihan Java antara lain :

1. Multiplatform. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip write once, run everywhere. Dengan kelebihan ini pemrogram cukup menulis sebuah program

java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan di atas sistem operasi Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris.

2. OOP, singkatan dari Object Oriented Programming yang artinya semua aspek yang terdapat di Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis object murni. Semua tipe data diturunkan dari kelas dasar Object. Hal ini sangat memudahkan programmer untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan java sebagai salah satu bahasa pemrograman termudah, bahkan untuk fungsi-fungsi yang advance seperti komunikasi antara komputer sekalipun.

3. Library kelas sangat lengkap. Java terkenal dengan kelengkapan library (kumpulan program-program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para programmer untuk membangun aplikasinya. Kelengkapan library ini ditambah dengan

keberadaan komunitas Java yang besar yang terus menerus membuat librarylibrary

baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

4. Garbage Collection, memiliki fasilitas pengaturan penggunaan memori sehingga programmer tidak perlu melakukan pengaturan memori secara langsung.

Sedangkan kekurangan dari Java antara lain :

1. Write once, debug everywhere. Ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari executable code menjadi source code. Ini dimungkinkan karena executable Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi seperti nama-nama kelas, method dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan.

3. Heavy memory usage. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena

trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin desktop berumur lebih dari 4 tahun.

1.4 Database MySQL

MySQL adalah suatu database server yang sangat terkenal di dunia dan merupakan Open Source SQL database (database SQL yang open source). MySQL

mendukung berbagai platform, serta kecepatan aksesnya cukup bisa diandalkan.

Selain itu, kemudahannya dalam integrasi ke berbagai aplikasi juga membantu dalam

mengembangkan sistem aplikasi. Pemakai-pun tidak harus membayar untuk bisa menggunakannya, karena paket program ini dapat digunakan secara bebas.

Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas.

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga dan masih banyak lagi.

2. Perangkat lunak *open source*.

MySQL didistribusikan sebagai perangkat lunak *Open Source*, di bawah lisensi GPL (*General Public License*) sehingga dapat digunakan secara gratis.

3. Multi-user.

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance tuning.

MySQL memiliki kecepatan yang menakjubkan dalam menangani query

sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data.

MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan fungsi.

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).

7. Keamanan.

MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan.

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*record*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas.

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix socket* (UNIX), atau *Named Pipes* (NT).

10. Lokalisasi.

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar muka.

MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. Klien dan peralatan.

MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan

petunjuk online.

13. Struktur tabel.

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.

BAB III

Analisis dan Perancangan

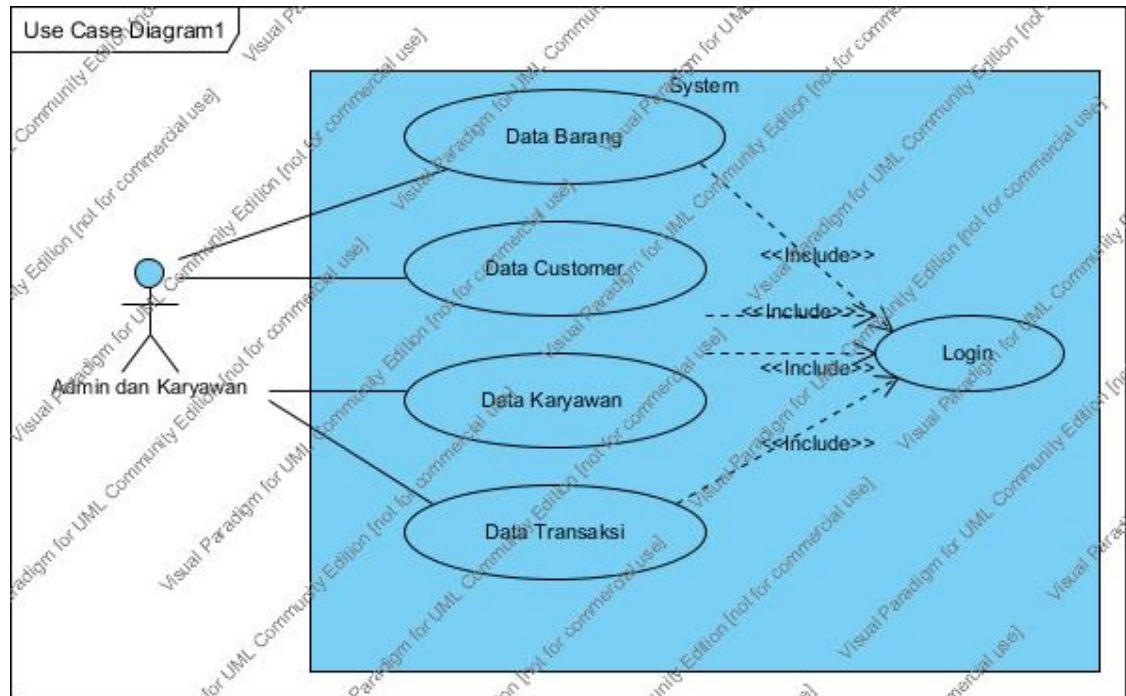
III.1 Diskripsi Umum Sistem

Bagian ini berisi analisis dan perancangan aplikasi yang kami buat.

Aplikasi kami menggunakan software XAMPP, NetBeans, dan iReport.

Aplikasi ini mempunyai 2 user yaitu admin dan karyawan. Kedua user tersebut dapat menjalankan semua aplikasi yang ada.

III.2 Use Case Diagram



Gambar III.2. 1Gambar *Use Case Diagram*

III.2.1 Skenario Use Case

III.2.1.1 Skenario Use Case Login

Use Case Name : *Login*

Actor : *Owner dan Karyawan*

Pre –Condition : *Actor* ingin menggunakan aplikasi dan belum melakukan *login*.

Post –Condition : *Actor* telah melakukan *login* dan berinteraksi dengan aplikasi.

Description : *Actor* melakukan *login* agar dapat menggunakan aplikasi.

Tabel III.2. 1 Skenario proses *login*

<i>Owner dan karyawan</i>	<i>System</i>
Menginputkan <i>user Id</i> dan <i>user password</i> .	
Mengklik tombol “ <i>Login</i> ”	eksekusi validasi <i>password</i> dengan

	yang tersimpan dalam <i>database user</i>
	Jika tidak sesuai, tampilkan <i>message box login gagal</i>
	Jika sesuai, tampilkan <i>message box login berhasil</i> , lanjut ke menu aplikasi
	Tampilkan menu utama aplikasi.
Melakukan aktivitas dalam aplikasi tersebut.	

III.2.1.2 Skenario Data Barang

Use Case Name : *Data Barang*

Actor : *Owner dan Karyawan*

Pre –Condition : *Actor* ingin menginput atau mengupdate atau delete atau mencari barang setelah melakukan login.

Post –Condition : *Actor* telah melakukan input atau update atau delete atau mencari barang.

Description : *Actor* melakukan input barang untuk menambah jumlah atau jenis barang, *actor* melakukan update barang untuk mengubah status barang yang telah ada, *actor* melakukan delete barang untuk menghapus barang, *actor* melakukan cari barang untuk mencari barang yang diinginkan.

Tabel III.2.1. 2 Skenario *use case Data Barang*

<i>Owner</i>	<i>System</i>
Memilih menu <i>Data Barang</i> pada menu utama.	Membuka <i>Form Data Barang</i> .

	Menampilkan isi di form data barang.
Menginput atau mengupdate atau delete atau mencari barang	
	Mengubah atau menampilkan status barang terbaru

III.2.1.3 Skenario Data Customer

<i>Use Case Name</i>	: <i>Data Customer</i>
<i>Actor</i>	: <i>Owner dan Karyawan</i>
<i>Pre –Condition</i>	: <i>Actor</i> ingin menginput atau mengupdate atau delete atau mencari customer setelah melakukan login.
<i>Post –Condition</i>	: <i>Actor</i> telah melakukan input atau update atau delete atau mencari customer.
<i>Description</i>	: <i>Actor</i> melakukan input customer untuk menambah jumlah customer, <i>actor</i> melakukan update customer untuk mengubah status customer yang telah ada, <i>actor</i> melakukan delete customer untuk menghapus data customer, <i>actor</i> melakukan cari customer untuk mencari data customer yang diinginkan.

Tabel III.2.1. 3 Skenario *use case Data Customer*

<i>Owner</i>	<i>System</i>
Memilih menu <i>Data Customer</i> pada menu utama.	Membuka <i>Form Data Customer</i> .
	Menampilkan isi di form data customer.
Menginput atau mengupdate atau	

delete atau mencari customer.	
	Mengubah atau menampilkan status customer terbaru

III.2.1.4 Skenario Data Karyawan

<i>Use Case Name</i>	: <i>Data Karyawan</i>
<i>Actor</i>	: <i>Owner dan Karyawan</i>
<i>Pre –Condition</i>	: <i>Actor</i> ingin menginput atau mengupdate atau delete atau mencari karyawan setelah melakukan login.
<i>Post –Condition</i>	: <i>Actor</i> telah melakukan input atau update atau delete atau mencari karyawan.
<i>Description</i>	: <i>Actor</i> melakukan input karyawan untuk menambah jumlah karyawan, <i>actor</i> melakukan update karyawan untuk mengubah status karyawan yang telah ada, <i>actor</i> melakukan delete karyawan untuk menghapus data karyawan, <i>actor</i> melakukan cari karyawan untuk mencari data karyawan yang diinginkan.

Tabel III.2.1.4 Skenario *use case* Data Karyawan

<i>Owner</i>	<i>System</i>
Memilih menu <i>Data Karyawan</i> pada menu utama.	Membuka <i>Form Data Karyawan</i> .
	Menampilkan isi di form data karyawan.
Menginput atau mengupdate atau delete atau mencari karyawan.	
	Mengubah atau menampilkan status karyawan terbaru.

III.2.1.5 Skenario Data Transaksi

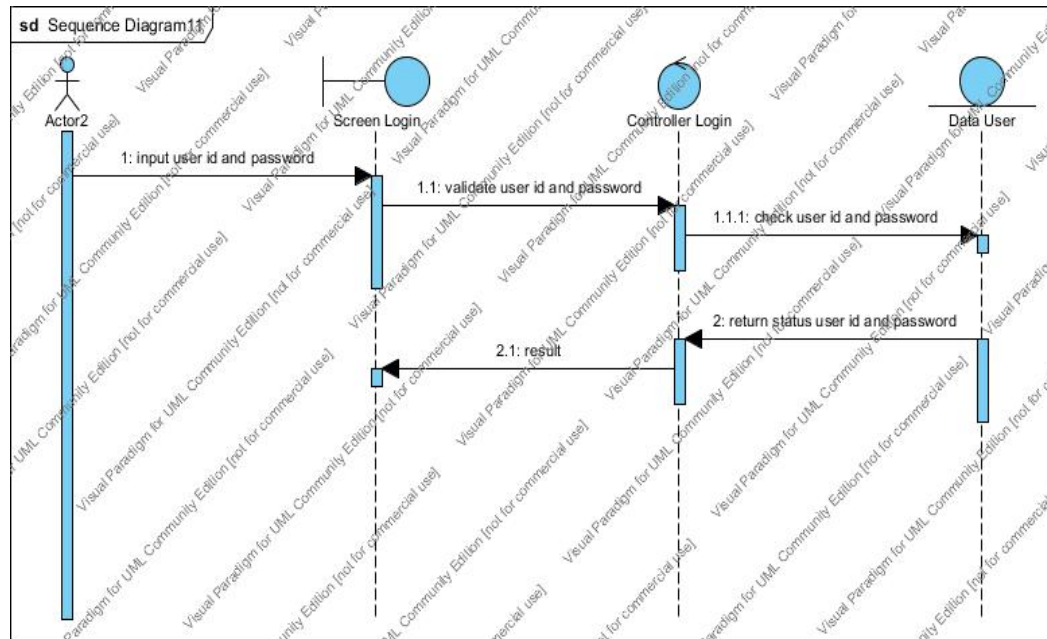
<i>Use Case Name</i>	: <i>Data Transaksi</i>
<i>Actor</i>	: <i>Owner dan Karyawan</i>
<i>Pre –Condition</i>	: <i>Actor</i> ingin menginput atau mengupdate atau delete atau mencari transaksi setelah melakukan login.
<i>Post –Condition</i>	: <i>Actor</i> telah melakukan input atau update atau delete atau mencari transaksi.
<i>Description</i>	: <i>Actor</i> melakukan input transaksi untuk menambah jumlah transaksi, <i>actor</i> melakukan update transaksi untuk mengubah status transaksi yang telah ada, <i>actor</i> melakukan delete transaksi untuk menghapus data transaksi, <i>actor</i> melakukan cari transaksi untuk mencari data transaksi yang diinginkan.

Tabel III.2.1.5 Skenario *use case* Data Transaksi

<i>Owner</i>	<i>System</i>
Memilih menu <i>Data Transaksi</i> pada menu utama.	Membuka <i>Form Data Transaksi</i> .
	Menampilkan isi di form data transaksi.
Menginput atau mengupdate atau delete atau mencari transaksi.	
	Mengubah atau menampilkan status transaksi terbaru.

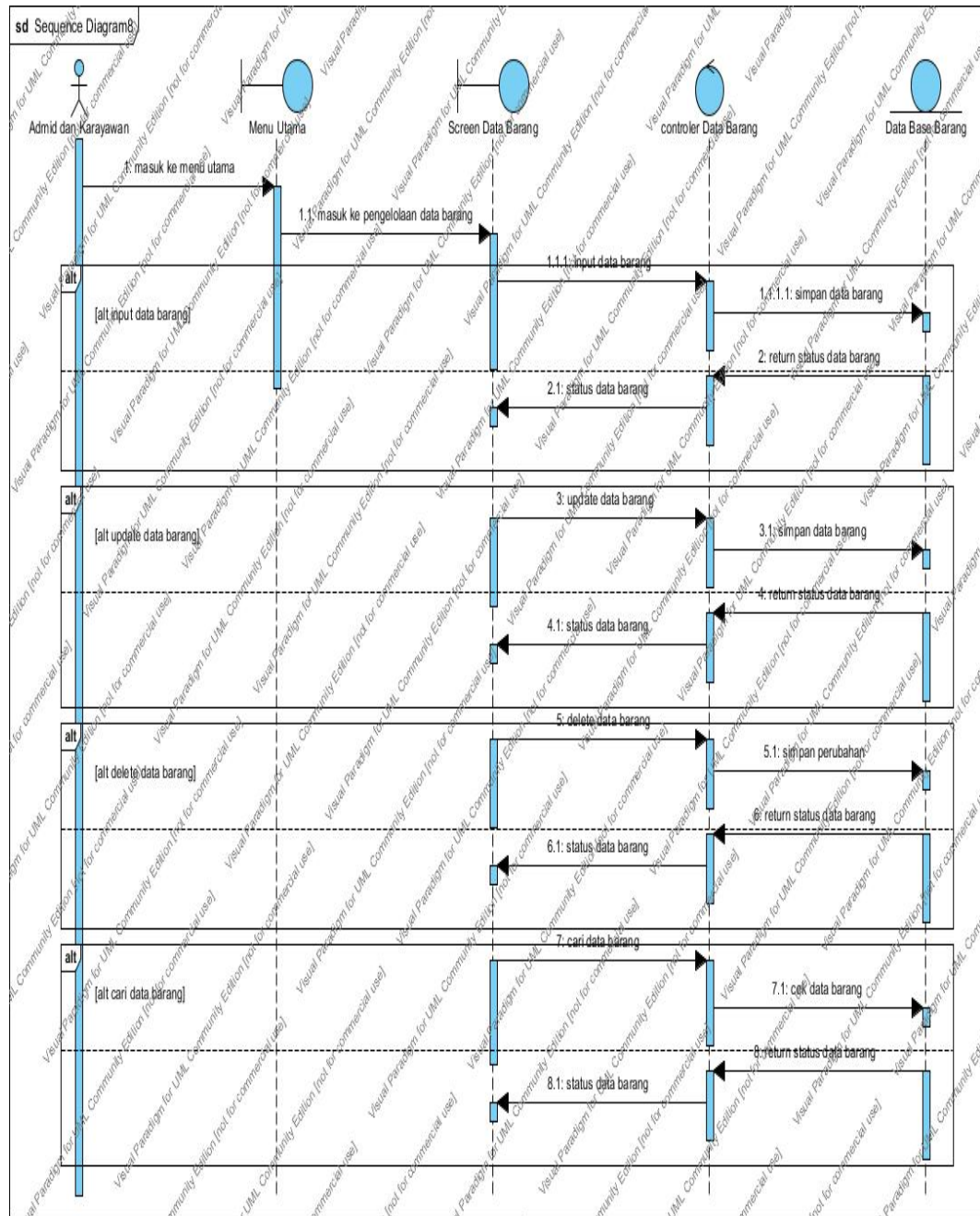
III.3 Sequence Diagram

III.3.1 Sequence Diagram Login



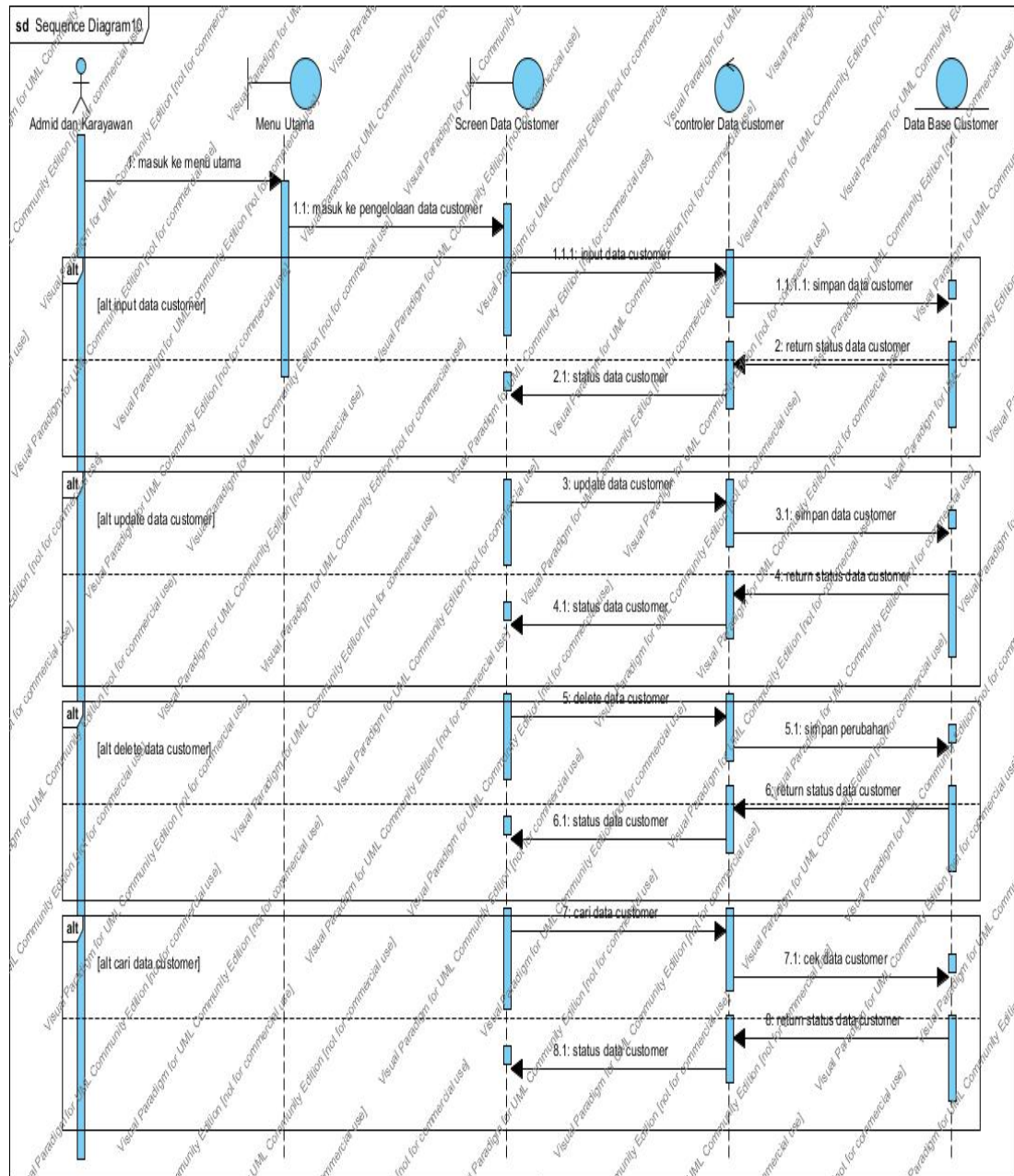
Gambar III.3.1 Sequence Diagram Login

III.3.2 Sequence Diagram Data Barang



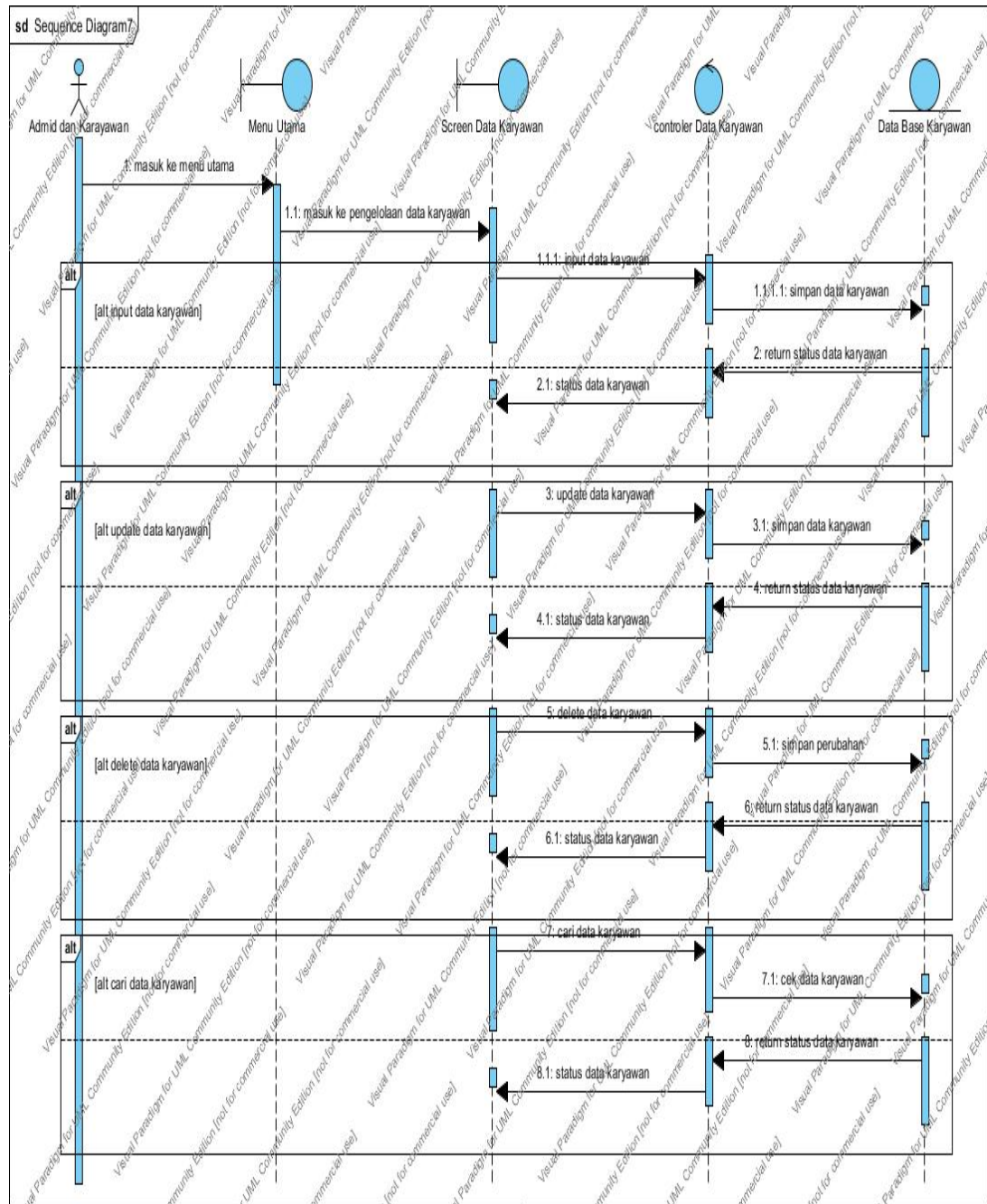
Gambar III.3.2 Sequence Diagram Data Barang

III.3.3 Sequence Diagram Data Customer



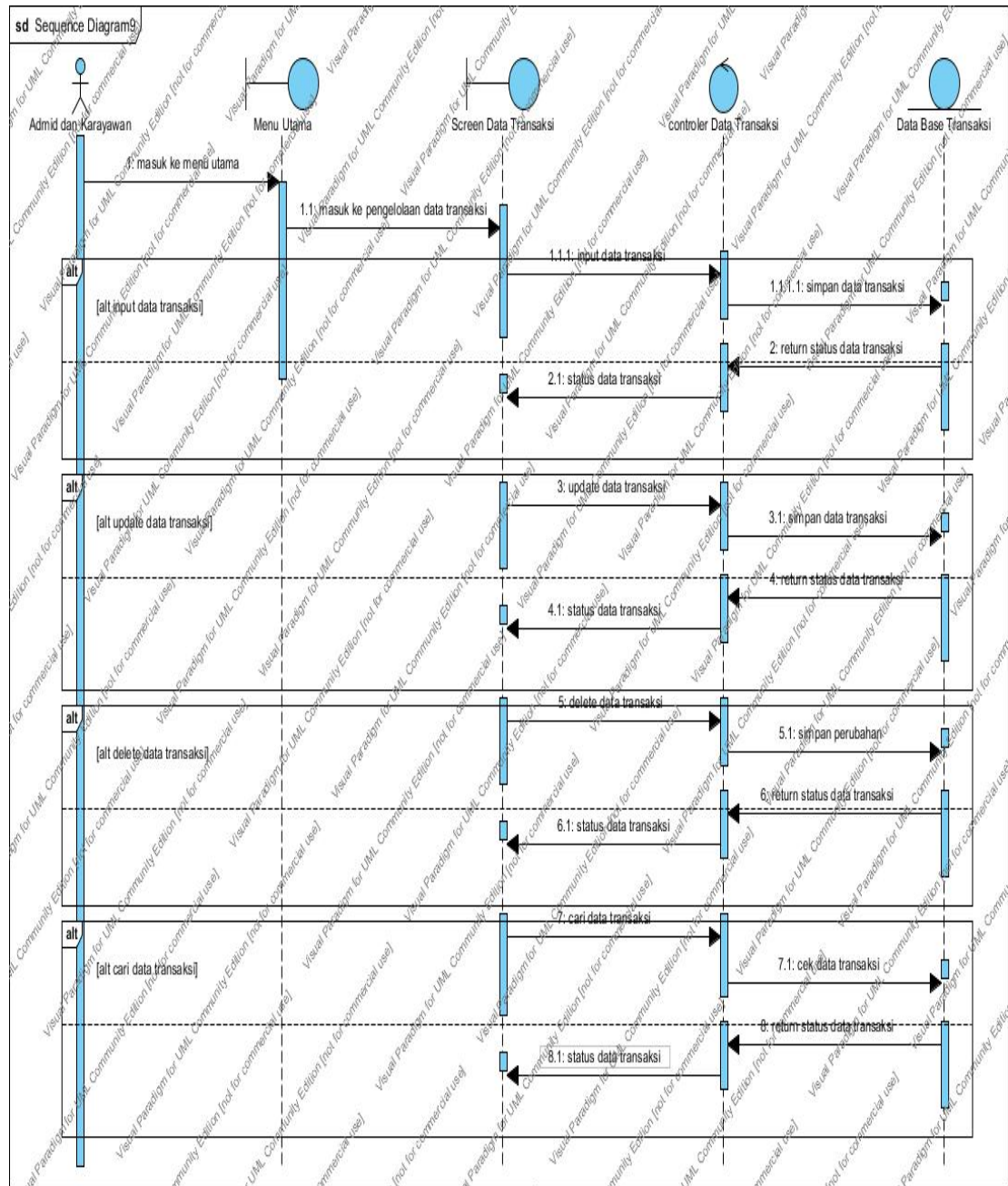
Gambar III.3.3 Sequence Diagram Data Customer

III.3.4 Sequence Diagram Data Karyawan



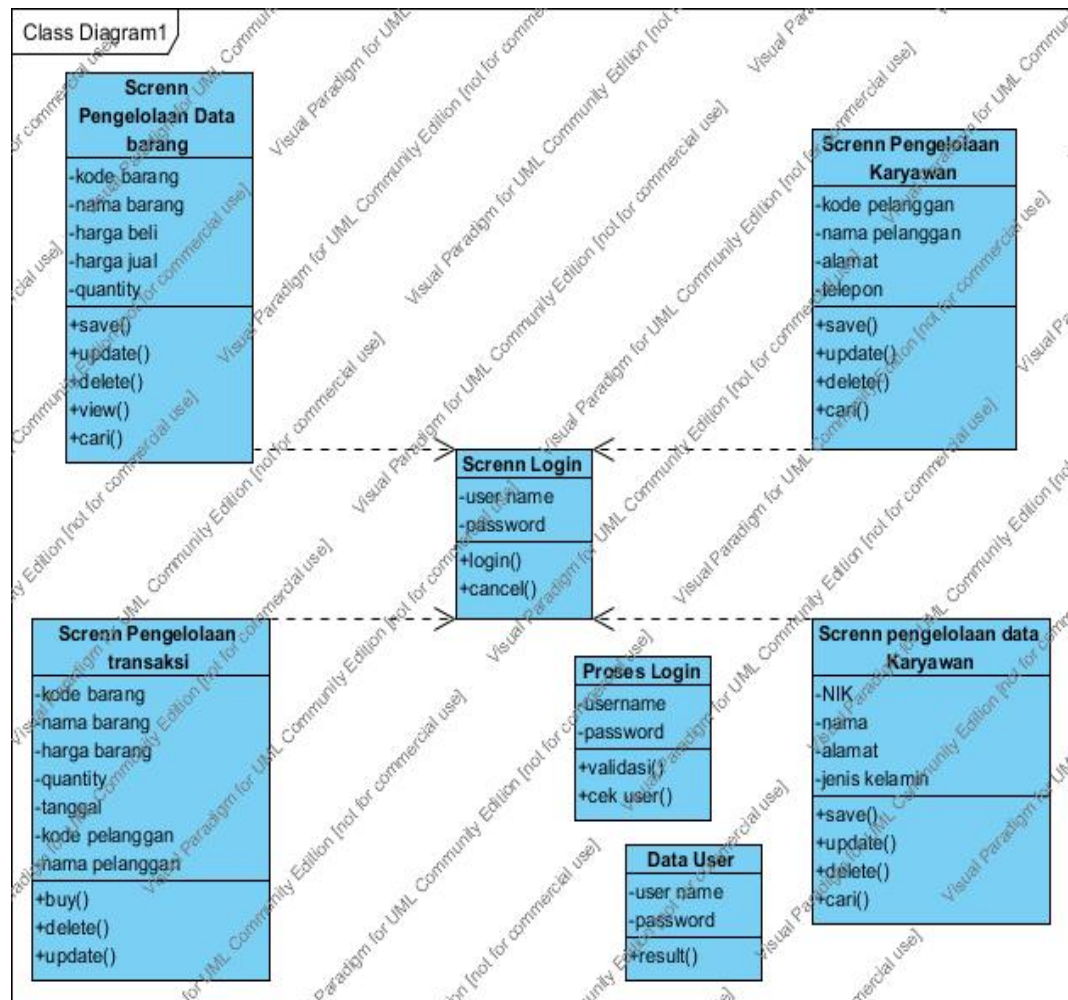
Gambar III.3.4 Sequence Diagram Data Karyawan

III.3.5 Sequence Diagram Data Transaksi



Gambar III.3.5 Sequence Diagram Data Transaksi

III.4 Class Diagram



Gambar III.4 Class Diagram

BAB IV Pembahasan

Bagian ini memuat uraian langkah implementasi dan pengujian/validasi. Dalam bab ini kami akan memaparkan aplikasi yang telah kami buat.

IV.1 Implementasi Program

Tujuan dilakukannya pengujian dan implementasi ini adalah untuk memastikan apakah fitur-fitur yang ada dalam aplikasi ini dapat berjalan sesuai fungsinya atau tidak. Implementasi ini akan dilakukan pada fitur-fitur yang tersedia pada aplikasi ini. Adapun fitur-fitur tersebut meliputi *update*, *view*, *insert*, *delete* yang terdapat pada menu Data Barang dan Transaksi Penjualan. Selain itu juga tersedia fitur *Login* yang dilakukan sebelum kasir berinteraksi dengan aplikasi.

IV.2 Metode Implementasi

Metode yang digunakan dalam proses implementasi ini adalah dengan menggunakan metode “**Black Box Testing**”. Metode *Black Box Testing* adalah suatu metode pengujian yang mengutamakan pengujian terhadap kebutuhan fungsi dari suatu program. Tujuannya dari metode *Black Box* adalah untuk mencari kesalahan fungsi.

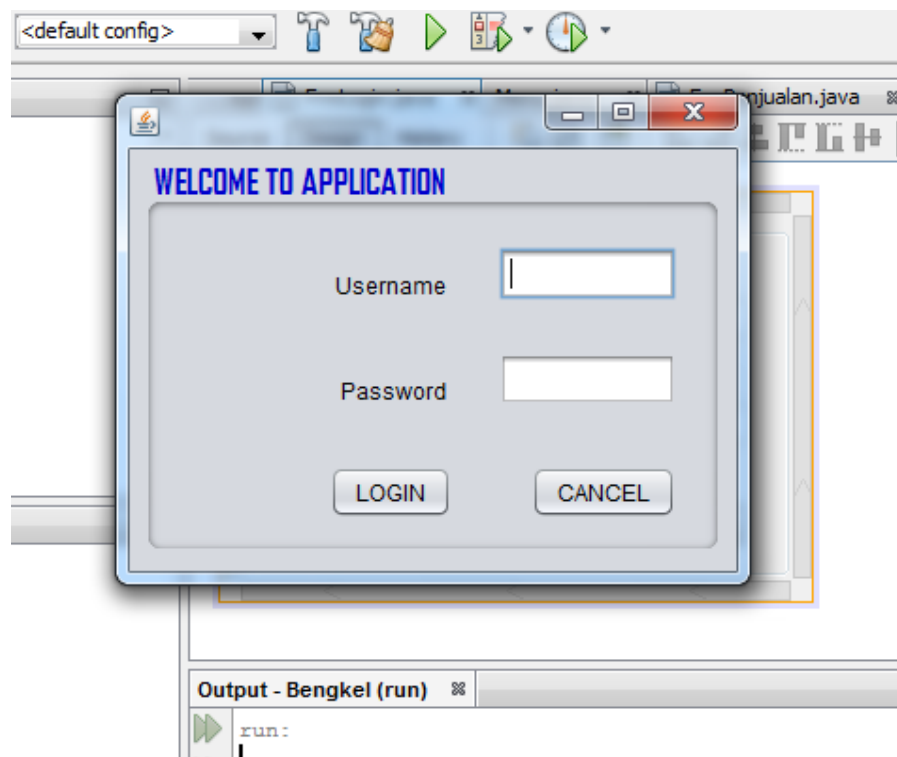
Pengujian dengan menggunakan metode *Black Box Testing* ini dilakukan dengan cara memberikan sejumlah *input* pada program aplikasi. *Input* tersebut akan di proses sesuai dengan kebutuhan fungsionalnya, untuk melihat apakah program aplikasi dapat menghasilkan *output* yang sesuai dengan yang diinginkan dan sesuai dengan fungsi dasar aplikasi tersebut. Jika dari *input* yang dimasukkan menghasilkan *output* yang sesuai dengan yang kebutuhan fungsionalnya maka program aplikasi tersebut sudah benar. Tetapi apabila *output* yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya maka program tersebut masih terdapat kesalahan. Selanjutnya akan dilakukan penelusuran kesalahan dan selanjutnya akan dilakukan penelusuran perbaikan (*Debugging*).

IV.3 Proses implementasi

Adapun proses implementasi akan dilakukan pada fitur-fitur yang ada pada aplikasi ini. Adapun fitur-fitur tersebut adalah *Login*, *main form*, *update*, *reset*, *delete*, *insert* dan *cari*. Berikut ilustrasi dan *screenshot* dari proses implementasi terhadap fitur-fitur tersebut.

IV.3.1 Fitur login

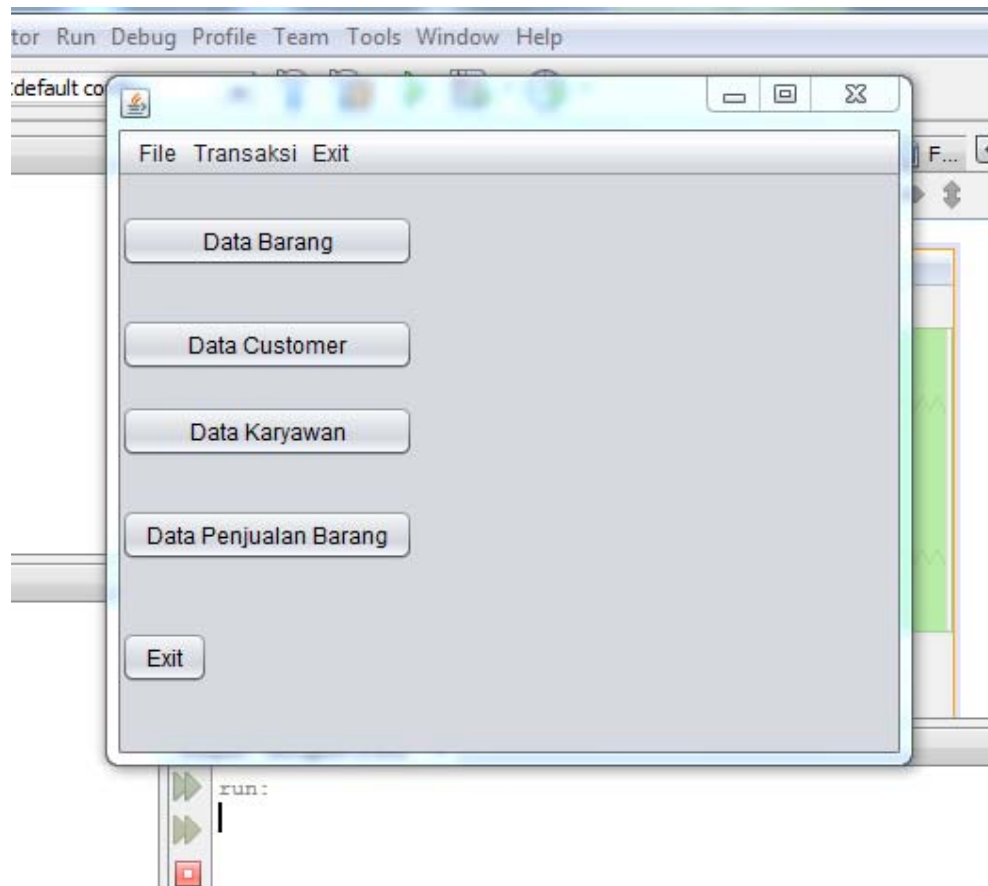
Fitur ini berfungsi untuk mengamankan data dari orang-orang yang tidak berkepentingan. Proses imlementasi pada fitur ini dilakukan dengan memasukkan *user name* dan *user ID*. Dapat dilihat pada gambar IV.3.1 tampilan *form* sebelum memasukkan *user name* dan *user ID*.



Gambar IV.3. 1 Gambar fitur Login dalam keadaan kosong

IV.3.2 Tampilan *Menu Utama*

Tampilan *Menu Utama* di berikan untuk mengantarkan pengguna ke fitur-fitur yang lainnya. Demikianlah tampilan Main form pada gambar



Gambar IV.3. 2 Gambar Menu Utama Aplikasi Penjualan Bengel

Menu Utama mempunyai menu *File* yang berfungsi membuka fitur-fitur yang lainnya, *Data Barang* berfungsi untuk membuka *Form Data Barang*, *Data Transaksi* untuk membuka *Form Data Transaksi*. *Data karyawan* untuk membuka *Form Karyawan*. *Data Penjualan Barang* untuk membuka *Form Transaksi*.

IV.3.3 Fitur Insert

Fitur ini berfungsi untuk memasukan data pada *database*. Proses implementasi pada fitur ini dilakukan dengan memasukkan *input* tertentu.

The screenshot shows a window titled "Entry Data Barang". On the left, there are five text input fields labeled "Kode Barang", "Nama Barang", "Harga Beli", "Harga Jual", and "Quantity", all of which are currently empty. To the right of these fields is a table with five columns: "Kode Bar...", "Nama Ba...", "Harga Beli", "Harga Jual", and "Quantity". The table contains four rows of data:

Kode Bar...	Nama Ba...	Harga Beli	Harga Jual	Quantity
B001	Oli	50000	60000	5
B002	Ban	50000	50000	5
B003	Aki	40000	50000	5
B004	Busi	20000	30000	5

Below the table are five buttons: "Save", "Delete", "Update", "View", and "Menu". At the bottom right, there is a search section with the text "Cari Berdasarkan Nama Barang" above a text input field and a "Search" button.

Gambar IV.3. 3.1 Gambar before input barang

Selanjutnya pada *text field* tersebut dimasukan *input* tertentu dan menekan *button insert*. Berikut tampilan ketika Proses insert telah berhasil dilakukan :

The screenshot shows the same "Entry Data Barang" window, but now the input fields are filled with data. The "Kode Barang" field contains "B009", "Nama Barang" contains "Jari-Jari", "Harga Beli" contains "100000", "Harga Jual" contains "115000", and "Quantity" contains "1". The table now has five rows of data:

Kode Bar...	Nama Ba...	Harga Beli	Harga Jual	Quantity
B002	Ban	50000	50000	5
B003	Aki	40000	50000	5
B004	Busi	20000	30000	5
B005	Jari - Jari	50000	60000	5

The buttons and search section remain the same as in the previous screenshot.

Gambar IV.3. 3.2 after input barang

IV.3.4 Fitur *Delete*

Fitur ini digunakan untuk proses menghapus data dari *text field* dan database. Proses implementasi pada fitur *delete* dilakukan dengan memilih data yang ingin di hapus dari *database* dan menekan *button delete*.

Entry Data Barang

Kode Barang:

Nama Barang:

Harga Beli:

Harga Jual:

Quantity:

Kode Bar...	Nama Ba...	Harga Beli	Harga Jual	Quantity
B001	Oli	50000	60000	5
B002	Ban	50000	50000	5
B003	Aki	40000	50000	5
B004	Busi	20000	30000	5

Buttons: Save, Delete, Update, View, Menu

Cari Berdasarkan Nama Barang

Search

Gambar IV.3. 4.1 before delete barang

Berikut tampilan *form* data barang setelah proses *Delete* berhasil dilakukan:

Entry Data Barang

Kode Barang:

Nama Barang:

Harga Beli:

Harga Jual:

Quantity:

Kode Bar...	Nama Ba...	Harga Beli	Harga Jual	Quantity
B002	Ban	50000	50000	5
B003	Aki	40000	50000	5
B004	Busi	20000	30000	5
B005	Jari - Jari	50000	60000	5

Buttons: Save, Delete, Update, View, Menu

Cari Berdasarkan Nama Barang

Search

Gambar IV.3. 4.2 after delete barang

IV.3.5 Fitur *Update*

Fitur *Update* digunakan untuk meng *Update* atau melakukan perubahan pada database atau transaksi.

The screenshot shows a software window titled "Entry Data Barang". On the left, there are five text input fields: "Kode Barang" (containing "B009"), "Nama Barang" (containing "Jari-Jari"), "Harga Beli" (containing "100000"), "Harga Jual" (containing "115000"), and "Quantity" (containing "1"). To the right of these fields is a table with five columns: "Kode Bar...", "Nama Ba...", "Harga Beli", "Harga Jual", and "Quantity". The table contains five rows of data. Below the table are five buttons: "Save", "Delete", "Update", "View", and "Menu". At the bottom right, there is a search section with the text "Cari Berdasarkan Nama Barang", an empty text box, and a "Search" button.

Kode Bar...	Nama Ba...	Harga Beli	Harga Jual	Quantity
B002	Ban	50000	50000	5
B003	Aki	40000	50000	5
B004	Busi	20000	30000	5
B005	Jari - Jari	50000	60000	5

Gambar IV.3. 5.1 gambar update barang

Selanjutnya masukan inputan apa yang akan kita *Update* kedalam *Text field*. Misalkan *Update stock* barang. Berikut tampilan *form* data barang pada saat melakukan proses *Update*, setelah menekan *Button Up date* maka akan terjadi perubahan pada *database* pada kolom *stock* barang

IV.3.6 Fitur cari data

Fitur ini berfungsi untuk mencari data pada *database* dengan masukan nama pelanggan untuk *Form* Data Transaksi dan masukkan nama barang untuk *Form* Data Barang.

Kode Bar...	Nama Ba...	Harga Beli	Harga Jual	Quantity
B002	Ban	50000	50000	5
B003	Aki	40000	50000	5
B004	Busi	20000	30000	5
B005	Jari - Jari	50000	60000	5

Gambar IV.3. 6.1 gambar cari barang

IV.4 SKENARIO PENGUJIAN

- Apakah fungsi Login bisa berjalan dengan baik ? ☐
- Apakah *Menu Utama* tampil saat *button Login* di tekan ? ☐
- Apakah menu *File* terdapat pada *Menu Utama* dan memiliki Sub menu Data Barang, Data Transaksi , Data Karyawan, Data Customer ? ☐
- Apakah sub menu Data berfungsi untuk membuka *Form* Data Barang? ☐

- Apakah sub menu Transaksi berfungsi untuk membuka *Form* Data Transaksi ? ☐
- Apakah fitur *insert, update, delete, reset* dan cari terdapat pada sub menu yang ada ? ☐
- Saat melakukan *insert* data pada *form* Data Barang, Data Transaksi, Data Karyawan, Data Customer. Apakah fungsi *insert* berfungsi dengan baik? ☐
- Saat melakukan *update* data pada *form* Data Barang, Data Transaksi, Data Karyawan, Data Customer. Apakah fungsi *update* berfungsi dengan baik? ☐
- Saat melakukan *delete* data pada *form* Data Barang, Data Transaksi, Data Karyawan, Data Customer. Apakah fungsi *delete* berfungsi dengan baik? ☐
- Saat melakukan *reset* data pada *form* Data Barang, Data Transaksi, Data Karyawan, Data Customer. Apakah fungsi *reset* berfungsi dengan baik? ☐
- Apakah fungsi cari, dapat mencari data hanya dengan memasukkan beberapa huruf , tanpa memasukkan nama barang atau nama pelanggan secara utuh? ☐

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Berdasarkan hasil analisis yang dilakukan terhadap *Aplikasi Transaksi Bengkel* maka dapat disimpulkan beberapa hal antara lain :

1. Untuk menjalankan aplikasi ini diperlukan aplikasi pendukung lainnya yaitu *Mysql*, *Netbeans* dan *iReport*.
2. Aplikasi ini hanya bisa menjalankan 5 fungsi utama saja yaitu *insert*, *delete*, *update*, *reset*, dan cari.
3. Proses pengujian pada aplikasi ini menggunakan “***Metode Black Box Testing***” yaitu metode pengujian dengan mengutamakan pengujian terhadap kebutuhan fungsional aplikasi.
4. Aplikasi ini tidak dapat diakses secara online.

5.2 Saran

Aplikasi ini masih membutuhkan pengembangan yang lebih baik lagi agar dapat digunakan dengan lebih mudah dan simple serta dapat diakses secara online. Untuk itulah saran dan kritik sangat kami butuhkan untuk membangun aplikasi ini kearah yang lebih baik kedepannya.

DAFTAR PUSTAKA

Referensi :

1. Kadir, Abdul. 2003. Dasar Pemrograman Java™ 2. Bandung: Andi.
2. Komputer, Wahana. 2008. Membuat Aplikasi Database Dengan Java dan MySql : Bandung: Andi.
3. Pressman, Roger S. 2002. Rekayasa Perangkat Lunak. Bandung : Andi.
4. <http://romisatriawahono.net>
5. www.Google.com

