

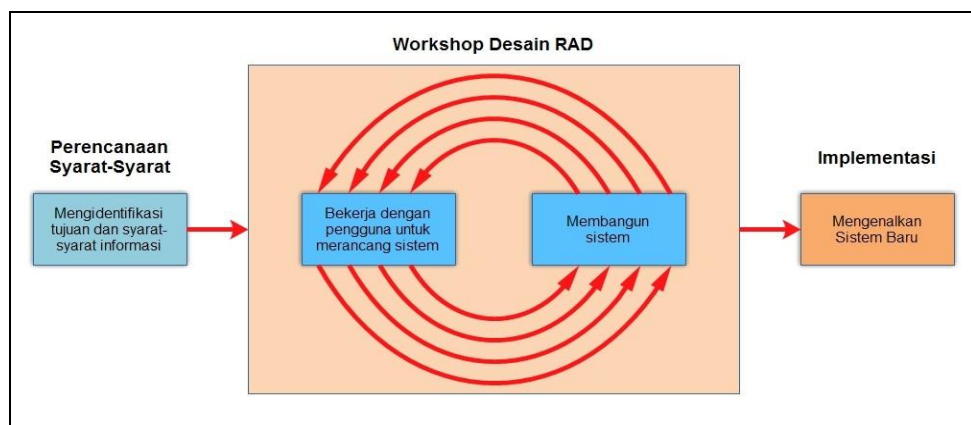
BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Metodologi Pengembangan Sistem

Metodologi adalah kesatuan metode-metode, prosedur-prosedur, konsep konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan oleh suatu ilmu pengetahuan, seni atau disiplin lainnya (Jogiyanto, 2005).

Metodologi pengembangan sistem yang akan digunakan adalah *Rapid Application Development (RAD)* atau *rapid prototyping*. Model ini adalah model proses pembangunan perangkat lunak yang termasuk dalam teknik bertingkat (*incremental*). Model RAD menekankan pada siklus pembangunan sistem informasi yang pendek, singkat, dan tepat. Batasan yang penting untuk model ini adalah waktu yang singkat. RAD menggunakan metode berulang (*iterative*) dalam pengembangan sistem, dimana model kerja (*working model*) sistem dikonstruksikan pada tahapan awal pengembangan dengan tujuan penetapan kebutuhan (*requirements*) *user* dan selanjutnya akan disingkirkan (Britton, 2001).



Gambar 3.1 *Rapid Application Development (RAD) Model*

(Sumber : Kendall, 2008)

Terdapat tiga fase besar dalam tahap RAD yang melibatkan pengguna dan analis dalam penilaian, desain, dan implementasi (Kendall, 2008), yaitu sebagai berikut :

A. *Requirements Planning* (Perencanaan Syarat-Syarat)

Dalam fase ini, pengguna dan analis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan

B. *RAD Design Workshop* (Workshop Desain RAD)

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *Workshop*. Analis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *Workshop* desain RAD, pengguna merespon prototipe yang ada dan analis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna. Menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi (Kendall, 2010).

C. *Implementation* (Implementasi)

Pada fase implementasi ini, analis bekerja dengan para pengguna secara intens selama *Workshop* dan merancang aspek bisnis dan nonteknis perusahaan. Setelah aspek disetujui dan dibangun serta disaring, maka diperkenalkan kepada perusahaan.

3.2. Perencanaan Sistem dengan Menggunakan RAD

Pada perencanaan sistem ini menggunakan metode Rapid Application Development (RAD) yang memiliki 3 tahapan yaitu : *Requirements Planning*, *RAD Design Workshop*, dan *Implementation*.

3.2.1. *Requirements Planning* (Perencanaan Syarat-Syarat)

Pada fase ini, dilakukan analisis mengenai analisis kebutuhan yang didapatkan melalui wawancara dengan pimpinan CV.Athala Lautan Mulia. Setelah mendapatkan data-data dari hasil wawancara maka disimpulkan bahwa CV.Athala Lautan Mulia membutuhkan aplikasi sistem penjualan untuk membantu transaksi penjualan sehari-hari.

A. Analisis Sistem

Dalam menganalisis kelemahan sistem yang lama, peneliti melakukan pembagian menjadi 2 bagian yang pertama yaitu, prosedur sistem yang sedang berjalan dan yang kedua, mengidentifikasi masalah untuk menemukan kelemahan pada sistem yang lama.

1. Prosedur Sistem Yang Sedang Berjalan

Analisis mengenai sistem yang sedang berjalan, untuk mendapatkan informasi yang jelas mengenai permasalahan yang ada pada sistem penjualan CV. Athala Lautan Mulia. Prosedur penjualan pada CV. Athala Lautan Mulia :

1. Pelanggan akan datang ke CV.Athala Lautan Mulia untuk melakukan pemesanan barang yang akan dibeli.
2. Karyawan akan mencatat pesanan apa yang akan dibeli oleh pelanggan.
3. Setelah pesanan dibuat, karyawan akan mengecek persediaan barang yang telah dipesan pelanggan di gudang.

4. Kemudian, karyawan akan mengkonfirmasi ketersediaan barang kepada pelanggan dan memberikan catatan barang tersebut kepada pelanggan untuk dilakukan pembayaran ke manager store
5. Setelah melakukan pembayaran manager store akan mengkonfirmasi ke gudang mengenai barang yang akan dikirimkan.
6. Karyawan akan melakukan pengiriman barang yang telah dipesan, setelah barang dikirimkan karyawan akan mengkonfirmasi kepada manager store, lalu manager store akan memberikan laporan transaksi penjualan kepada Direktur/pimpinan

2. Identifikasi Masalah

Identifikasi masalah yang terdapat pada CV. Athala Lautan Mulia dilakukan dengan menggunakan Analisis *PIECES* yang memiliki 6 kriteria yaitu *Performance*, *Information*, *Economic*, *Control*, *Efficiency*, dan *Service*.

1. Analisis *PIECES*

A. Analisis *Performance* (Analisis Kinerja)

Analisis kinerja merupakan peningkatan terhadap kinerja sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari *throughput* (T_p) dan *response time* (T_r). *Throughput* (T_p) adalah jumlah dari pekerjaan yang dapat dilakukan suatu sistem tertentu, sedangkan *response time* (T_r) adalah rata-rata waktu yang tertunda diantara dua pekerjaan ditambah dengan *response time* untuk menangani pekerjaan tersebut.

Setelah dilakukan analisis terhadap sistem lama pada CV. Athala Lautan Mulia terdapat beberapa kelemahan pada kinerjanya, yaitu proses pengolahan data dan

transaksi memerlukan waktu yang cukup lama untuk merekap data nya. Hal ini terjadi karena proses pencatatan nya yang masih manual.

B. Analisis Informasi (*Information*)

Merupakan peningkatan terhadap kualitas informasi yang diberikan. Setelah melakukan analisis terhadap sistem lama pada CV. Athala Lautan Mulia, adanya kelemahan pada segi informasi dimana sering adanya kesalahan untuk mengetahui harga jual barang, serta sering terjadi kesalahan perhitungan total harga pada transaksi.

C. Analisis Ekonomi (*Economy*)

Merupakan peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan maupun penurunan-penurunan biaya yang terjadi. Setelah melakukan analisis terhadap sistem lama pada CV. Athala Lautan Mulia memiliki kelemahan dari segi ekonomi yaitu untuk menyimpan data masih menggunakan kertas dan pena untuk mencatat transaksi yang ada, dan sering terjadinya kesalahan dalam masalah laporan keuangan sehingga menimbulkan kerugian bagi perusahaan.

D. Analisis Pengendalian (*Control*)

Merupakan peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta kecurangan-kecurangan yang terjadi. Setelah melakukan analisis terhadap sistem lama pada CV. Athala Lautan Mulia, masih terdapat kelemahan dari segi pengendalian, yaitu data transaksi pembelian serta penjualan tidak terdata dengan baik, sehingga sulit untuk menentukan keuntungan maupun kerugian yang didapatkan oleh CV. Athala Lautan Mulia.

E. Analisis Efisiensi (*Efficiency*)

Merupakan peningkatan terhadap efisiensi operasi. Efisiensi berbeda dengan ekonomis. Bila ekonomis berhubungan dengan jumlah sumber daya yang digunakan, efisiensi berhubungan dengan bagaimana sumber data digunakan dengan pemborosan paling minimum. Efisiensi dapat diukur dari output dibagi dengan input.

Setelah melakukan analisis terhadap sistem lama pada CV. Athala Lautan Mulia terdapat kelemahan pada segi efisiensi. Pada sistem yang lama, CV. Athala Lautan Mulia masih menyimpan data mengenai penjualan, pembelian, ataupun transaksi secara manual dan jika terjadi hal yang tidak diinginkan seperti kebakaran/bencana alam maka, data akan hilang dan tidak dapat dikembalikan.

F. Analisis Pelayanan (*Service*)

Merupakan peningkatan terhadap pelayanan yang diberikan oleh sistem. Setelah melakukan analisis terhadap sistem lama pada CV. Athala Lautan Mulia, terdapat kelemahan pada bagian pelayanan yang lama, yaitu pelayanan masih harus dilakukan secara manual dan mengharuskan pelanggan untuk datang langsung jika ingin membeli barang. Hal ini tentu membutuhkan waktu yang lama, seperti pencatatan transaksi barang yang dibeli oleh pelanggan dan juga perhitungan total yang harus dibayarkan oleh pelanggan.

2. Analisis Sebab Akibat

Berdasarkan hasil dari analisis *PIECES* diatas yang membahas mengenai permasalahan yang terdapat pada CV. Athala Lautan

Mulia, maka akan dijelaskan mengenai akibat yang terjadi dari permasalahan yang ada pada tabel di bawah ini.

Tabel 3.1 Analisis Sebab Akibat

| No | Kriteria | Permasalahan | Sebab | Akibat |
|----|----------------------------------|--|---|--|
| 1. | Kinerja/ <i>Perfomance</i> | Mencatat data transaksi yang masih secara manual | Proses pencatatan yang masih manual oleh karyawan | Memakan waktu lama serta penggunaan kertas yang banyak dan risiko kehilangan data tersebut lebih besar |
| 2 | Informasi/ <i>Information</i> | Kesalahan dalam menentukan harga jual barang | Penentuan harga untuk barang yang dijual sering mengalami kesalahan | Sulit untuk mentotal seluruh harga barang yang dibeli oleh pelanggan |
| 3 | Ekonomi/ <i>Economy</i> | Kesalahan dalam melapor masalah keuangan | Pencatatan masih secara manual | Dampak kerugian yang diberikan kepada perusahaan |

Tabel 3.1 Analisis Sebab Akibat (*Lanjutan*)

| No | Kriteria | Permasalahan | Sebab | Akibat |
|----|---------------------------------|--|---|--|
| 4 | Pengendalian/ <i>Control</i> | Data penjualan dan pembelian yang tidak terdata dengan baik | Data tidak tersimpan akibat pencatatan secara manual dan juga karena adanya kesalahan dari karyawan | Sulit untuk menemukan keuntungan dan kerugian pada CV. Athala Lautan Mulia |
| 5 | Efisiensi/ <i>Efficiency</i> | Data transaksi disimpan secara manual pada perusahaan | Data transaksi disimpan secara manual yang dicatat oleh karyawan | Data dapat hilang jika terjadi kesalahan saat menginput data ataupun ketika adanya kebakaran/ bencana alam |
| 6 | Pelayanan/ <i>Service</i> | Pelayanan secara langsung yang mengharuskan pelanggan untuk datang ke lokasi penjualan | Pelanggan datang secara langsung untuk membeli barang yang dibutuhkan | Memakan waktu yang lama untuk pencatatan barang yang dibeli oleh pelanggan serta untuk menjumlahkan total harga yang harus dibayarkan oleh |

Tabel 3.1 Analisis Sebab Akibat (*Lanjutan*)

| No | Kriteria | Permasalahan | Sebab | Akibat |
|----|----------|--------------|-------|--|
| | | | | pelanggan karena masih dihitung secara manual menggunakan kalkulator |

Berdasarkan dari tabel analisis sebab akibat diatas, dapat disimpulkan bahwa CV. Athala Lautan Mulia membutuhkan solusi untuk menyelesaikan masalah-masalah yang timbul. Solusi yang dapat diberikan oleh peneliti adalah dengan membuat aplikasi sistem penjualan berbasis *web*.

3. Analisis Kebutuhan Sistem yang Diusulkan

Dalam menganalisis kebutuhan pada sistem akan dibagi menjadi 2 bagian, yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional berisi proses-proses apa saja yang harus dilakukan oleh sistem pada situasi khusus, kebutuhan fungsional ini juga berisi informasi-informasi apa saja yang harus dihasilkan oleh sistem. Kebutuhan non-fungsional berisi properti perilaku yang dimiliki oleh sistem proses pengembangan, standarisasi, operasional, dan kinerja.

A. Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya akan dilakukan oleh sistem. Kebutuhan fungsional pada CV. Athala Lautan Mulia adalah sebagai berikut :

1. Sistem yang bisa menambah informasi mengenai pelanggan.

2. Sistem yang menyediakan barang yang dijual beserta dengan informasi harga
3. Sistem yang dapat menyimpan data stok barang
4. Sistem yang dapat membuat transaksi beserta dengan total pembelian serta kembalian pelanggan
5. Sistem yang memiliki *multi-user* sehingga bisa diakses oleh karyawan maupun pemimpin.
6. Sistem yang mampu memanipulasi data stok barang melalui user level admin. Admin mampu melakukan *update*, *delete* serta menyimpan stok barang pada *database*.
7. Sistem yang dapat menyajikan data barang yang dimiliki oleh CV. Athala Lautan Mulia
8. Sistem yang dapat memberikan laporan per tanggal, per bulan, dan per tahun

B. Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang diperlukan pada properti perilaku yang dimiliki oleh sistem. Beberapa kebutuhan non-fungsional yang tentu saja dapat dipenuhi dengan pembangunan “Aplikasi Sistem Penjualan pada CV. Athala Lautan Mulia”.

1. Keamanan

Sistem yang dibuat memiliki proteksi *username* dan *password* sehingga hanya orang yang memiliki kewenangan yang bisa memodifikasi data.

2. Operasional

A. Perangkat keras (*hardware*) yang dibutuhkan untuk sistem pada CV. Athala Lautan Mulia yang sederhana, yaitu *RAM 2GB*, *monitor*, *keyboard*, dan *mouse*.

B. Perangkat lunak (*software*) yang diperlukan untuk pembuatan aplikasi sistem penjualan pada CV. Athala

Lautan Mulia meliputi sistem operasi *Windows 10*, dan *software XAMPP*.

3. Informasi

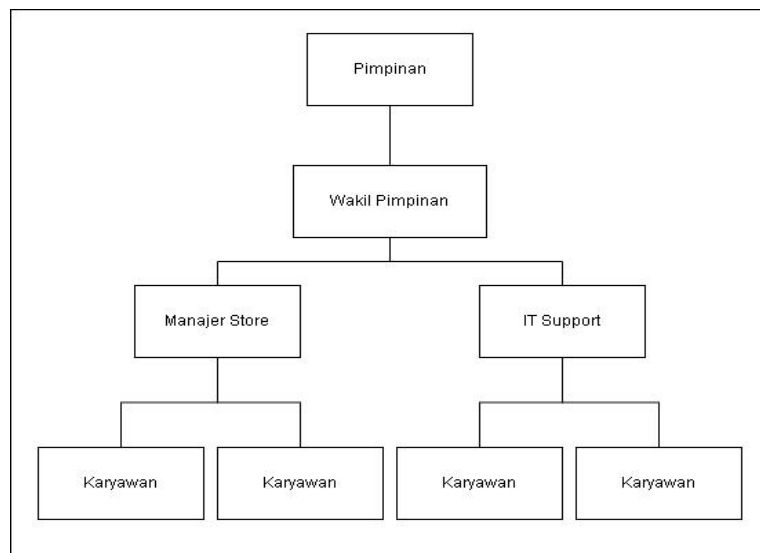
Dengan adanya aplikasi sistem penjualan pada CV. Athala Lautan Mulia, karyawan sekaligus pemimpin bisa mendapatkan informasi mengenai data transaksi yang sudah tersimpan

4. Kinerja

Dengan pembuatan aplikasi sistem penjualan pada CV. Athala Lautan Mulia, karyawan dapat dengan cepat mendapatkan data transaksi, data barang jual, serta pencarian data barang, sedangkan pada sistem lama karyawan akan sulit melakukan hal tersebut secara manual

5. Struktur Organisasi

Dengan pembuatan aplikasi sistem penjualan pada CV. Athala Lautan Mulia, diharapkan juga dapat menentukan struktur organisasi yang baru agar memiliki bagian yang bisa merawat aplikasi tersebut.



Gambar 3.2 Struktur Organisasi Usulan

C. Prosedur Sistem yang Diusulkan

Prosedur sistem baru yang akan diusulkan adalah sebagai berikut :

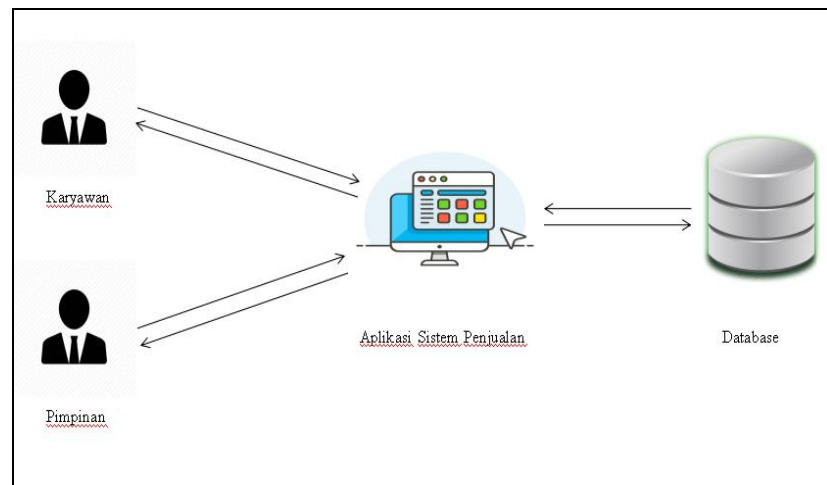
- a. Sistem yang diusulkan akan memiliki menu *login* untuk bisa mengakses aplikasi, sehingga hanya pihak yang memiliki kewenangan yang bisa menggunakan sistem.
- b. Sistem yang diusulkan dapat melakukan proses penginputan data barang.
- c. Sistem yang diusulkan dapat menyimpan semua data barang
- d. Sistem yang diusulkan dapat menyajikan semua data barang
- e. Sistem yang diusulkan dapat menyimpan data pelanggan
- f. Sistem yang diusulkan dapat menyajikan data pelanggan
- g. Sistem yang diusulkan dapat menghitung jumlah total yang harus dibayarkan oleh pelanggan
- h. Sistem yang diusulkan dapat menghitung diskon yang diberikan toko kepada pelanggan
- i. Sistem yang diusulkan dapat memberikan laporan penjualan per tanggal, bulan dan tahun

D. Perancangan Sistem yang Diusulkan

Gambaran mengenai sistem yang akan dibuat akan direpresentasikan dalam beberapa bentuk model, antara lain :

a. Perancangan Arsitektur Sistem

Arsitektur dari sistem yang akan dibuat secara keseluruhan pada CV. Athala Lautan Mulia dapat dilihat pada gambar dibawah ini



Gambar 3.3 Arsitektur Sistem yang Diusulkan

Gambar diatas menjelaskan bahwa seluruh user dapat menginput data-data ke komputer dan tersimpan kedalam *database*. *Database* yang telah disimpan dapat ditampilkan lagi di dalam aplikasi penjualan dan dapat memberikan informasi kepada semua user yang membutuhkan informasi tersebut.

b. Pemodelan Proses Sistem

UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak (Gata dan Gata, 2013). UML merupakan metodologi dalam mengembangkan sistem berorientasi objek. Alat bantu/*tools* yang digunakan dalam perancangan berorientasi objek berbasis UML pada pembangunan Aplikasi sistem penjualan pada CV. Athala Lautan Mulia adalah sebagai berikut :

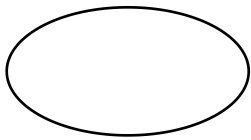
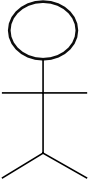


1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam

sistem informasi dan siapa saja yang berhak menggunakan fungsi tersebut.

Simbol-simbol yang terdapat dalam *Use Case Diagram* yaitu :

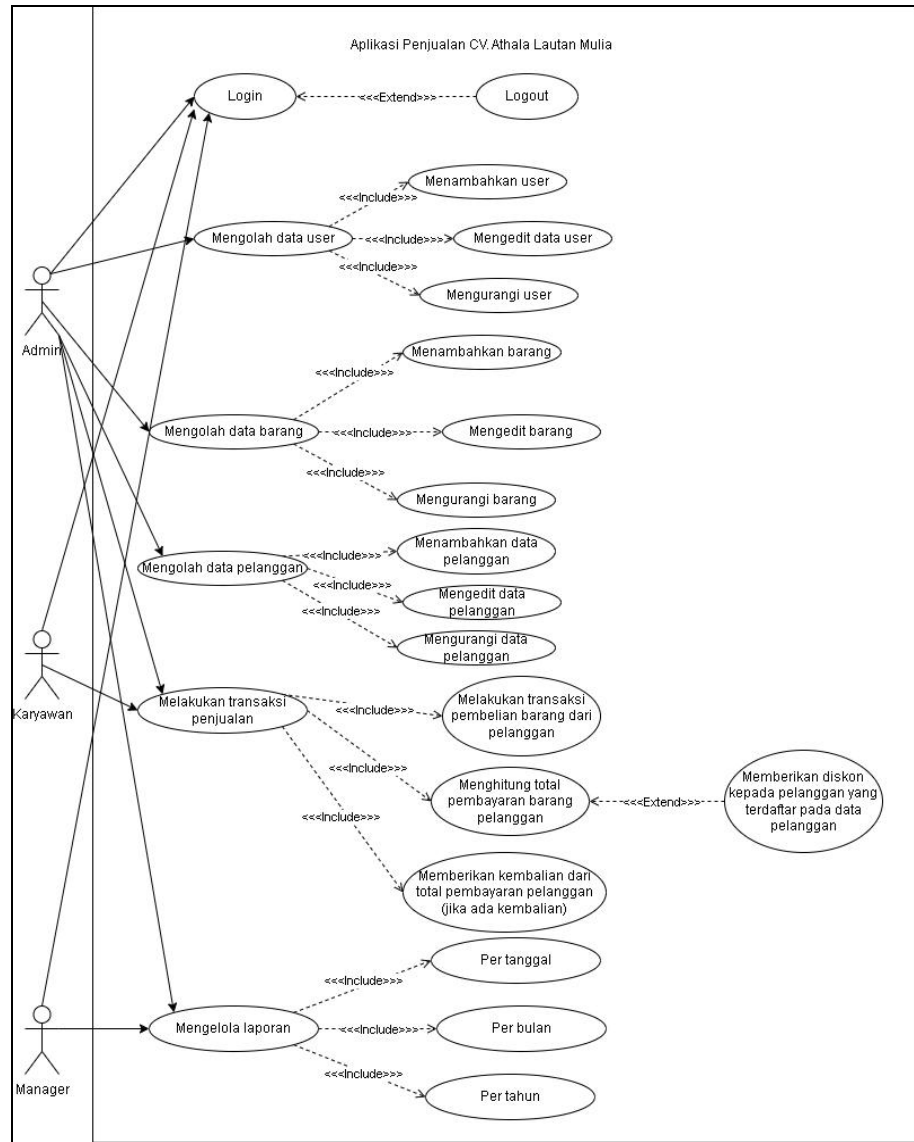
Tabel 3.2 Simbol-Simbol *Use Case Diagram*

| Simbol | Keterangan |
|---|--|
|  | <i>Use Case</i> : fungsi yang disediakan sistem untuk pertukaran pesan dengan aktor. Biasanya dimulai dengan kata kerja untuk penamaan. |
|  | <i>Actor</i> : Orang atau sistem yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi <i>actor</i> , harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan pada koneksi target sistem. |
|  | Komunikasi antara <i>actor</i> dengan <i>use case</i> yang saling memiliki interaksi. |
|  | Generalisasi adalah hubungan antara dua buah <i>use case</i> , di mana yang satu adalah yang lebih umum. |
| ---<<<<Extend>>>>--- | <i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi. |
| ---<<<<Include>>>>--- | <i>Include</i> : Di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain. |

A. *Use Case Diagram* yang Diusulkan

Pada diagram *use case* yang diusulkan akan dijelaskan mengenai perilaku pengguna pada sistem dan pengguna tersebut bisa melakukan

apa saja di dalam sistem. *Use Case Diagram* yang diusulkan dapat dilihat pada Gambar 3.4.



Gambar 3.4 Use Case Diagram Aplikasi Penjualan CV. Athala Lautan Mulia

Berdasarkan Gambar 3.4 diatas, ada 3 aktor dalam *use case diagram* yang diusulkan yaitu *Admin*, *Karyawan*, dan *Manager* yang bisa menggunakan aplikasi penjualan. *Admin* dapat mengelola seluruh data yang ada pada aplikasi. *Karyawan* dapat mengelola data transaksi. *Manager* dapat mengelola data laporan penjualan. Semua aktor tersebut




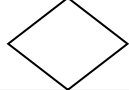
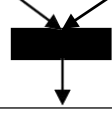
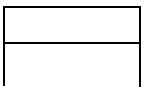
jika ingin menggunakan aplikasi penjualan harus melakukan login terlebih dahulu.

2. Activity Diagram

Activity Diagram memberikan gambaran mengenai *workflow* (alur kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

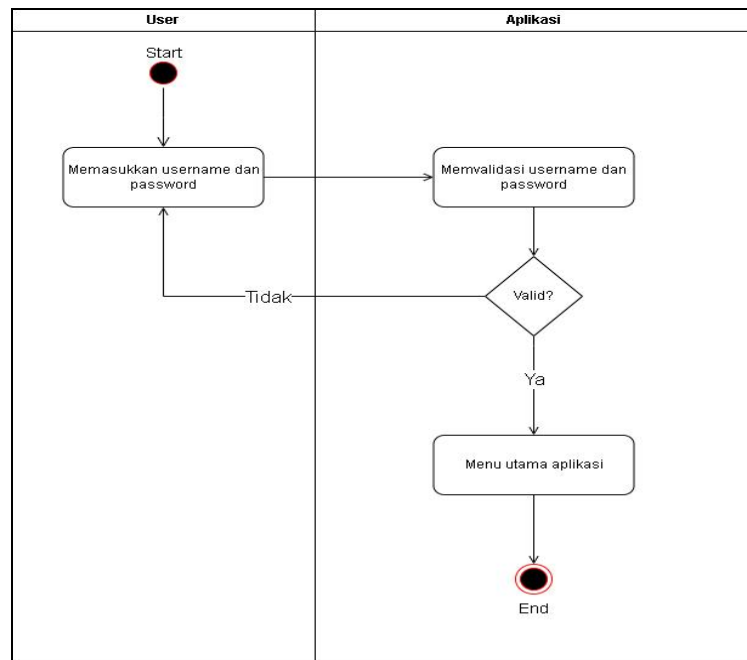
Berikut simbol-simbol yang ada pada *activity diagram* yang bisa dilihat pada tabel 3.3

Tabel 3.3 Simbol-simbol Activity Diagram

| Simbol | Keterangan |
|---|---|
|  | <i>Start Point</i> : untuk menyatakan awal dari suatu proses |
|  | <i>End Point</i> : untuk menyatakan akhir dari suatu proses |
|  | <i>Activities</i> : aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
|  | <i>Decision</i> : asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
|  | <i>Join</i> : asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
|  | <i>Swimlane</i> : memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

A. Activity Diagram Login yang diusulkan

Pada *activity diagram login* yang diusulkan akan dijelaskan mengenai aktivitas yang dilakukan oleh *user* ketika *login* ke dalam aplikasi yang telah dibangun. *Activity diagram login* yang diusulkan dapat dilihat pada Gambar 3.5.

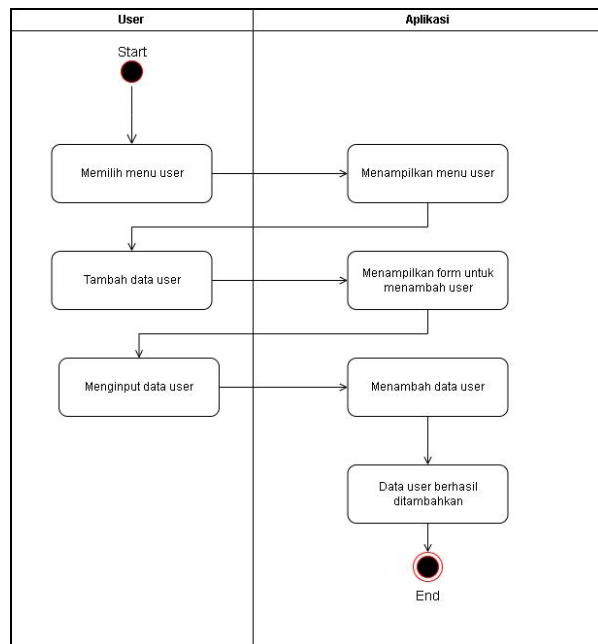


Gambar 3.5 Activity Diagram Login Aplikasi Penjualan CV.Athala Lautan Mulia

Berdasarkan Gambar 3.5 *activity diagram login* yang diusulkan dapat dilihat bahwa *user* yang ingin menggunakan aplikasi harus memasukkan *username* dan *password* terlebih dahulu, dan setelah itu sistem akan memvalidasi apakah sudah sesuai dengan data yang ada pada *database*, jika valid maka *user* akan lanjut ke menu utama aplikasi dan jika tidak valid maka akan kembali kepada halaman *login*.

B. Activity Diagram Menambah *user* yang diusulkan

Pada *activity diagram* menambah *user* yang diusulkan akan dijelaskan mengenai *user* dalam mengelola tambah *user*. *Activity diagram* menambah *user* dapat dilihat pada Gambar 3.6.

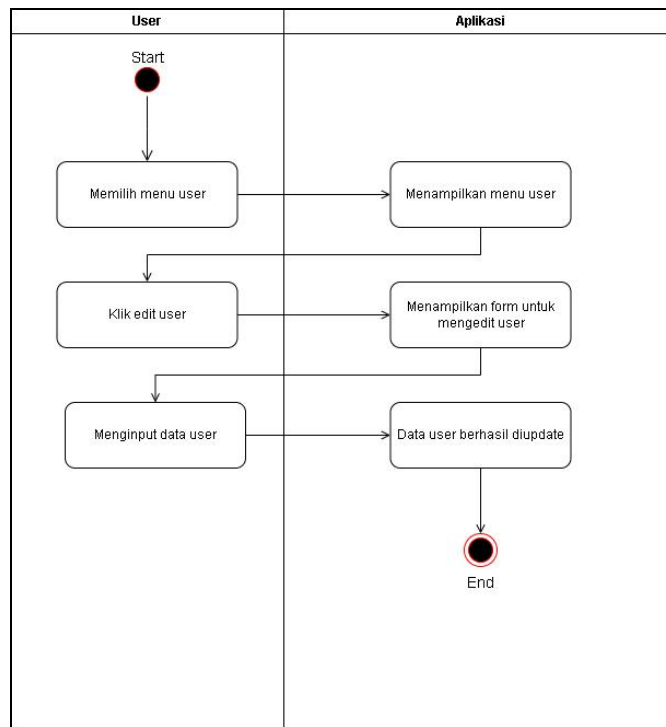


Gambar 3.6 Activity Diagram Menambah User Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* tambah *user* baru yang dilakukan *user* pertama kali adalah memilih menu *user*, setelah itu aplikasi akan menampilkan data *user*. Setelah data *user* tampil, selanjutnya *user* mengklik tombol tambah untuk menambah *user* baru, lalu aplikasi akan menampilkan *form* tambah *user*, *user* lalu menginput data *user*. Setelah itu data *user* yang baru berhasil ditambahkan.

C. Activity Diagram Edit User yang diusulkan

Pada *activity diagram edit user* yang diusulkan akan dijelaskan mengenai aktivitas *user* untuk mengedit data *user* yang sebelumnya di-inputkan di aplikasi. *Activity diagram edit user* yang diusulkan dapat dilihat pada Gambar 3.7.

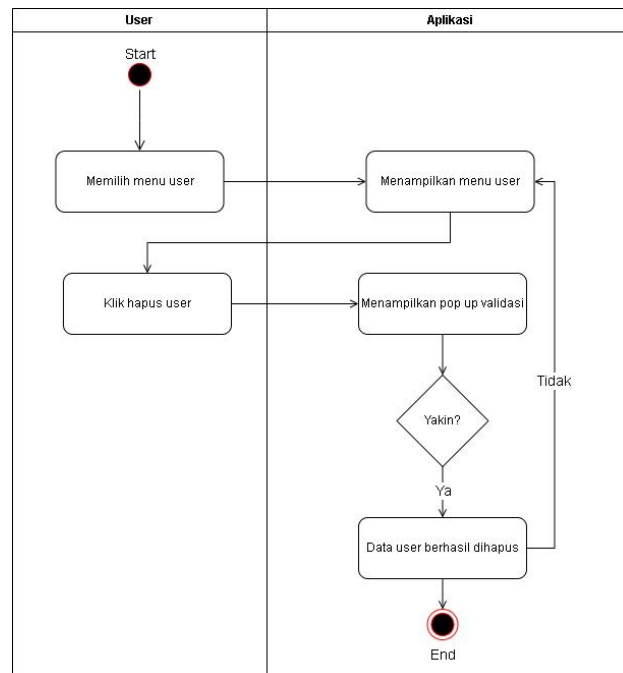


Gambar 3.7 Activity Diagram Edit User Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* edit *user* yang pertama kali dilakukan *user* adalah memilih menu *user*, setelah diklik aplikasi akan menampilkan seluruh data *user*. Setelah seluruh data *user* tampil, *user* akan memilih data yang akan di edit dan mengklik tombol edit lalu aplikasi akan menampilkan *form* edit *user*, *user* akan mengedit data *user* sesuai dengan kebutuhan. Setelah mengisi *form* edit, data *user* berhasil di *update*.

D. Activity Diagram Hapus User yang diusulkan

Pada *activity diagram* hapus *user* yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam menghapus data *user* yang sebelumnya sudah di-*inputkan* ke dalam aplikasi. *Activity diagram* hapus *user* yang diusulkan dapat dilihat pada Gambar 3.8.

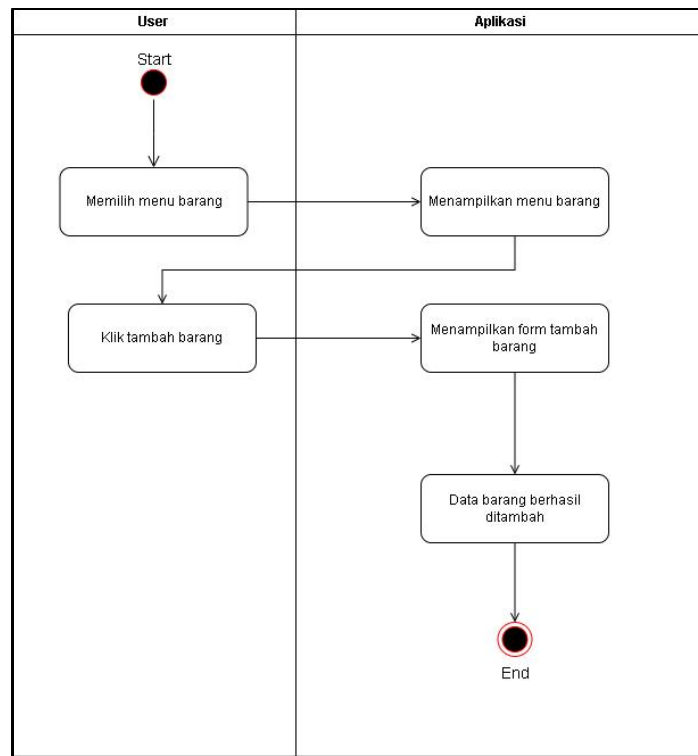


Gambar 3.8 Activity Diagram Hapus User Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* hapus *user* yang pertama kali dilakukan *user* adalah memilih menu *user*, setelah diklik maka aplikasi akan menampilkan semua data *user*. Setelah seluruh data *user* tampil, *user* akan memilih data yang akan di hapus dan mengklik tombol hapus data lalu aplikasi akan menampilkan *pop-up* validasi : ‘Yakin Ingin Menghapus?’, kemudian jika *user* memilih ya, maka aplikasi akan menghapus data tersebut, jika *user* memilih batal, maka aplikasi akan kembali menampilkan data *user*.

E. Activity Diagram Tambah Barang yang diusulkan

Pada *activity diagram* tambah barang yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mengelola data barang pada aplikasi. *Activity diagram* tambah barang yang diusulkan dapat dilihat pada Gambar 3.9.

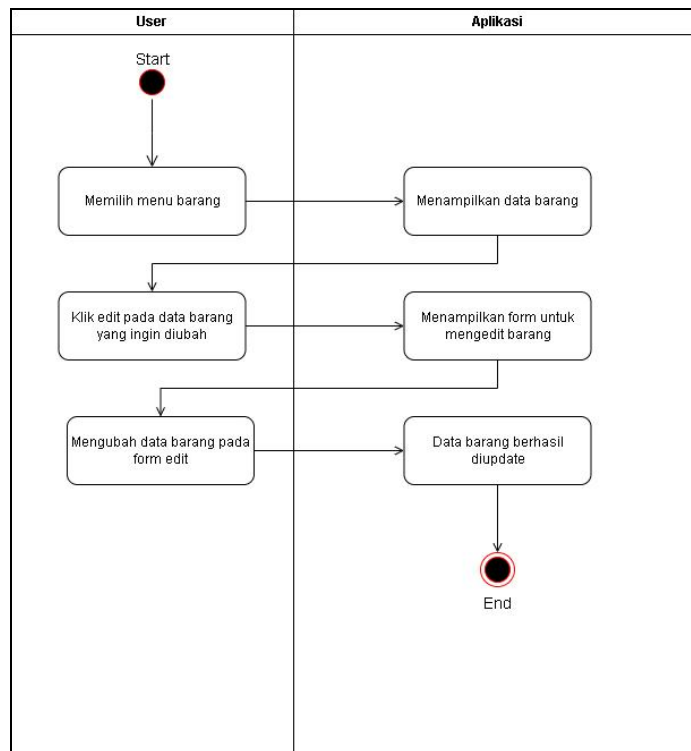


Gambar 3.9 Activity Diagram Tambah data Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* tambah data barang yang pertama kali dilakukan oleh *user* adalah memilih menu barang. Kemudian aplikasi akan menampilkan data barang, setelah data barang tampil *user* mengklik tombol tambah barang lalu aplikasi akan menampilkan *form* untuk menambahkan barang, *user* akan meng-*input* data barang. Setelah data barang di-*input*kan maka data barang berhasil ditambahkan.

F. Activity Diagram Edit data Barang yang diusulkan

Pada *activity diagram* edit data barang yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mengedit data barang yang sebelumnya sudah di-*input*kan ke dalam aplikasi. *Activity diagram* edit data barang yang diusulkan dapat dilihat pada Gambar 3.10.

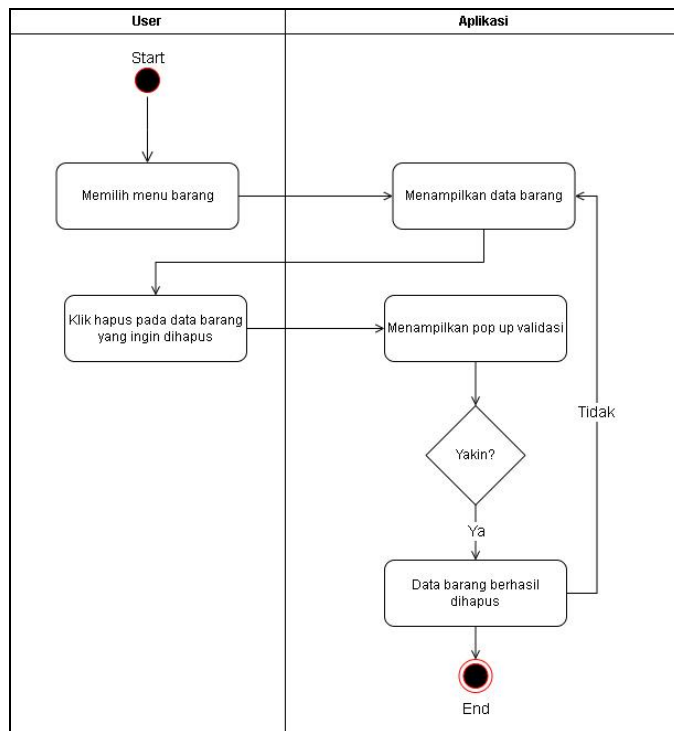


Gambar 3.10 Activity Diagram Edit Data Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* yang pertama kali dilakukan oleh *user* adalah memilih menu barang, setelah itu aplikasi akan menampilkan semua data barang. Setelah semua data barang tampil, *user* akan memilih data yang akan di edit dan mengklik tombol edit data lalu aplikasi akan menampilkan *form* edit barang, *user* akan mengedit data barang sesuai dengan kebutuhan. Setelah mengisi *form* edit, data barang berhasil di *update*.

G. Activity Diagram Hapus Data Barang yang diusulkan

Pada *activity diagram* hapus data barang yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam menghapus data barang yang sebelumnya sudah di-*inputkan* ke dalam aplikasi. *Activity diagram* hapus data barang yang diusulkan dapat dilihat pada Gambar 3.11.

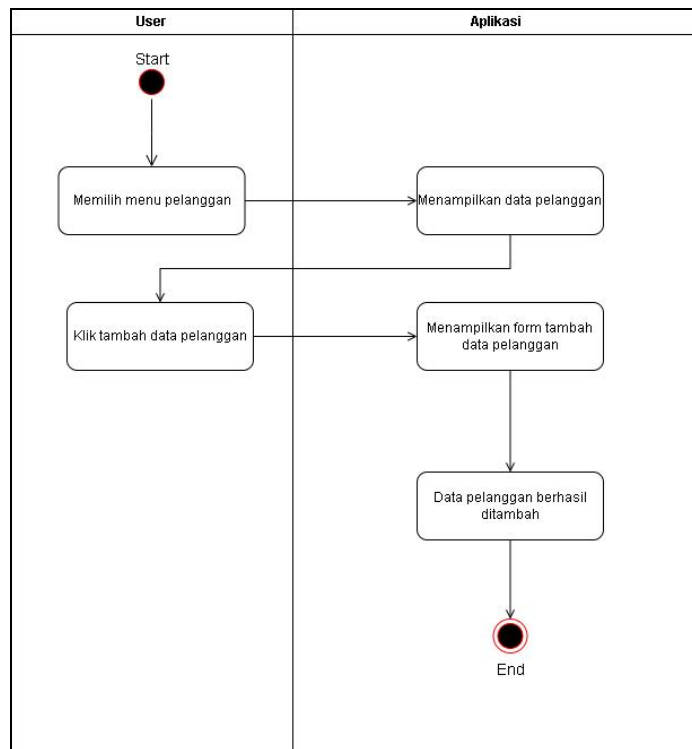


Gambar 3.11 Activity Diagram Hapus Data Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* yang pertama kali dilakukan oleh *user* adalah memilih menu barang. Setelah memilih menu barang maka aplikasi akan menampilkan seluruh data barang. Setelah seluruh data barang tampil, *user* akan memilih data yang akan di hapus dan mengklik tombol hapus data lalu aplikasi akan menampilkan *pop-up* validasi : ‘Yakin Ingin Menghapus Data?’, jika *user* memilih ya, maka aplikasi akan menghapus data tersebut, jika *user* memilih batal, maka aplikasi akan kembali menampilkan data barang.

H. Activity Diagram Tambah Data Pelanggan yang diusulkan

Pada *activity diagram* tambah data pelanggan yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mengelola data pelanggan. *Activity diagram* tambah data pelanggan yang diusulkan dapat dilihat pada Gambar 3.12.

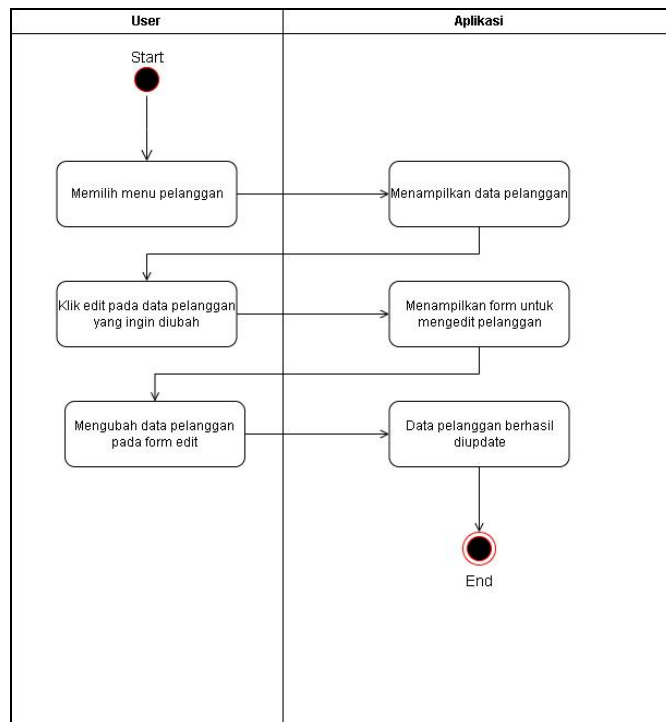


Gambar 3.12 Activity Diagram Tambah Data Pelanggan Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* yang dilakukan pertama kali oleh *user* adalah memilih menu pelanggan setelah diklik maka aplikasi akan menampilkan seluruh data pelanggan. Setelah data pelanggan tampil, *user* akan mengklik tombol tambah data lalu aplikasi akan menampilkan *form* tambah pelanggan, *user* akan meng-*inputkan* data pelanggan. Setelah data pelanggan di-*inputkan* maka data pelanggan berhasil ditambahkan.

I. Activity Diagram Edit Data Pelanggan yang diusulkan

Pada *activity diagram* edit data pelanggan yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mengedit data pelanggan yang sebelumnya sudah di-*inputkan* ke dalam aplikasi. *Activity diagram* edit data pelanggan yang diusulkan dapat dilihat pada Gambar 3.13.

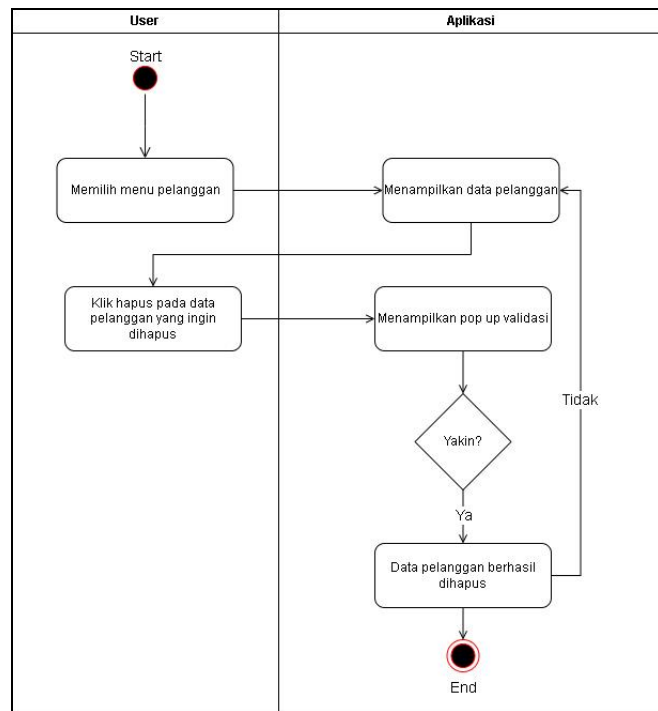


Gambar 3.13 Activity Diagram Edit Data Pelanggan Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* yang pertama kali dilakukan oleh *user* adalah memilih menu pelanggan setelah diklik maka aplikasi akan menampilkan seluruh data pelanggan. Setelah seluruh data pelanggan tampil, *user* akan memilih data yang akan di edit dan mengklik tombol edit data lalu aplikasi akan menampilkan *form* edit pelanggan, *user* akan mengedit data pelanggan sesuai dengan kebutuhan. Setelah mengisi *form* edit, maka data pelanggan berhasil di *update*.

J. Activity Diagram Hapus Data Pelanggan yang diusulkan

Pada *activity diagram* hapus data pelanggan yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam menghapus data pelanggan yang sebelumnya sudah di-*inputkan* ke dalam aplikasi. *Activity diagram* hapus data pelanggan yang diusulkan dapat dilihat pada Gambar 3.14.

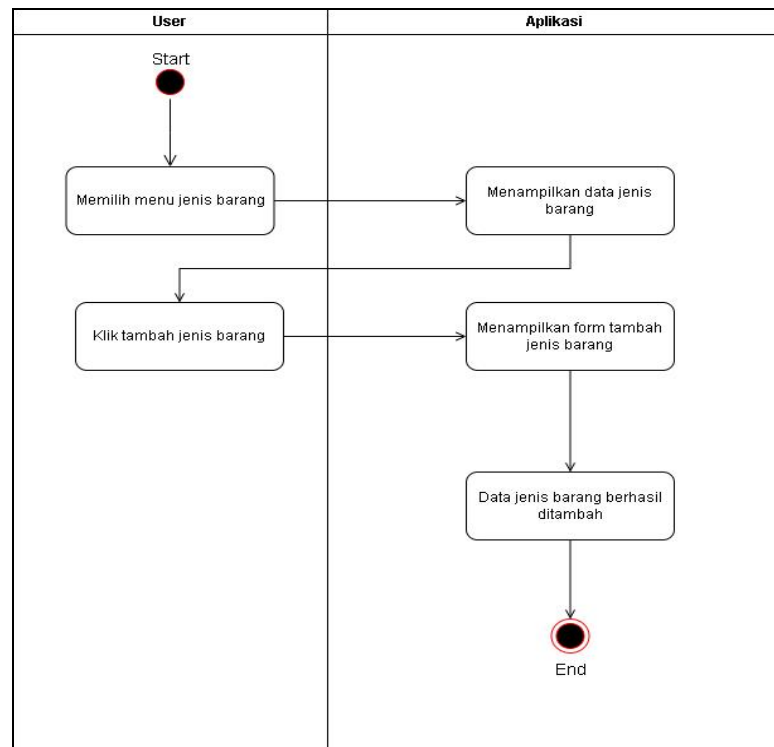


Gambar 3.14 Activity Diagram Hapus Data Pelanggan Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* yang pertama kali dilakukan *user* adalah memilih menu pelanggan setelah diklik maka aplikasi akan menampilkan seluruh data pelanggan. Setelah seluruh data pelanggan tampil, *user* akan memilih data yang akan dihapus dan mengklik tombol hapus data lalu aplikasi akan menampilkan *pop-up* validasi : ‘Yakin Ingin Menghapus Data?’, jika *user* memilih ya, maka aplikasi akan menghapus data tersebut, jika *user* memilih batal maka aplikasi akan kembali menampilkan data pelanggan.

K. Activity Diagram Tambah Data Jenis Barang yang diusulkan

Pada *activity diagram* tambah data jenis barang yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mengelola data jenis barang. *Activity diagram* tambah data jenis barang yang diusulkan dapat dilihat pada Gambar 3.15.

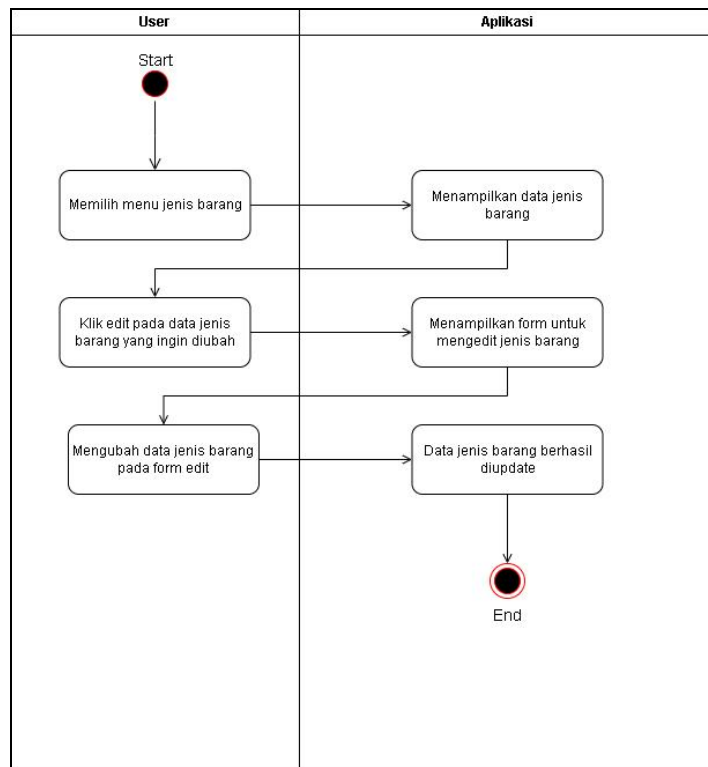


Gambar 3.15 Activity Diagram Tambah Data Jenis Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* yang pertama kali dilakukan oleh *user* adalah memilih menu jenis barang setelah diklik maka aplikasi akan menampilkan semua data jenis barang. Setelah semua data jenis barang tampil, *user* akan mengklik tombol tambah data lalu aplikasi akan menampilkan *form* tambah jenis barang, *user* akan meng-*input* data jenis barang. Setelah data jenis barang di-*input*kan maka data jenis barang berhasil ditambahkan.

L. Activity Diagram Edit Data Jenis Barang yang diusulkan

Pada *activity diagram* edit data jenis barang yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mengedit data jenis barang yang sebelumnya sudah di-*input*kan ke dalam aplikasi. *Activity diagram* edit data jenis barang yang diusulkan dapat dilihat pada Gambar 3.16.



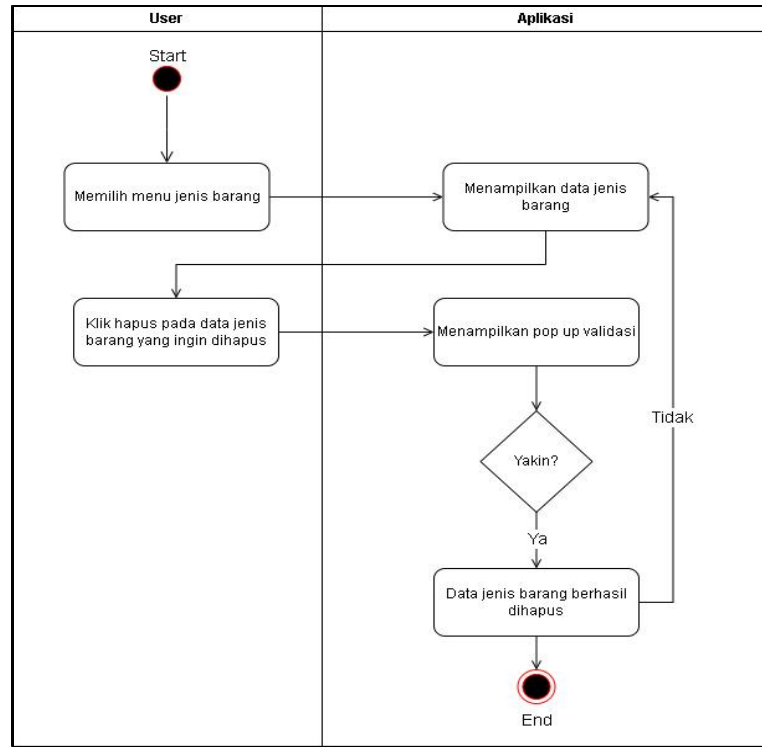
Gambar 3.16 Activity Diagram Edit Data Jenis Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* aktivitas yang dilakukan pertama kali oleh *user* adalah memilih menu jenis barang setelah diklik maka aplikasi akan menampilkan seluruh data jenis barang. Setelah seluruh data jenis barang tampil, *user* akan memilih data yang akan di edit dan mengklik tombol edit data lalu aplikasi akan menampilkan *form* edit, *user* akan mengedit data jenis barang sesuai dengan kebutuhan. Setelah mengisi *form* edit, maka data jenis barang berhasil di update.

M. Activity Diagram Hapus Data Jenis Barang yang diusulkan

Pada *activity diagram* hapus data jenis barang yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam menghapus data jenis barang yang sebelumnya telah di-*inputkan* ke dalam aplikasi.

Activity diagram hapus data jenis barang yang diusulkan dapat dilihat pada Gambar 3.17.

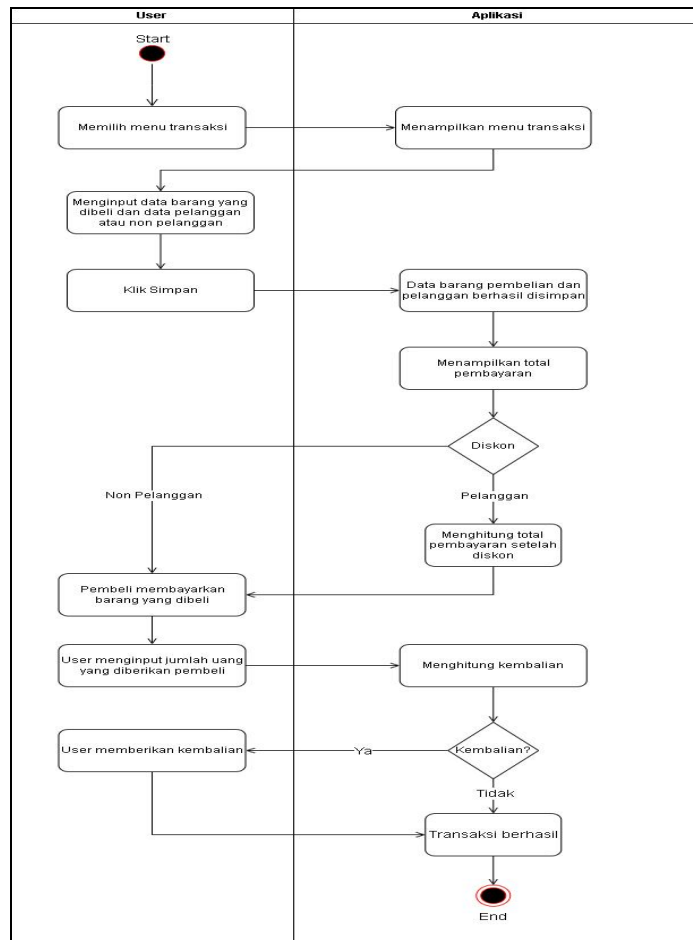


Gambar 3.17 Activity Diagram Hapus Data Jenis Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Pada activity diagram aktivitas yang pertama kali dilakukan oleh user adalah memilih menu jenis barang, setelah diklik maka aplikasi akan menampilkan seluruh data jenis barang. Setelah seluruh data jenis barang tampil, user akan memilih data yang akan dihapus dan mengklik tombol hapus data lalu aplikasi akan menampilkan pop-up validasi : ‘Yakin Ingin Menghapus Data?’, jika user memilih ya, maka aplikasi akan menghapus data tersebut, jika user memilih batal maka aplikasi akan kembali menampilkan data jenis barang.

N. Activity Diagram Transaksi yang diusulkan

Pada *activity diagram* transaksi yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam melakukan transaksi menggunakan aplikasi. *Activity diagram* transaksi yang diusulkan dapat dilihat pada Gambar 3.18.



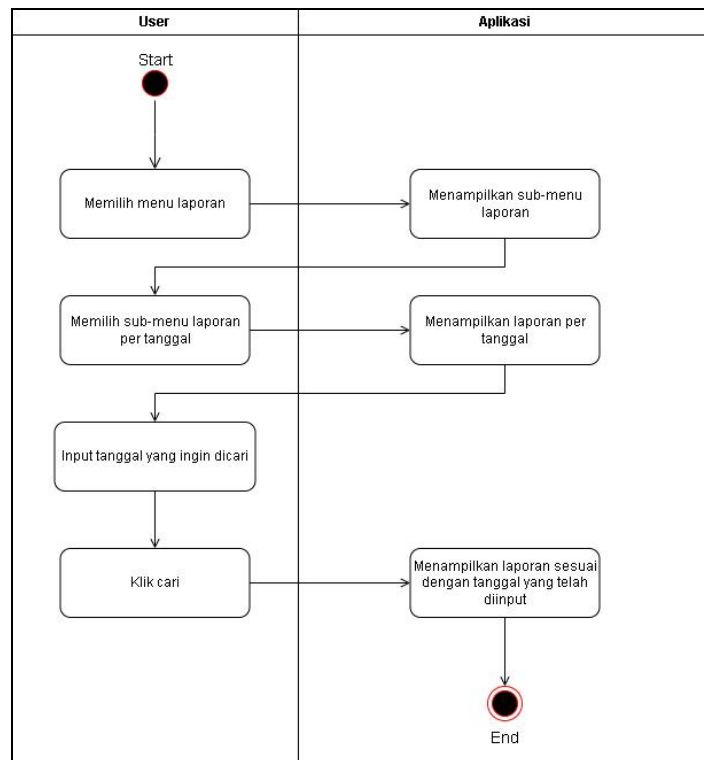
Gambar 3.18 Activity Diagram Transaksi Aplikasi Penjualan CV. Athala Lautan Mulia

Pada *activity diagram* aktivitas yang pertama kali dilakukan oleh user adalah memilih menu transaksi setelah diklik maka aplikasi akan menampilkan menu transaksi. Kemudian user menginputkan data barang yang dibeli beserta dengan apakah pembeli tersebut pelanggan atau non-pelanggan lalu klik simpan untuk memasukkan data yang telah diinput tadi kedalam data

transaksi. Setelah data tadi tersimpan aplikasi akan menampilkan jumlah total yang harus dibayarkan, jika pelanggan akan mendapatkan diskon dan aplikasi akan mulai menghitung diskon yang didapatkan oleh pembeli, jika tidak maka pembeli langsung membayarkan barang yang telah dibeli. Kemudian user menginputkan jumlah yang dibayarkan oleh pembeli, jika uang yang dibayarkan pembeli lebih dari jumlah yang harus dibayarkan maka user akan memberikan kembalian, jika tidak maka transaksi akan langsung selesai.

O. *Activity Diagram* Cari Laporan Per Tanggal yang diusulkan

Pada *activity diagram* cari laporan per tanggal yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mencari laporan per tanggal. *Activity diagram* cari laporan per tanggal yang diusulkan dapat dilihat pada gambar 3.19.

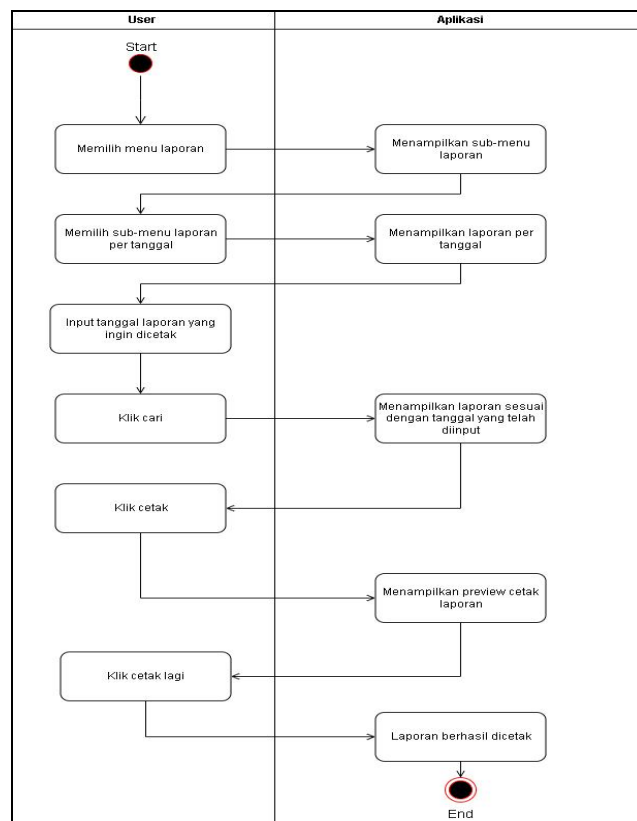


Gambar 3.19 *Activity Diagram* Cari Laporan Per Tanggal
Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* aktivitas yang pertama kali dilakukan oleh *user* adalah memilih menu laporan, setelah diklik maka aplikasi akan menampilkan *sub-menu* laporan. Kemudian *user* memilih *sub-menu* laporan per tanggal, setelah diklik maka aplikasi akan menampilkan seluruh data laporan per tanggal. Setelah seluruh data laporan per tanggal tampil, *user* akan menginput tanggal laporan yang ingin di cari lalu klik tombol cari, kemudian data pada tanggal yang telah diinputkan tadi akan tampil.

P. *Activity Diagram* Cetak Laporan Per Tanggal yang diusulkan

Pada *activity diagram* cetak laporan per tanggal yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mencetak laporan per tanggal. *Activity diagram* cetak laporan per tanggal yang diusulkan dapat dilihat pada Gambar 3.20.

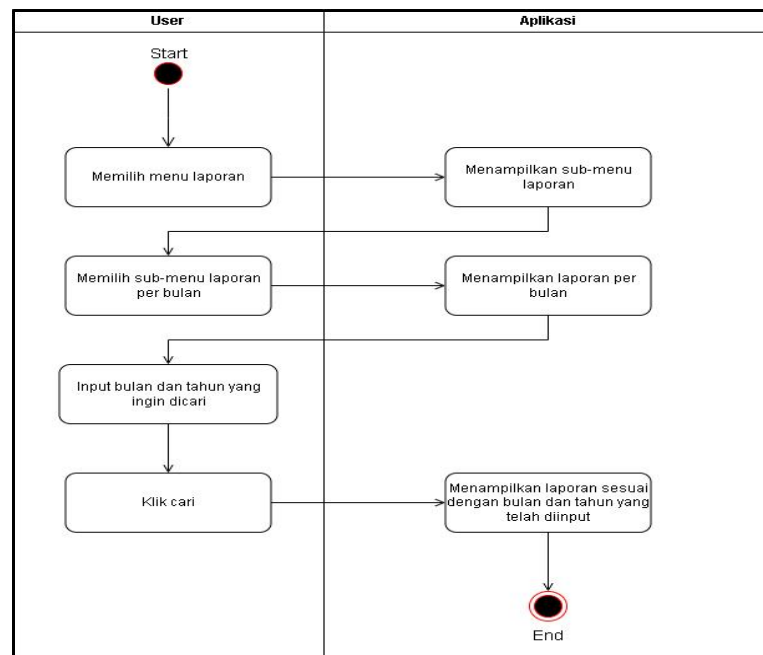


Gambar 3.20 *Activity Diagram* Cetak Laporan Per Tanggal Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* aktivitas yang pertama kali dilakukan oleh *user* adalah memilih menu laporan, setelah diklik maka aplikasi akan menampilkan *sub-menu* laporan. Kemudian *user* memilih *sub-menu* laporan per tanggal, setelah diklik maka aplikasi akan menampilkan seluruh data laporan per tanggal. Setelah seluruh data laporan per tanggal tampil, *user* akan menginput tanggal laporan yang ingin di cetak lalu klik tombol cari, kemudian data pada tanggal yang telah diinputkan tadi akan tampil. Kemudian klik tombol cetak lalu aplikasi akan menampilkan preview cetak. Setelah itu klik tombol cetak untuk mencetak laporan per tanggal, lalu laporan berhasil dicetak.

Q. *Activity Diagram* Cari Laporan Per Bulan yang diusulkan

Pada *activity diagram* cari laporan per bulan yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mencari laporan per bulan. *Activity diagram* cari laporan per tanggal yang diusulkan dapat dilihat pada Gambar 3.21.

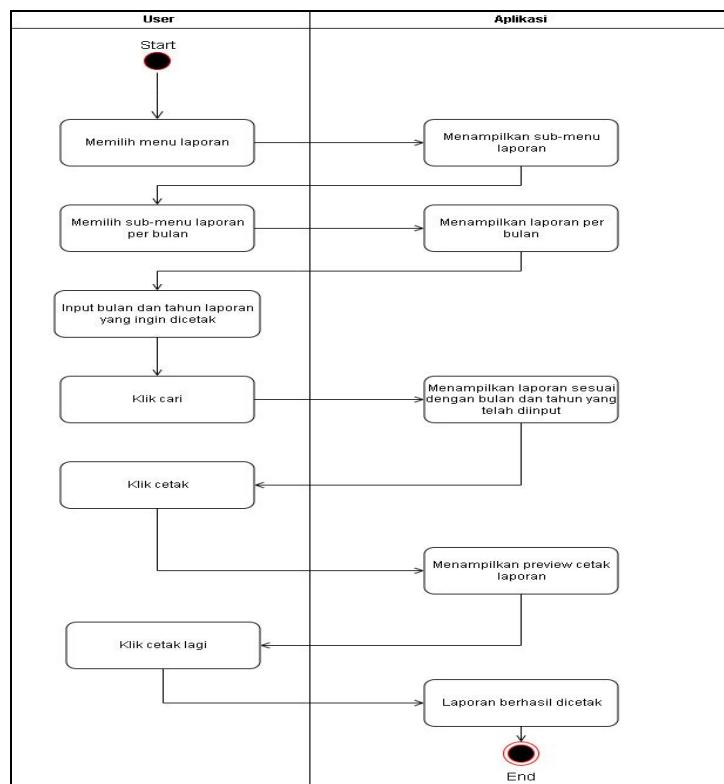


Gambar 3.21 *Activity Diagram* Cari Laporan Per Bulan Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* aktivitas yang pertama kali dilakukan oleh *user* adalah memilih menu laporan, setelah diklik maka aplikasi akan menampilkan *sub-menu* laporan. Kemudian *user* memilih *sub-menu* laporan per bulan, setelah diklik maka aplikasi akan menampilkan seluruh data laporan per bulan. Setelah seluruh data laporan per bulan tampil, *user* akan menginput bulan dan tahun laporan yang ingin di cari lalu klik tombol cari, kemudian data pada bulan yang telah diinputkan tadi akan tampil.

R. *Activity Diagram* Cetak Laporan Per Bulan yang diusulkan

Pada *activity diagram* cetak laporan per bulan yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mencetak laporan per tanggal. *Activity diagram* cetak laporan per bulan yang diusulkan dapat dilihat pada Gambar 3.22.

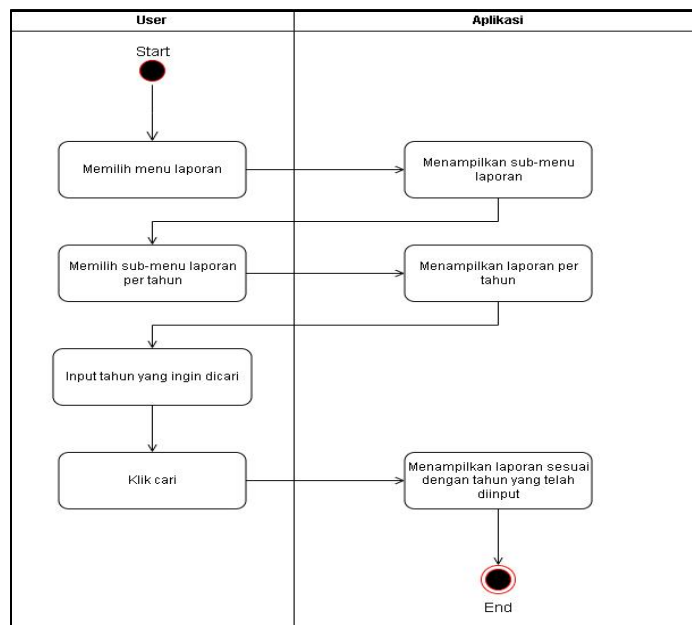


Gambar 3.22 *Activity Diagram* Cetak Laporan Per Bulan Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* aktivitas yang pertama kali dilakukan oleh *user* adalah memilih menu laporan, setelah diklik maka aplikasi akan menampilkan *sub-menu* laporan. Kemudian *user* memilih *sub-menu* laporan per bulan, setelah diklik maka aplikasi akan menampilkan seluruh data laporan per bulan. Setelah seluruh data laporan per bulan tampil, *user* akan menginput bulan dan tahun laporan yang ingin di cetak lalu klik tombol cari, kemudian data pada bulan dan tahun yang telah diinputkan tadi akan tampil. Kemudian klik tombol cetak lalu aplikasi akan menampilkan preview cetak. Setelah itu klik tombol cetak untuk mencetak laporan per bulan, lalu laporan berhasil dicetak.

S. *Activity Diagram* Cari Laporan Per Tahun yang diusulkan

Pada *activity diagram* cari laporan per bulan yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mencari laporan per bulan. *Activity diagram* cari laporan per tanggal yang diusulkan dapat dilihat pada Gambar 3.23.

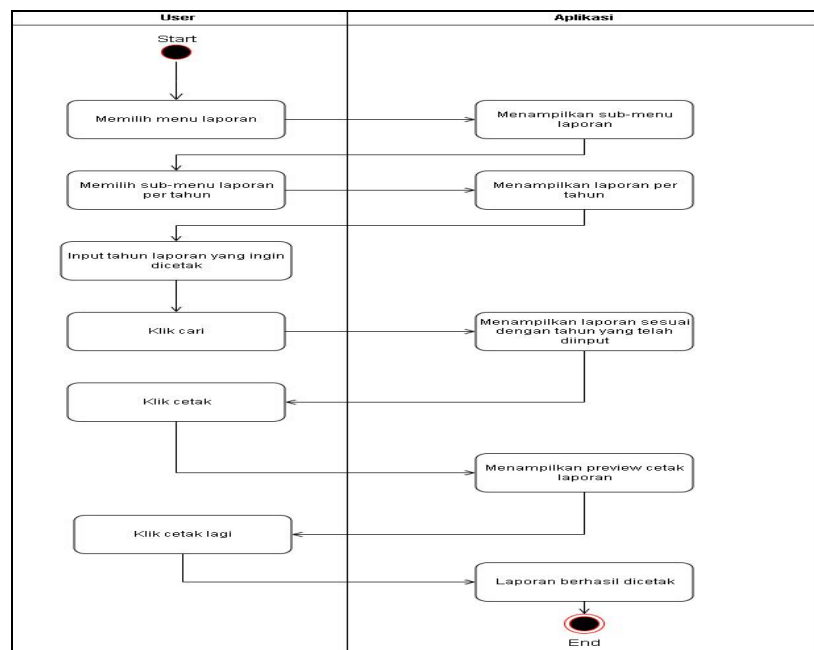


Gambar 3.23 *Activity Diagram* Cari Laporan Per Tahun Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* aktivitas yang pertama kali dilakukan oleh *user* adalah memilih menu laporan, setelah diklik maka aplikasi akan menampilkan *sub-menu* laporan. Kemudian *user* memilih *sub-menu* laporan per tahun, setelah diklik maka aplikasi akan menampilkan seluruh data laporan per tahun. Setelah seluruh data laporan per tahun tampil, *user* akan menginput tahun laporan yang ingin di cari lalu klik tombol cari, kemudian data pada tahun yang telah diinputkan tadi akan tampil.

T. *Activity Diagram* Cetak Laporan Per Tahun yang diusulkan

Pada *activity diagram* cetak laporan per bulan yang diusulkan akan dijelaskan mengenai aktivitas *user* dalam mencetak laporan per tanggal. *Activity diagram* cetak laporan per bulan yang diusulkan dapat dilihat pada Gambar 3.24.



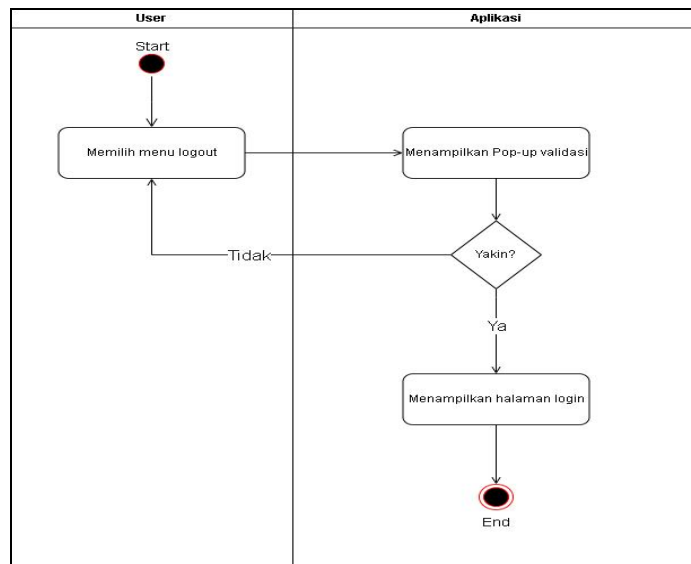
Gambar 3.24 *Activity Diagram* Cetak Laporan Per Tahun Aplikasi Penjualan CV.Athala Lautan Mulia

Pada *activity diagram* aktivitas yang pertama kali dilakukan oleh *user* adalah memilih menu laporan, setelah diklik maka aplikasi

akan menampilkan *sub-menu* laporan. Kemudian *user* memilih *sub-menu* laporan per bulan, setelah diklik maka aplikasi akan menampilkan seluruh data laporan per bulan. Setelah seluruh data laporan per bulan tampil, *user* akan menginput bulan dan tahun laporan yang ingin di cetak lalu klik tombol cari, kemudian data pada bulan dan tahun yang telah diinputkan tadi akan tampil. Kemudian klik tombol cetak lalu aplikasi akan menampilkan preview cetak. Setelah itu klik tombol cetak untuk mencetak laporan per bulan, lalu laporan berhasil dicetak.

U. *Activity Diagram Logout yang diusulkan*

Pada *activity diagram logout* yang diusulkan akan dijelaskan mengenai aktivitas yang dilakukan oleh *user* ketika akan *logout* dari aplikasi. *Activity diagram logout* yang diusulkan dapat dilihat pada Gambar 3.25.






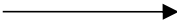

**Gambar 3.25 *Activity Diagram Logout* Aplikasi Penjualan
CV.Athala Lautan Mulia**

3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

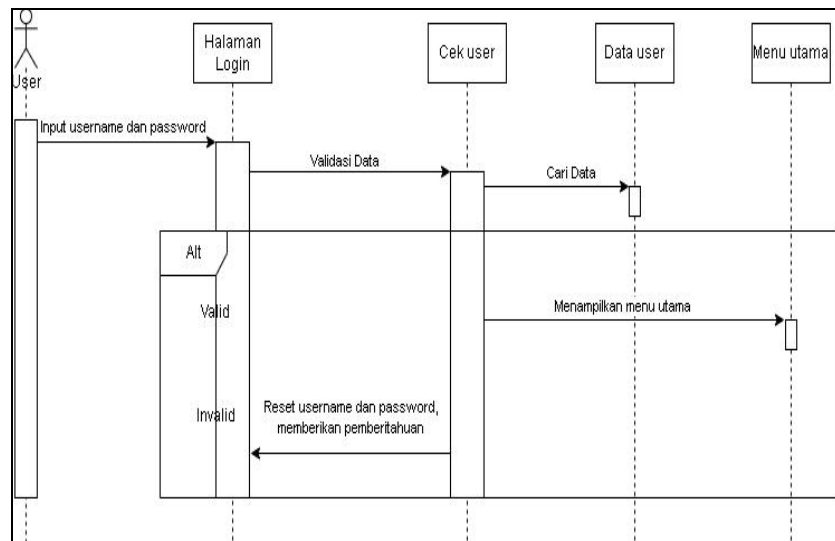
Simbol-simbol yang ada pada *sequence diagram* dapat dilihat pada tabel 3.4 dibawah ini :

Tabel 3.4 Simbol-simbol *Sequence Diagram*

| Simbol | Keterangan |
|---|--|
|  | <i>Actor</i> : orang, proses atau sistem yang berinteraksi dengan sistem. |
|  | <i>Lifeline</i> : garis titik – titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> . |
|  | <i>Object</i> : Menyatakan objek yang berinteraksi pesan |
|  | <i>Message</i> : symbol mengirim pesan antar <i>class</i> . |
|  | <i>Activation</i> : mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi. |

A. *Sequence Diagram Login* yang diusulkan

Pada *sequence diagram* yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* akan *login* ke aplikasi. *Sequence diagram login* yang diusulkan dapat dilihat pada Gambar 3.26.

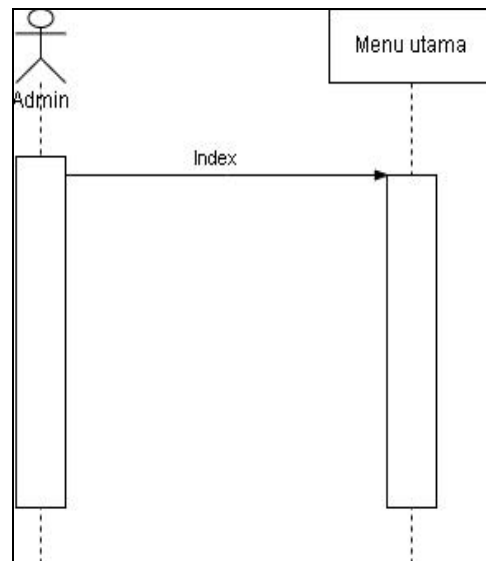


Gambar 3.26 Sequence Diagram Login Aplikasi Penjualan CV. Athala Lautan Mulia

Berdasarkan Gambar 3.26 dapat dilihat bahwa proses yang terjadi saat akan masuk ke dalam aplikasi, *username* dan *password user* akan di validasi terlebih dahulu, jika valid maka aplikasi akan menampilkan menu utama, jika tidak valid maka aplikasi akan mereset *username* dan *password* pada halaman *login* dan memberikan pemberitahuan bahwa *username* dan *password* yang dimasukkan salah.

B. Sequence Diagram Menu Utama yang diusulkan

Pada *sequence diagram* menu utama yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* berhasil *login* ke aplikasi. Menu utama yang berisi nama CV. Athala Lautan Mulia beserta dengan *sub-menu*. *Sequence diagram* menu utama yang diusulkan dapat dilihat pada Gambar 3.27.

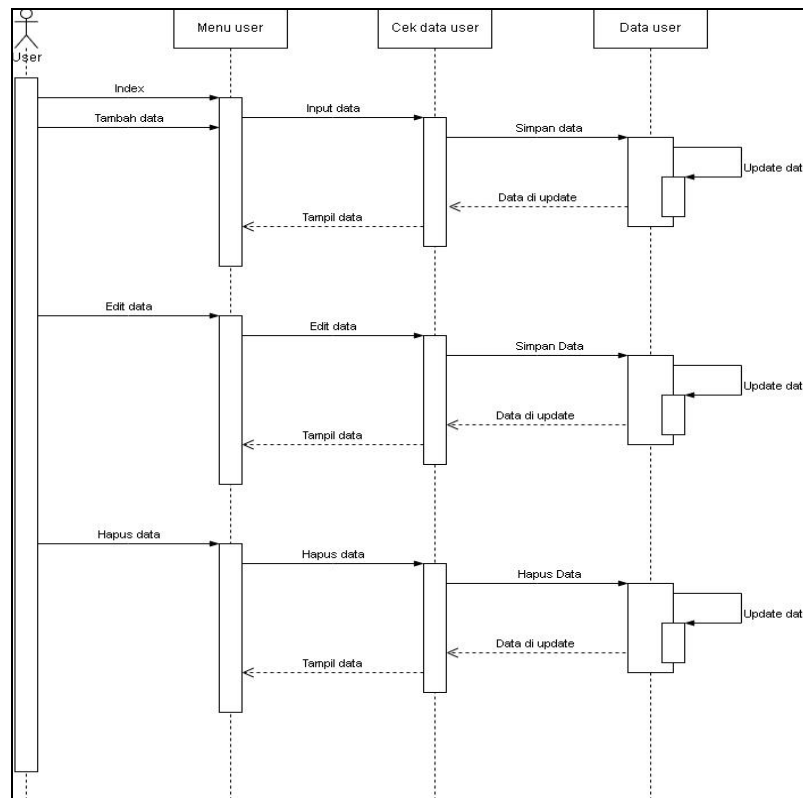


Gambar 3.27 Sequence Diagram Menu Utama Aplikasi Penjualan CV.Athala Lautan Mulia

Berdasarkan Gambar 3.27 dapat dilihat bahwa proses yang terjadi ketika user berhasil login, aplikasi akan menampilkan index dari menu utama. Index menu utama terdiri dari tulisan CV. Athala Lautan Mulia dan sub-menu seperti menu user, menu jenis barang, menu barang, menu pelanggan, menu transaksi, menu laporan, dan logout.

C. Sequence Diagram User yang diusulkan

Pada *sequence diagram user* yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* memilih menu *user*. Pada menu *user*, *user* dapat mengolah data *user*, yaitu dengan tambah *user*, edit *user*, dan hapus *user*. *Sequence diagram user* yang diusulkan dapat dilihat pada Gambar 3.28.

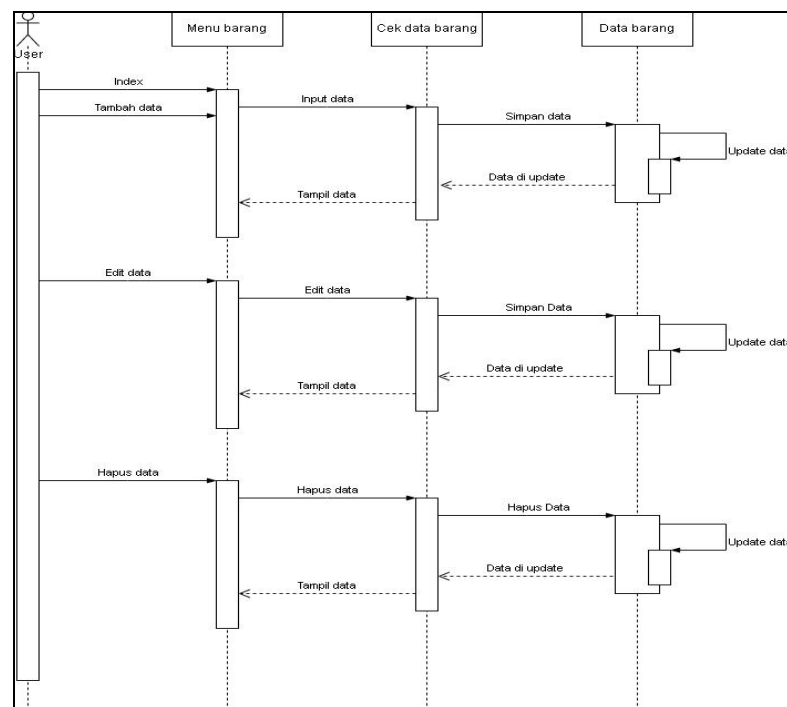


**Gambar 3.28 Sequence Diagram User Aplikasi Penjualan
CV.Athala Lautan Mulia**

Berdasarkan Gambar 3.28 dapat dilihat bahwa pada menu *user*, yang akan ditampilkan pertama kali adalah index yang berisi data *user*, *user* dapat mengolah data *user* dengan melakukan tambah *user*, edit *user*, dan hapus *user*. Pada pilihan tambah *user*, jika sudah mengisi *form* tambah *user*, maka aplikasi akan menyimpan data tersebut ke *database* dan akan memperbarui data yang akan ditampilkan pada menu *user*. Pada pilihan edit *user*, *user* dapat mengedit data sesuai dengan kebutuhan, akan diperbarui di *database*, yang kemudian akan ditampilkan di menu *user*. Pada pilihan hapus *user*, *user* dapat menghapus data jika diperlukan, *user* akan memilih data mana yang ingin dihapus, setelah itu maka data akan dihapus dari *database*, dan akan diperbarui pada menu *user*.

D. Sequence Diagram Barang yang diusulkan

Pada *sequence diagram* barang yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* memilih menu barang. Pada menu barang, *user* dapat mengolah data barang, yaitu dengan tambah barang, edit barang, dan hapus barang. *Sequence diagram* barang yang diusulkan dapat dilihat pada Gambar 3.29.



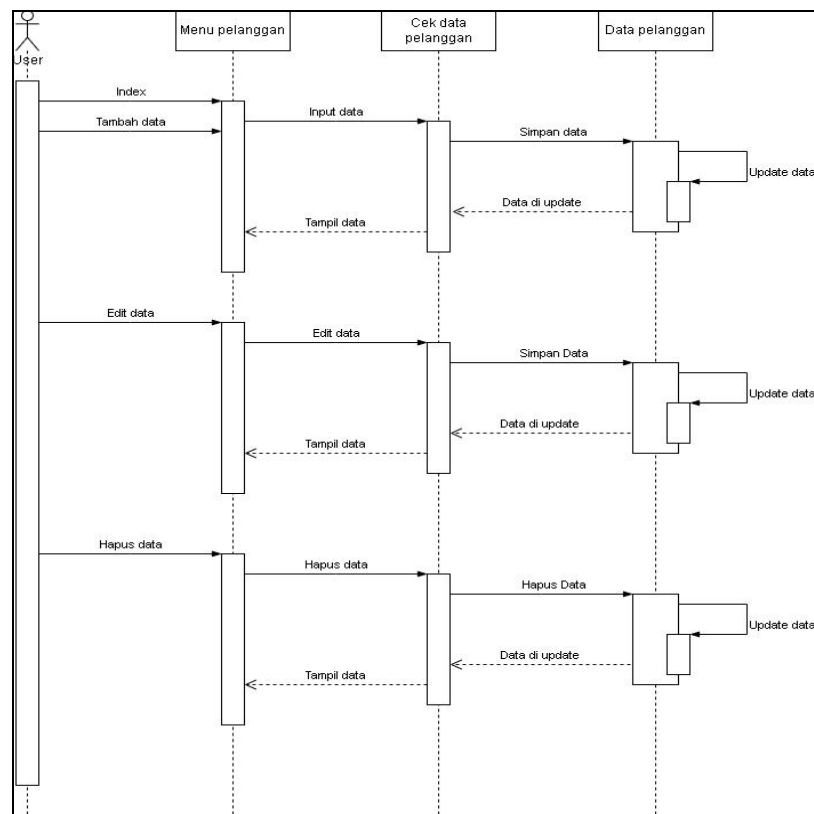
Gambar 3.29 Sequence Diagram Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Berdasarkan Gambar 3.29 dapat dilihat bahwa pada menu barang, yang akan ditampilkan pertama kali adalah index yang berisi data barang, *user* dapat mengolah data barang dengan melakukan tambah barang, edit barang, dan hapus barang. Pada pilihan tambah barang, jika sudah mengisi *form* tambah barang, maka aplikasi akan menyimpan data tersebut ke *database* dan akan memperbarui data yang akan ditampilkan pada menu barang. Pada pilihan edit barang, *user* dapat mengedit data sesuai dengan kebutuhan, akan diperbarui di *database*, yang kemudian akan

ditampilkan di menu barang. Pada pilihan hapus barang, *user* dapat menghapus data jika diperlukan, *user* akan memilih data mana yang ingin dihapus, setelah itu maka data akan dihapus dari *database*, dan akan diperbarui pada menu barang.

E. *Sequence Diagram* Pelanggan yang diusulkan

Pada *sequence diagram* pelanggan yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* memilih menu pelanggan. Pada menu pelanggan, *user* dapat mengolah data pelanggan, yaitu dengan tambah pelanggan, edit pelanggan, dan hapus pelanggan. *Sequence diagram* pelanggan yang diusulkan dapat dilihat pada Gambar 3.30.

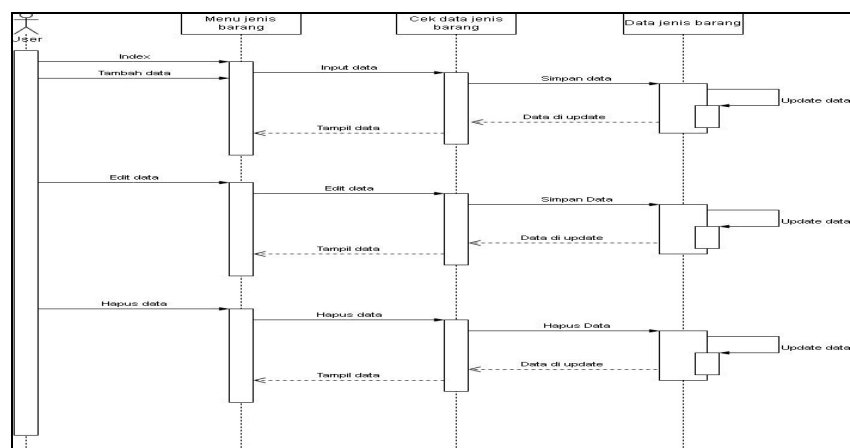


Gambar 3.30 *Sequence Diagram* Pelanggan Aplikasi Penjualan CV.Athala Lautan Mulia

Berdasarkan Gambar 3.30 dapat dilihat bahwa pada menu pelanggan yang akan ditampilkan pertama kali adalah index yang berisi data pelanggan, *user* dapat mengolah data pelanggan dengan melakukan tambah pelanggan, edit pelanggan, dan hapus pelanggan. Pada pilihan tambah pelanggan, jika sudah mengisi *form* tambah pelanggan, maka aplikasi akan menyimpan data tersebut ke *database* dan akan memperbarui data yang akan ditampilkan pada menu pelanggan. Pada pilihan edit pelanggan, *user* dapat mengedit data sesuai dengan kebutuhan, akan diperbarui di *database*, yang kemudian akan ditampilkan di menu pelanggan. Pada pilihan hapus pelanggan, *user* dapat menghapus data jika diperlukan, *user* akan memilih data mana yang ingin dihapus, setelah itu maka data akan dihapus dari *database*, dan akan diperbarui pada menu pelanggan.

F. *Sequence Diagram* Jenis Barang yang diusulkan

Pada *sequence diagram* jenis barang yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* memilih menu jenis barang. Pada menu jenis barang, *user* dapat mengolah data jenis barang, yaitu dengan tambah jenis barang, edit jenis barang, dan hapus jenis barang. *Sequence diagram* jenis barang yang diusulkan dapat dilihat pada Gambar 3.31.

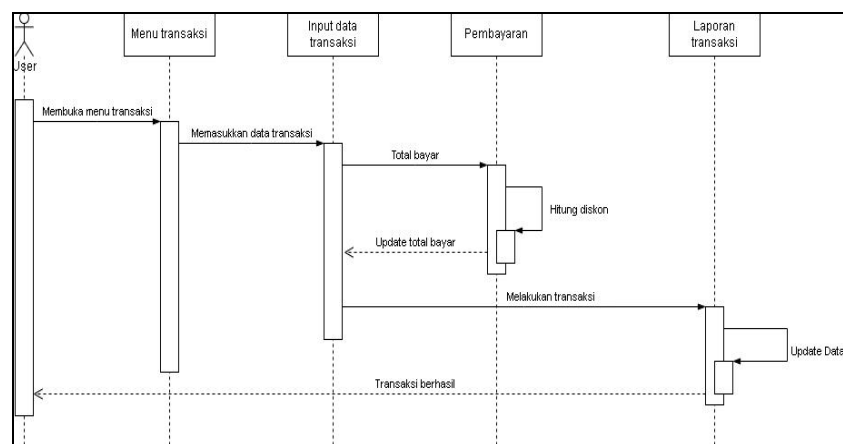


Gambar 3.31 *Sequence Diagram* Jenis Barang Aplikasi Penjualan CV.Athala Lautan Mulia

Berdasarkan Gambar 3.31 dapat dilihat bahwa pada menu jenis barang yang akan ditampilkan pertama kali adalah index yang berisi data jenis barang, *user* dapat mengolah data jenis barang dengan melakukan tambah jenis barang, edit jenis barang, dan hapus jenis barang. Pada pilihan tambah jenis barang, jika sudah mengisi *form* tambah jenis barang, maka aplikasi akan menyimpan data tersebut ke *database* dan akan memperbarui data yang akan ditampilkan pada menu jenis barang. Pada pilihan edit jenis barang, *user* dapat mengedit data sesuai dengan kebutuhan, akan diperbarui di *database*, yang kemudian akan ditampilkan di menu jenis barang. Pada pilihan hapus jenis barang, *user* dapat menghapus data jika diperlukan, *user* akan memilih data mana yang ingin dihapus, setelah itu maka data akan dihapus dari *database*, dan akan diperbarui pada menu jenis barang.

G. *Sequence Diagram* Transaksi yang diusulkan

Pada *sequence diagram* transaksi yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* memilih menu transaksi. Pada menu transaksi, *user* dapat melakukan transaksi dengan pembeli terkait barang yang dibeli. *Sequence diagram* transaksi yang diusulkan dapat dilihat pada Gambar 3.32.

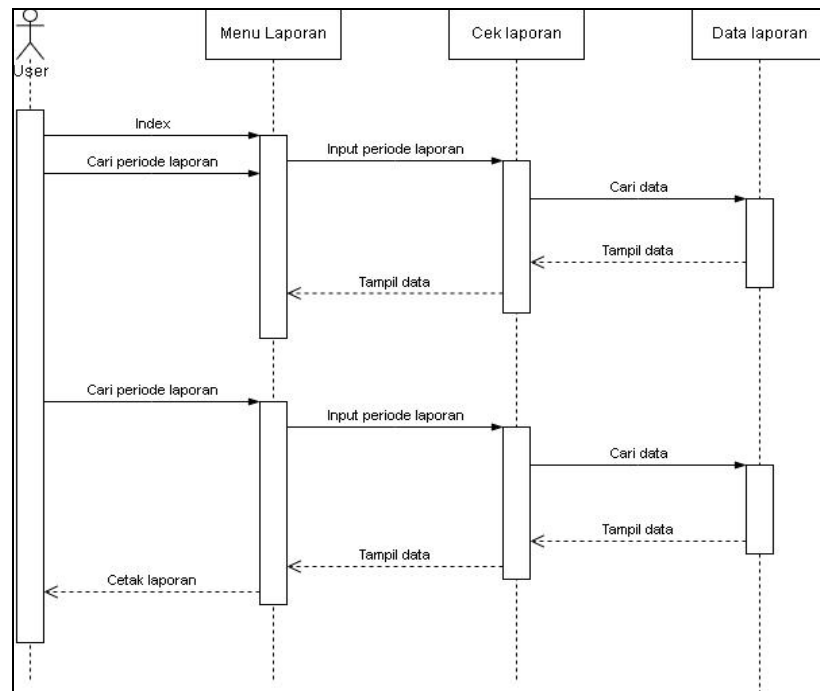


Gambar 3.32 *Sequence Diagram* Transaksi Aplikasi Penjualan CV. Athala Lautan Mulia

Berdasarkan Gambar 3.32 dapat dilihat bahwa untuk melakukan transaksi yang pertama dilakukan oleh *user* adalah membuka menu transaksi lalu *user* menginputkan data transaksi. Setelah itu aplikasi akan menampilkan total harga yang harus dibayarkan, jika pelanggan maka total harga akan mendapatkan diskon, jika non-pelanggan tidak mendapatkan diskon dan langsung melakukan transaksi, setelah itu transaksi berhasil data transaksi akan di update ke dalam *database* laporan.

H. *Sequence Diagram* Laporan yang diusulkan

Pada *sequence diagram* Laporan yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* memilih menu laporan. Pada menu laporan, *user* dapat melakukan pencarian dan pencetakan laporan. *Sequence diagram* transaksi yang diusulkan dapat dilihat pada Gambar 3.33.

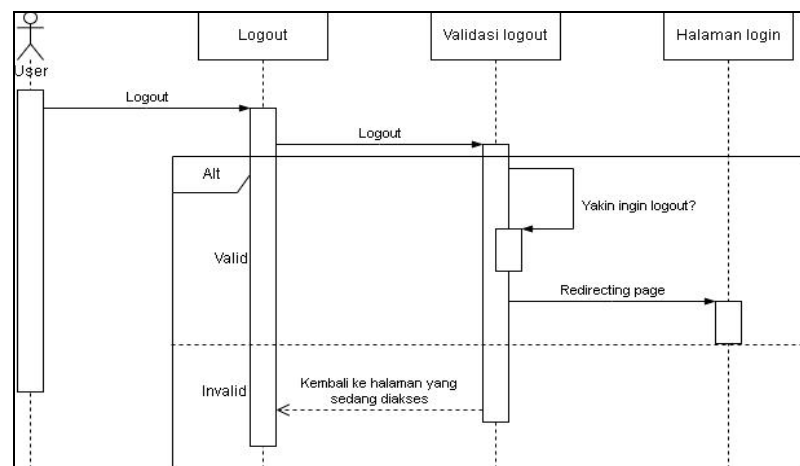


**Gambar 3.33 Sequence Diagram Laporan Aplikasi Penjualan
CV.Athala Lautan Mulia**

Berdasarkan Gambar 3.33 yang pertama kali ditampilkan aplikasi adalah semua data laporan. *User* dapat mencari periode laporan sesuai dengan kebutuhan dan dapat mencetak laporan tersebut. Ketika *user* hendak mencari laporan transaksi yang ada sesuai dengan periode yang dibutuhkan, *user* akan memasukkan periode mulai dari tanggal, bulan dan tahun awal sampai dengan tanggal, bulan dan tahun akhir laporan yang ingin dicari, kemudian aplikasi akan mencari data laporan tersebut, dan menampilkannya pada menu laporan. Jika *user* ingin mencetak maka *user* harus terlebih dahulu mencari periode laporan yang ingin dicetak, setelah laporan yang ingin dicetak muncul maka *user* bisa langsung mencetak laporan tersebut.

I. *Sequence Diagram Logout* yang diusulkan

Pada *sequence diagram* yang diusulkan akan dijelaskan mengenai proses yang terjadi ketika *user* akan *login* ke aplikasi. *Sequence diagram login* yang diusulkan dapat dilihat pada Gambar 3.34.



Gambar 3.34 *Sequence Diagram Logout* Aplikasi Penjualan CV.Athala Lautan Mulia

Berdasarkan Gambar 3.34 dapat dilihat bahwa proses yang terjadi saat akan keluar dari aplikasi, aplikasi akan menampilkan *pop-up*, apakah yakin ingin *logout?*, jika valid maka aplikasi akan memproses dan kemudian menampilkan halaman *login*, jika tidak valid maka aplikasi akan kembali menampilkan menu yang sedang diakses oleh *user*.

3.2.2. RAD Design Workshop

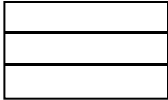


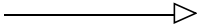

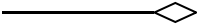
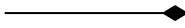
Pada fase ini, dilakukan diskusi terkait dengan aplikasi yang telah dibuat dengan meminta tanggapan dari pembimbing lapangan di CV.Athala Lautan Mulia. Bertujuan agar aplikasi yang telah dirancang dan dibuat sesuai dengan apa yang dibutuhkan oleh CV.Athala Lautan Mulia

A. Pemodelan Data Sistem

Pemodelan data sistem pada pembangunan aplikasi ini menggunakan UML *Class Diagram*. *Class diagram* merupakan hubungan antar kelas di dalam model desain suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* meliputi : kelas (*class*), relasi *Associations*, *Generalitation* dan *Aggregation*, atribut (*attributes*), operasi (*operation/method*), dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality*.

Simbo-simbol yang terdapat pada Class diagram bisa dilihat pada tabel dibawah ini.

Tabel 3.5 Simbol simbol *Class Diagram*

| Simbol | Keterangan |
|---|---|
|  | Kelas : kelas pada struktur sistem |
|  | Antarmuka : sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek |
|  | Asosiasi : relasi antarkelas dengan makna umum |
|  | Generalisasi : relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus) |
|  | Kebergantungan : relasi antarkelas dengan makna kebergantungan kelas |
|  | Agregasi : relasi antarkelas dengan makna semua-bagian |
|  | Komposisi : sebuah kelas tidak bisa berdiri sendiri dan harus merupakan bagian dari kelas yang lain, maka kelas memiliki relasi komposisi terhadap kelas tempat dia bergantung tersebut |

Simbol *Multiplicity* yang digunakan dalam *class diagram* bisa dilihat pada tabel dibawah ini.

Tabel 3.6 Simbol-simbol *Multiplicity Class Diagram*

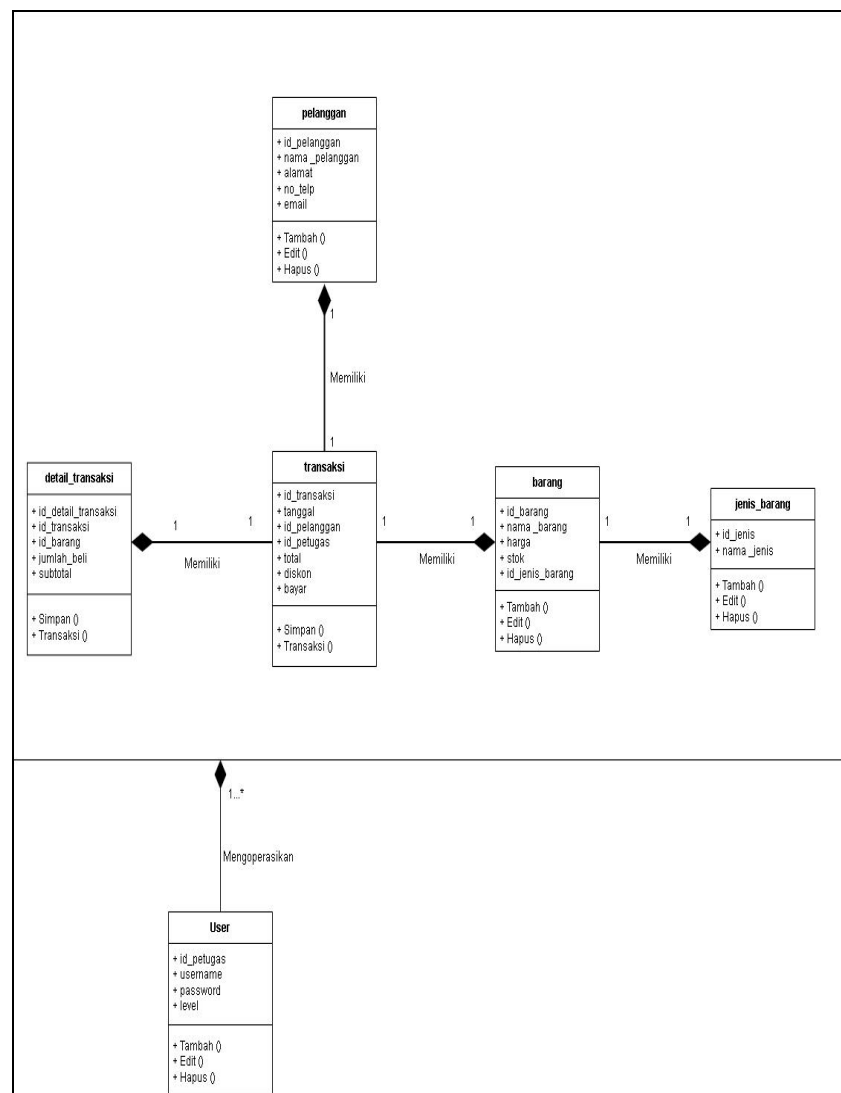
| Simbol | Keterangan |
|--------|-----------------------------------|
| 1 | Satu dan hanya satu |
| 0...* | Boleh tidak ada atau 1 atau lebih |
| 0...1 | Boleh tidak ada maksimal 1 |

Tabel 3.6 Simbol-simbol *Multiplicity Class Diagram* (Lanjutan)

| Simbol | Keterangan |
|--------|--|
| n...n | Batasan antara. Contoh 1...3 mempunyai arti minimal 1 maksimal 3 |

A. *Class Diagram* yang diusulkan

Pada *class diagram* yang diusulkan akan dijelaskan mengenai relasi yang dilakukan antara entitas ke sistem. *Class diagram* yang diusulkan dapat dilihat pada Gambar 3.35.



Gambar 3.35 *Class Diagram* Aplikasi Penjualan CV.Athala Lautan Mulia

Berdasarkan Gambar 3.35 *class diagram* memiliki 6 entitas yaitu *user*, jenis barang, barang, pelanggan, transaksi, dan laporan. *Class diagram* memiliki tiga level *user* yaitu Admin, Manager, Karyawan. Level admin bisa mengakses dan mengelola seluruh menu beserta data yang ada pada aplikasi, level manager hanya bisa mengakses dan mengelola data pada menu laporan, level karyawan hanya bisa mengakses menu transaksi untuk melakukan transaksi jika ada pembelian dari pembeli. Pada *class diagram* ini barang dapat dikelompokkan melalui jenis barangnya. Menu transaksi memiliki akses untuk mengambil data barang dan data pelanggan serta menu transaksi akan secara otomatis mengupload data ke dalam laporan jika ada transaksi yang terjadi. Menu laporan digunakan untuk melihat histori transaksi yang terjadi.

B. Spesifikasi Database

1. Spesifikasi Tabel Petugas

Nama tabel : petugas

Primary Key : id_petugas

Foreign Key :

Tabel 3.7 Spesifikasi Tabel Petugas

| No | Nama Field | Tipe | Panjang |
|----|------------|---------|---------|
| 1 | id_petugas | varchar | 7 |
| 2 | username | varchar | 200 |
| 3 | password | varchar | 200 |
| 4 | level | varchar | 15 |

2. Spesifikasi Tabel Transaksi

Nama tabel : transaksi

Primary Key : id_transaksi

Foreign Key : id_petugas

Tabel 3.8 Spesifikasi Tabel Transaksi

| No | Nama Field | Tipe | Panjang |
|----|--------------|---------|---------|
| 1 | id_transaksi | varchar | 8 |
| 2 | id_pelanggan | varchar | 20 |
| 3 | tanggal | date | |
| 4 | id_petugas | varchar | 7 |
| 5 | total | int | 12 |
| 6 | diskon | int | 12 |
| 7 | bayar | int | 12 |

3. Spesifikasi Tabel Detail Transaksi

Nama tabel : detail_transaksi

Primary Key : id_detail_transaksi

Foreign Key : id_transaksi

Tabel 3.9 Spesifikasi Tabel Detail Transaksi

| No | Nama Field | Tipe | Panjang |
|----|---------------------|---------|---------|
| 1 | id_detail_transaksi | int | 8 |
| 2 | id_transaksi | varchar | 8 |
| 3 | id_barang | varchar | 8 |
| 4 | jumlah_beli | int | 10 |
| 5 | subtotal | int | 12 |

4. Spesifikasi Tabel Jenis Barang

Nama tabel : jenis_barang

Primary Key : id_jenis_barang

Foreign Key :

Tabel 3.10 Spesifikasi Tabel Jenis Barang

| No | Nama Field | Tipe | Panjang |
|----|-----------------|---------|---------|
| 1 | id_jenis_barang | varchar | 8 |
| 2 | nama_jenis | varchar | 40 |

5. Spesifikasi Tabel Barang

Nama tabel : barang

Primary Key : id_barang

Foreign Key : id_jenis_barang

Tabel 3.11 Spesifikasi Tabel Barang

| No | Nama Field | Tipe | Panjang |
|----|-----------------|---------|---------|
| 1 | id_barang | varchar | 10 |
| 2 | nama_barang | varchar | 150 |
| 3 | harga | int | 50 |
| 4 | stok | int | 50 |
| 5 | id_jenis_barang | varchar | 8 |

6. Spesifikasi Tabel Pelanggan

Nama tabel : pelanggan

Primary Key : id_pelanggan

Foreign Key :

Tabel 3.12 Spesifikasi Tabel Pelanggan


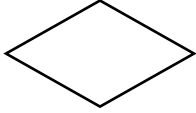
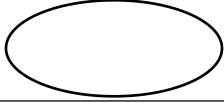

| No | Nama Field | Tipe | Panjang |
|----|--------------|---------|---------|
| 1 | id_pelanggan | varchar | 10 |
| 2 | nama | varchar | 30 |
| 3 | alamat | varchar | 35 |
| 4 | no_telp | varchar | 15 |
| 5 | email | varchar | 40 |

C. Entity Relationship Diagram (ERD)

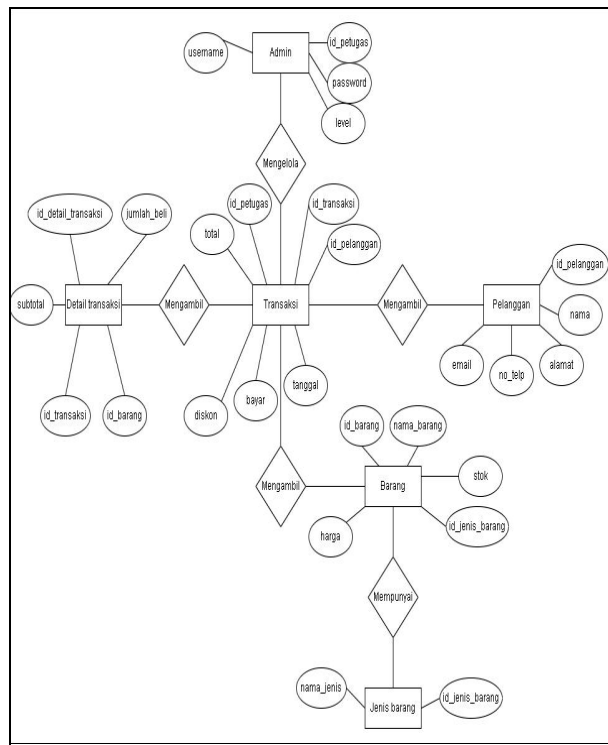
Entity Relationship Diagram (ERD) adalah notasi yang digunakan untuk melakukan aktivitas pemodelan data. Tujuan utama dari ERD adalah mewakili objek data. ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan suatu model jaringan (*network*) yang menggunakan susunan data yang disimpan dari sistem secara baik atau merupakan konseptual yang mendeskripsikan hubungan antar data *store* dalam DFD (Pressman, 2002).

Simbol-simbol yang terdapat pada ERD dapat dilihat pada tabel dibawah ini.

Tabel 3.13 Simbol-Simbol ERD

| Simbol | Keterangan |
|---|--|
|  | <i>Entity</i> : suatu objek yang ada pada dunia nyata dan dapat dibedakan dari objek lainnya yang didefinisikan secara unik. Entitas dapat berupa lingkungan elemen, <i>resource</i> , atau suatu transaksi yang sangat penting. Entitas disimbolkan dengan persegi panjang. |
|  | <i>Relationship</i> : hubungan yang terjadi antara satu entitas atau lebih dan digambarkan dengan suatu prisma yang diberi label berbentuk kata kerja. |
|  | <i>Attribute</i> : menyatakan atribut |
|  | <i>Connection</i> : yaitu garis penghubung antar himpunan relasi dengan himpunan entitas dengan atribut. |

➤ **Entity Relationship Diagram (ERD) yang diusulkan**

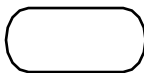


Gambar 3.36 ERD Admin Aplikasi Sistem Penjualan CV.Athala Lautan Mulia


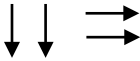

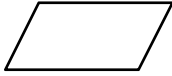

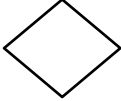
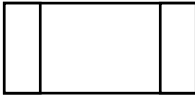
D. Perancangan *Flowchart* Program

Bagan alir (*flowchart*) merupakan sebuah bagan yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika (Jogiyanto, 2005). Bagan alir digunakan untuk alat bantu komunikasi dan untuk dokumentasi. Sedangkan *flowchart* program adalah bagan yang menjelaskan secara detail langkah-langkah dari proses program. Simbol-simbol yang digunakan dalam merancang *flowchart* dapat dilihat pada tabel dibawah ini.

Tabel 3.14 Simbol-simbol *Flowchart*

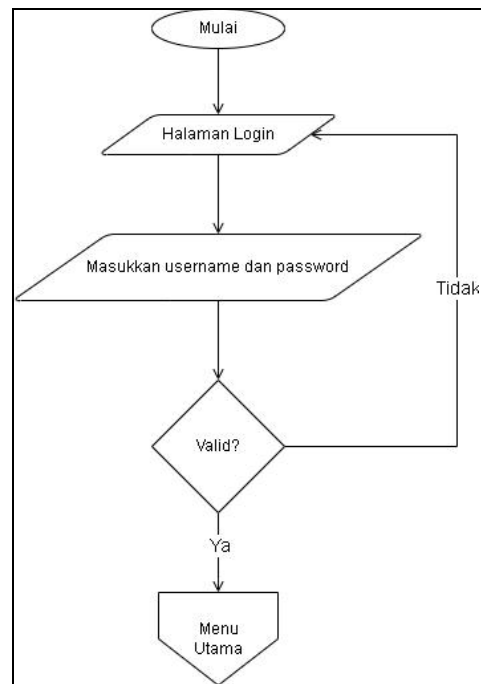
| Simbol | Nama | Fungsi |
|---|---------------|--|
|  | Mulai/selesai | Menunjukkan mulai atau selesainya suatu proses |

Tabel 3.14 Simbol-simbol *Flowchart* (Lanjutan)

| Simbol | Nama | Fungsi |
|---|---------------------------|---|
|  | Proses | Menunjukkan kegiatan proses dari operasi aplikasi/sistem |
|  | Garis Alir | Menunjukkan arus dari proses |
|  | Simbol Penghubung | Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus |
|  | <i>Input/Output</i> | Digunakan untuk mewakili data <i>input/output</i> |
|  | Simbol Persiapan | Digunakan untuk memberi nilai awal suatu besaran |
|  | Keputusan | Digunakan untuk suatu penyeleksian kondisi di dalam suatu program |
|  | <i>Predefined Process</i> | Menyatakan bahwa terdapat proses lagi didalamnya |

A. Perancangan Flowchart *Login*

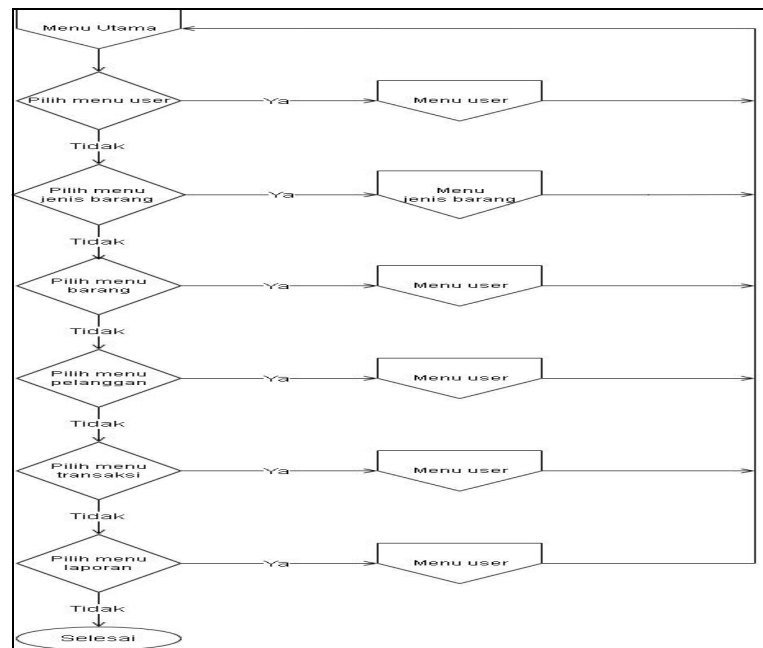
Pada *flowchart login* ditampilkan proses untuk *login*/masuk ke aplikasi. Sebelum masuk ke aplikasi, sistem akan melakukan proses validasi terhadap *username* dan *password* yang di-input oleh *user*, apakah sudah valid sesuai dengan data yang ada pada *database*. Jika valid maka *user* dapat masuk ke aplikasi, jika tidak valid maka *user* harus mengulang pengisian *username* dan *password*. *Flowchart login* dapat dilihat pada Gambar 3.37.



Gambar 3.37 *Flowchart Login*

B. Perancangan *Flowchart* Menu Utama

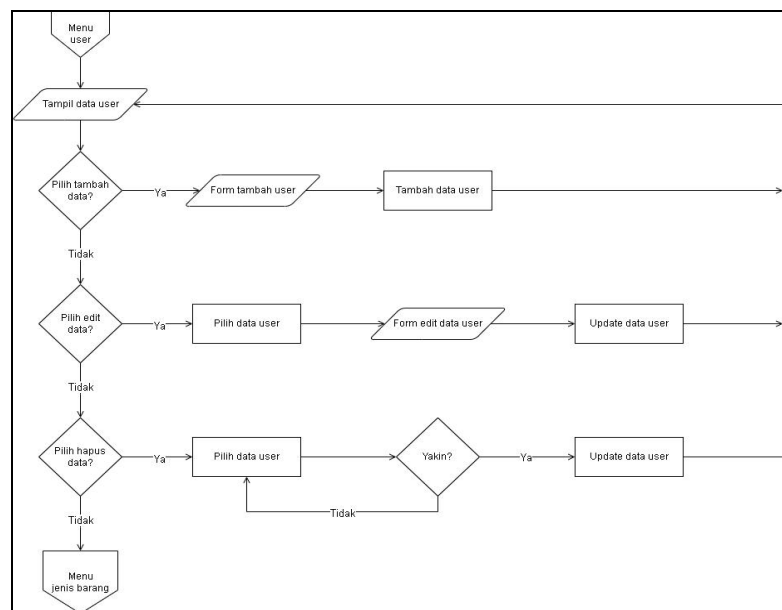
Pada *flowchart* menu utama menampilkan seluruh menu yang ada pada aplikasi. Menu yang ada yaitu menu *user*, jenis barang, barang, pelanggan, transaksi, dan laporan. Menu utama adalah halaman yang pertama kali muncul setelah *user* berhasil *login*. *Flowchart* menu utama dapat dilihat pada Gambar 3.38.



Gambar 3.38 Flowchart Menu Utama

C. Perancangan *Flowchart User*

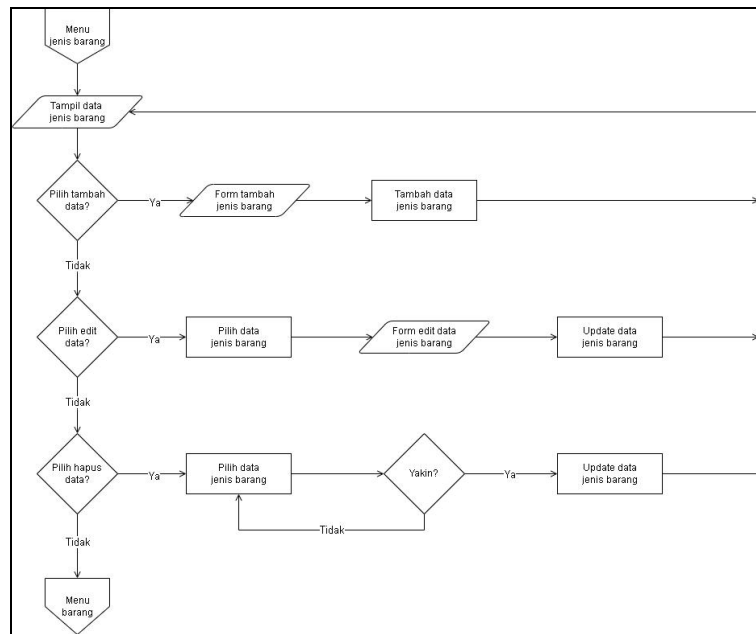
Pada *flowchart user* ini, *user* dapat melakukan input data *user*. Dalam proses ini *user* dapat menambah, mengedit, dan menghapus data sesuai dengan kebutuhan. *Flowchart user* dapat dilihat pada Gambar 3.39.



Gambar 3.39 Flowchart User

D. Perancangan *Flowchart* Jenis Barang

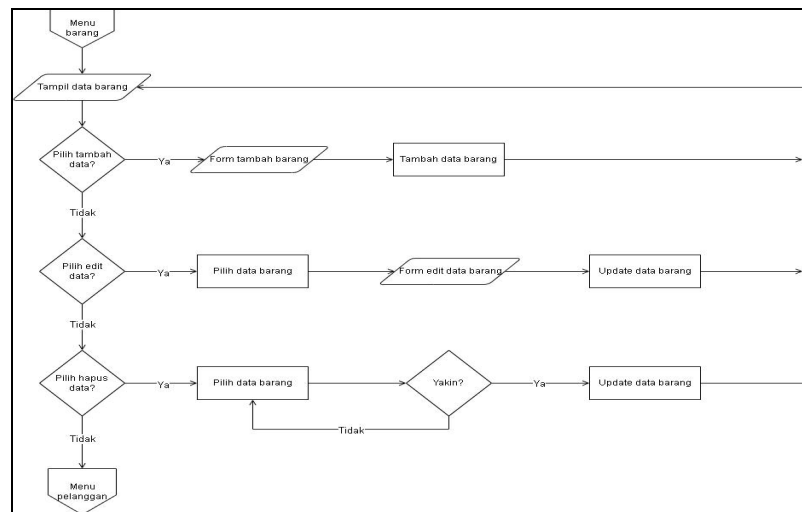
Pada *flowchart* jenis barang ini, *user* dapat melakukan input data jenis barang. Dalam proses ini *user* dapat menambah, mengedit, dan menghapus data sesuai dengan kebutuhan. *Flowchart* jenis barang dapat dilihat pada Gambar 3.40.



Gambar 3.40 *Flowchart* Jenis Barang

E. Perancangan *Flowchart* Barang

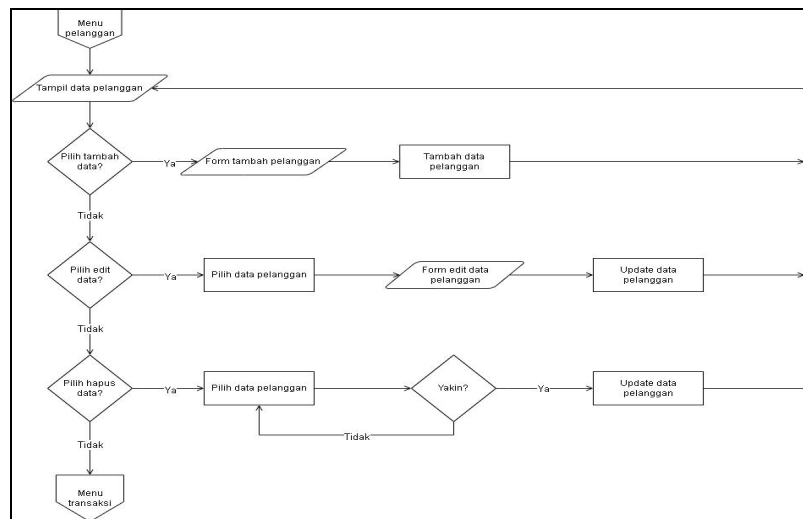
Pada *flowchart* barang ini, *user* dapat melakukan input data barang. Dalam proses ini *user* dapat menambah, mengedit, dan menghapus data sesuai dengan kebutuhan. *Flowchart* barang dapat dilihat pada Gambar 3.41.



Gambar 3.41 Flowchart Barang

F. Perancangan Flowchart Pelanggan

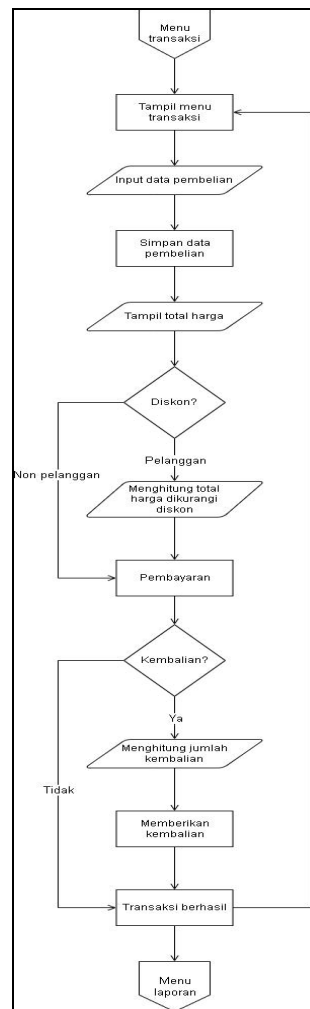
Pada *flowchart* pelanggan ini, *user* dapat melakukan input data pelanggan. Dalam proses ini *user* dapat menambah, mengedit, dan menghapus data sesuai dengan kebutuhan. *Flowchart* pelanggan dapat dilihat pada Gambar 3.42.



Gambar 3.42 Flowchart Pelanggan

G. Perancangan *Flowchart* Transaksi

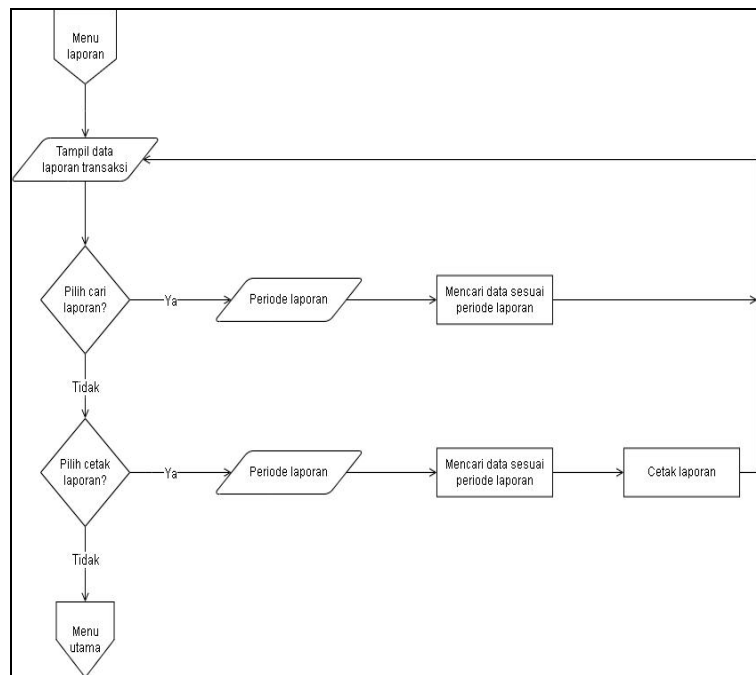
Pada *flowchart* transaksi, *user* dapat melakukan transaksi pembelian terkait dengan barang yang dibeli oleh pembeli. *Flowchart* transaksi dapat dilihat pada Gambar 3.43.



Gambar 3.43 *Flowchart* Transaksi

H. Perancangan *Flowchart* Laporan

Pada *flowchart* laporan, *user* dapat melihat data laporan dari transaksi yang berhasil dilakukan. Dalam proses ini *user* bisa mencari dan mencetak data transaksi yang dibutuhkan. *Flowchart* laporan bisa dilihat pada Gambar 3.44.



Gambar 3.44 Flowchart Laporan

c. Perancangan Antarmuka Sistem

Proses perancangan antarmuka sistem pada aplikasi penjualan CV. Athala Lautan Mulia meliputi Halaman *login*, Menu utama, menu *user*, menu jenis barang, menu barang, menu pelanggan, menu transaksi, menu laporan, dan proses *logout*.

A. Halaman *Login*

Perancangan antarmuka proses *login* digunakan oleh *user* untuk masuk ke aplikasi penjualan pada CV. Athala Lautan Mulia. *User* akan memasukkan *username* dan *password*, jika *username* dan *password* sesuai dengan data yang ada di *database* maka *user* akan berhasil login, jika *username* dan *password* salah maka akan ada peringatan bahwa *username* dan *password* salah. Perancangan antarmuka antarmuka proses *login* dapat dilihat pada Gambar 3.45.

| | |
|--|---|
| <p>Login</p> <p>Masukkan Username dan Password Anda</p> <p>Username <input type="text"/></p> <p>Password <input type="password"/></p> <p><input type="button" value="Login"/></p> | <p>CV.Athala Lautan Mulia</p> <p>Silahkan Login Terlebih Dahulu Untuk Menggunakan Aplikasi Ini</p> |
|--|---|

Gambar 3.45 Perancangan Antarmuka Proses *Login*

B. Menu Utama

Perancangan antarmuka menu utama adalah menu yang pertama kali muncul ketika user berhasil *login* ke aplikasi. Pada menu utama hanya ada beranda saja dan beberapa pilihan menu yaitu menu *user*, menu jenis barang, menu barang, menu pelanggan, menu transaksi, menu laporan, dan *logout*. Perancangan antarmuka menu utama dapat dilihat pada Gambar 3.46.

| | |
|--------------|---|
| | |
| Beranda | Beranda |
| User | |
| Jenis Barang | Aplikasi Penjualan CV.Athala Lautan Mulia |
| Barang | |
| Pelanggan | |
| Transaksi | |
| Laporan | |
| Logout | |
| | |

Gambar 3.46 Perancangan Antarmuka Menu Utama

C. Menu User

Perancangan antarmuka menu *user* digunakan oleh *user* untuk melihat data *user*, menambah data *user*, mengedit data *user*, dan menghapus data *user*. Perancangan antarmuka menu *user* dapat dilihat pada Gambar 3.47 sampai dengan Gambar 3.50.

| | | | | | | |
|--------------|---------------------------------------|----------|----------|-------|--------------------------------------|-------------------------------------|
| | | | | | | |
| Beranda | User | | | | | |
| User | | | | | | |
| Jenis Barang | <input type="button" value="Tambah"/> | | | | | |
| Barang | | | | | | |
| Pelanggan | ID Petugas | Username | Password | Level | Action | |
| Transaksi | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> | <input type="button" value="Edit"/> |
| Laporan | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> | <input type="button" value="Edit"/> |
| Logout | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> | <input type="button" value="Edit"/> |

Gambar 3.47 Perancangan Antarmuka Menu User

| | | | | | | |
|--------------|---------------------------------------|--|--|--|--|--|
| | | | | | | |
| Beranda | Input User | | | | | |
| User | | | | | | |
| Jenis Barang | ID User <input type="text"/> | | | | | |
| Barang | Username <input type="text"/> | | | | | |
| Pelanggan | Password <input type="text"/> | | | | | |
| Transaksi | Level | | | | | |
| Laporan | --Pilih Level-- <input type="text"/> | | | | | |
| Logout | <input type="button" value="Simpan"/> | | | | | |

Gambar 3.48 Perancangan Antarmuka Tambah Data User

| | | | | | | |
|--------------|---------------------------------------|--|--|--|--|--|
| | | | | | | |
| Beranda | Input User | | | | | |
| User | | | | | | |
| Jenis Barang | ID User <input type="text"/> | | | | | |
| Barang | Username <input type="text"/> | | | | | |
| Pelanggan | Password <input type="text"/> | | | | | |
| Transaksi | Level | | | | | |
| Laporan | --Pilih Level-- <input type="text"/> | | | | | |
| Logout | <input type="button" value="Simpan"/> | | | | | |

Gambar 3.49 Perancangan Antarmuka Edit User

| Beranda | User | | | | | | | | | | | | | | | | | | | | |
|--------------|--|------------|----------|------------|-------|--------|-----|-----|-----|-----|------------|-----|-----|-----|-----|------------|-----|-----|-----|-----|------------|
| User | | | | | | | | | | | | | | | | | | | | | |
| Jenis Barang | Tambah | | | | | | | | | | | | | | | | | | | | |
| Barang | | | | | | | | | | | | | | | | | | | | | |
| Pelanggan | <table border="1"> <thead> <tr> <th>ID Petugas</th> <th>Username</th> <th>Password</th> <th>Level</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>Hapus Edit</td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>Hapus Edit</td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>Hapus Edit</td> </tr> </tbody> </table> | ID Petugas | Username | Password | Level | Action | xxx | xxx | xxx | xxx | Hapus Edit | xxx | xxx | xxx | xxx | Hapus Edit | xxx | xxx | xxx | xxx | Hapus Edit |
| ID Petugas | Username | Password | Level | Action | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | Hapus Edit | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | Hapus Edit | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | Hapus Edit | | | | | | | | | | | | | | | | | |
| Transaksi | | | | | | | | | | | | | | | | | | | | | |
| Laporan | | | | | | | | | | | | | | | | | | | | | |
| Logout | | | | | | | | | | | | | | | | | | | | | |

Gambar 3.50 Perancangan Antarmuka Hapus Data *User*

D. Menu Jenis Barang

Perancangan antarmuka menu jenis barang digunakan oleh *user* untuk melihat data jenis barang, menambah data jenis barang, mengedit data jenis barang, dan menghapus data jenis barang. Perancangan antarmuka menu jenis barang dapat dilihat pada Gambar 3.51 sampai dengan Gambar 3.54.

| Beranda | Jenis Barang | | | | | | | | | | | | |
|--------------|---|------------|------------|--------|-----|-----|------------|-----|-----|------------|-----|-----|------------|
| User | | | | | | | | | | | | | |
| Jenis Barang | Tambah | | | | | | | | | | | | |
| Barang | | | | | | | | | | | | | |
| Pelanggan | <table border="1"> <thead> <tr> <th>ID Jenis</th> <th>Nama Jenis</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>xxx</td> <td>xxx</td> <td>Hapus Edit</td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>Hapus Edit</td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>Hapus Edit</td> </tr> </tbody> </table> | ID Jenis | Nama Jenis | Action | xxx | xxx | Hapus Edit | xxx | xxx | Hapus Edit | xxx | xxx | Hapus Edit |
| ID Jenis | Nama Jenis | Action | | | | | | | | | | | |
| xxx | xxx | Hapus Edit | | | | | | | | | | | |
| xxx | xxx | Hapus Edit | | | | | | | | | | | |
| xxx | xxx | Hapus Edit | | | | | | | | | | | |
| Transaksi | | | | | | | | | | | | | |
| Laporan | | | | | | | | | | | | | |
| Logout | | | | | | | | | | | | | |

Gambar 3.51 Perancangan Antarmuka Menu Jenis Barang

| | |
|--------------|---------------------------------------|
| | |
| Beranda | Input Jenis Barang |
| User | |
| Jenis Barang | ID Jenis <input type="text"/> |
| Barang | Nama Jenis <input type="text"/> |
| Pelanggan | <input type="button" value="Simpan"/> |
| Transaksi | |
| Laporan | |
| Logout | |
| | |

Gambar 3.52 Perancangan Antarmuka Tambah Data Jenis Barang

| | |
|--------------|---------------------------------------|
| | |
| Beranda | Input Jenis Barang |
| User | |
| Jenis Barang | ID Jenis <input type="text"/> |
| Barang | Nama Jenis <input type="text"/> |
| Pelanggan | <input type="button" value="Simpan"/> |
| Transaksi | |
| Laporan | |
| Logout | |
| | |

Gambar 3.52 Perancangan Antarmuka Edit Data Jenis Barang

| Beranda | Jenis Barang | | | | | | | | | | | | |
|--------------|---|--|------------|--------|-----|-----|--|-----|-----|--|-----|-----|--|
| User | | | | | | | | | | | | | |
| Jenis Barang | <input type="button" value="Tambah"/> | | | | | | | | | | | | |
| Barang | <table border="1"> <thead> <tr> <th>ID Jenis</th> <th>Nama Jenis</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> </tbody> </table> | ID Jenis | Nama Jenis | Action | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> |
| ID Jenis | Nama Jenis | Action | | | | | | | | | | | |
| xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | |
| xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | |
| xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | |
| Pelanggan | | | | | | | | | | | | | |
| Transaksi | | | | | | | | | | | | | |
| Laporan | | | | | | | | | | | | | |
| Logout | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

Gambar 3.53 Perancangan Antarmuka Hapus Data Jenis Barang

E. Menu Barang

Perancangan antarmuka menu barang digunakan oleh *user* untuk melihat data barang, menambah data barang, mengedit data barang, dan menghapus barang. Perancangan antarmuka menu barang dapat dilihat pada Gambar 3.55 sampai dengan Gambar 3.58.

| | | | | | | | |
|--------------|---------------------------------------|-------------|-------|------|--------------|--------------------------------------|-------------------------------------|
| | | | | | | | |
| Beranda | Barang | | | | | | |
| User | | | | | | | |
| Jenis Barang | <input type="button" value="Tambah"/> | | | | | | |
| Barang | ID Barang | Nama Barang | Harga | Stok | Jenis Barang | Action | |
| Pelanggan | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> | <input type="button" value="Edit"/> |
| Transaksi | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> | <input type="button" value="Edit"/> |
| Laporan | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> | <input type="button" value="Edit"/> |
| Logout | | | | | | | |

Gambar 3.55 Perancangan Antarmuka Menu Barang

| | | |
|--------------|---|--|
| | | |
| Beranda | Input Barang | |
| User | | |
| Jenis Barang | <input type="text" value="ID Barang"/> | |
| Barang | <input type="text" value="Nama Barang"/> | |
| Pelanggan | <input type="text" value="Jenis Barang"/> | |
| Transaksi | <input type="text" value="Pilih Jenis Barang"/> | |
| Laporan | <input type="text" value="Harga Barang"/> | |
| Logout | <input type="text" value="Stok Barang"/> | |
| | <input type="button" value="Simpan"/> | |

Gambar 3.56 Perancangan Antarmuka Tambah Data Barang

| | |
|--------------|---|
| | |
| Beranda | Input Barang |
| User | |
| Jenis Barang | ID Barang <input type="text"/> |
| Barang | Nama Barang <input type="text"/> |
| Pelanggan | Jenis Barang <input type="text"/> |
| Transaksi | Pilih Jenis Barang <input type="text"/> |
| Laporan | Harga Barang <input type="text"/> |
| Logout | Stok Barang <input type="text"/> |
| | <input type="button" value="Simpan"/> |

Gambar 3.57 Perancangan Antarmuka Edit Data Barang

| Beranda | Barang | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---|-----------|-------------|--------------|--|--------------|--------|-----|-----|-----|-----|-----|--|-----|--|--|--|--|--|-----|-----|-----|-----|-----|--|
| User | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jenis Barang | <input type="button" value="Tambah"/> | | | | | | | | | | | | | | | | | | | | | | | | |
| Barang | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pelanggan | <table border="1"> <thead> <tr> <th>ID Barang</th> <th>Nama Barang</th> <th>Harga</th> <th>Stok</th> <th>Jenis Barang</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> <tr> <td>xxx</td> <td colspan="4" style="text-align: center;"> Yakin Ingin Menghapus ? <input type="button" value="Oke"/> <input type="button" value="Batal"/> </td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> </tbody> </table> | ID Barang | Nama Barang | Harga | Stok | Jenis Barang | Action | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | Yakin Ingin Menghapus ? <input type="button" value="Oke"/> <input type="button" value="Batal"/> | | | | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> |
| ID Barang | Nama Barang | Harga | Stok | Jenis Barang | Action | | | | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| xxx | Yakin Ingin Menghapus ? <input type="button" value="Oke"/> <input type="button" value="Batal"/> | | | | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| Transaksi | | | | | | | | | | | | | | | | | | | | | | | | | |
| Laporan | | | | | | | | | | | | | | | | | | | | | | | | | |
| Logout | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 3.58 Perancangan Antarmuka Hapus Data Barang

F. Menu Pelanggan

Perancangan antarmuka menu pelanggan digunakan oleh *user* untuk melihat data pelanggan, menambah data pelanggan, mengedit data pelanggan, dan menghapus pelanggan. Perancangan antarmuka menu barang dapat dilihat pada Gambar 3.59 sampai dengan Gambar 3.62.

| Beranda | Pelanggan | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|--|--------------|----------------|--------|--|-------|--------|-----|-----|-----|-----|-----|--|-----|-----|-----|-----|-----|--|-----|-----|-----|-----|-----|--|
| User | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jenis Barang | <input type="button" value="Tambah"/> | | | | | | | | | | | | | | | | | | | | | | | | |
| Barang | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pelanggan | <table border="1"> <thead> <tr> <th>ID Pelanggan</th> <th>Nama Pelanggan</th> <th>Alamat</th> <th>No.Telp</th> <th>Email</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td><input type="button" value="Hapus"/> <input type="button" value="Edit"/></td> </tr> </tbody> </table> | ID Pelanggan | Nama Pelanggan | Alamat | No.Telp | Email | Action | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> |
| ID Pelanggan | Nama Pelanggan | Alamat | No.Telp | Email | Action | | | | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| Transaksi | | | | | | | | | | | | | | | | | | | | | | | | | |
| Laporan | | | | | | | | | | | | | | | | | | | | | | | | | |
| Logout | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 3.59 Perancangan Antarmuka Menu Pelanggan

| | |
|--------------|---------------------------------------|
| | |
| Beranda | Input Pelanggan |
| User | |
| Jenis Barang | ID Pelanggan |
| Barang | Nama Pelanggan |
| Pelanggan | Alamat |
| Transaksi | No.Telp |
| Laporan | Email |
| Logout | |
| | <input type="button" value="Simpan"/> |

Gambar 3.60 Perancangan Antarmuka Tambah Data Pelanggan

| | |
|--------------|---------------------------------------|
| | |
| Beranda | Input Pelanggan |
| User | |
| Jenis Barang | ID Pelanggan |
| Barang | Nama Pelanggan |
| Pelanggan | Alamat |
| Transaksi | No.Telp |
| Laporan | Email |
| Logout | |
| | <input type="button" value="Simpan"/> |

Gambar 3.61 Perancangan Antarmuka Edit Data Pelanggan

| Beranda | Pelanggan | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---|--------------|----------------|--------|--|-------|--------|-----|-----|-----|-----|-----|--|-----|--|--|--|-----|--|-----|-----|-----|-----|-----|--|
| User | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jenis Barang | <input type="button" value="Tambah"/> | | | | | | | | | | | | | | | | | | | | | | | | |
| Barang | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pelanggan | <table border="1"> <thead> <tr> <th>ID Pelanggan</th> <th>Nama Pelanggan</th> <th>Alamat</th> <th>No.Telp</th> <th>Email</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td> <input type="button" value="Hapus"/> <input type="button" value="Edit"/> </td> </tr> <tr> <td>xxx</td> <td colspan="3"> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> Yakin Ingin Menghapus ? <input type="button" value="Oke"/> <input type="button" value="Batal"/> </div> </td> <td>xxx</td> <td> <input type="button" value="Hapus"/> <input type="button" value="Edit"/> </td> </tr> <tr> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td>xxx</td> <td> <input type="button" value="Hapus"/> <input type="button" value="Edit"/> </td> </tr> </tbody> </table> | ID Pelanggan | Nama Pelanggan | Alamat | No.Telp | Email | Action | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | <div style="border: 1px solid black; padding: 5px; display: inline-block;"> Yakin Ingin Menghapus ? <input type="button" value="Oke"/> <input type="button" value="Batal"/> </div> | | | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> |
| ID Pelanggan | Nama Pelanggan | Alamat | No.Telp | Email | Action | | | | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| xxx | <div style="border: 1px solid black; padding: 5px; display: inline-block;"> Yakin Ingin Menghapus ? <input type="button" value="Oke"/> <input type="button" value="Batal"/> </div> | | | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| xxx | xxx | xxx | xxx | xxx | <input type="button" value="Hapus"/> <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | |
| Transaksi | | | | | | | | | | | | | | | | | | | | | | | | | |
| Laporan | | | | | | | | | | | | | | | | | | | | | | | | | |
| Logout | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 3.62 Perancangan Antarmuka Hapus Data Pelanggan

G. Menu Transaksi

Perancangan antarmuka menu transaksi digunakan oleh *user* ketika akan melakukan transaksi dengan pembeli terkait barang yang dibeli. Perancangan antarmuka menu transaksi dapat dilihat pada Gambar 3.63.

| | | | | | | |
|--------------|----------------------|----------------------|----------------------|--|-------------|--------------------------------------|
| Beranda | Transaksi | | | | | |
| User | | | | | | |
| Jenis Barang | ID Transaksi | <input type="text"/> | | | | |
| Barang | Tanggal Transaksi | <input type="text"/> | | | | |
| Pelanggan | Kategori | ID Pelanggan | Nama Pelanggan | | | |
| Transaksi | Pilih kategori | <input type="text"/> | <input type="text"/> | | | |
| Laporan | ID Barang | Nama Barang | Harga | | | |
| Logout | Jumlah | Total | <input type="text"/> | | | |
| | | | | <input type="button" value="Simpan"/> | | |
| | No | ID Transaksi | Nama Barang | Harga | Jumlah Beli | Total |
| | | | | | | Action |
| | | | | | | <input type="button" value="Hapus"/> |
| | Sub Total | Diskon | Total Bayar | | | |
| | <input type="text"/> | <input type="text"/> | <input type="text"/> | | | |
| | Bayar | Kembali | <input type="text"/> | | | |
| | <input type="text"/> | <input type="text"/> | <input type="text"/> | | | |
| | | | | <input type="button" value="Transaksi"/> | | |

Gambar 3.63 Perancangan Antarmuka Menu Transaksi

H. Menu Laporan

Perancangan antarmuka menu laporan ini terdiri dari beberapa sub-menu yaitu laporan per tanggal, laporan per bulan dan laporan per tahun. Pada menu laporan *user* bisa mencari laporan sesuai dengan tanggal, bulan atau tahun yang dibutuhkan, *user* juga bisa mencetak laporan sesuai dengan tanggal, bulan, atau tahun yang dibutuhkan. Perancangan antarmuka menu laporan dapat dilihat pada Gambar 3.64 sampai dengan Gambar 3.69.

| | | | | | | |
|--------------|---|---------|-----------|-------|--------|-------------|
| | | | | | | |
| Beranda | Laporan Per Tanggal | | | | | |
| User | | | | | | |
| Jenis Barang | Dari <input type="text"/> s/d <input type="text"/> <input type="button" value="Cari"/> <input type="button" value="Cetak"/> | | | | | |
| Barang | | | | | | |
| Pelanggan | ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
| Transaksi | xxx | xxx | xxx | xxx | xxx | xxx |
| Laporan | xxx | xxx | xxx | xxx | xxx | xxx |
| Logout | xxx | xxx | xxx | xxx | xxx | xxx |
| | | | | | | |

Gambar 3.64 Perancangan Menu Laporan Per Tanggal

| | | | | | | |
|--------------|--|---------|-----------|-------|--------|-------------|
| | | | | | | |
| Beranda | Laporan Per Bulan | | | | | |
| User | | | | | | |
| Jenis Barang | Bulan <input type="text"/> Tahun <input type="text"/> <input type="button" value="Cari"/> <input type="button" value="Cetak"/> | | | | | |
| Barang | | | | | | |
| Pelanggan | ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
| Transaksi | xxx | xxx | xxx | xxx | xxx | xxx |
| Laporan | xxx | xxx | xxx | xxx | xxx | xxx |
| Logout | xxx | xxx | xxx | xxx | xxx | xxx |
| | | | | | | |

Gambar 3.65 Perancangan Menu Laporan Per Bulan

| | | | | | | |
|--------------|---|---------|-----------|-------|--------|-------------|
| | | | | | | |
| Beranda | Laporan Per Tahun | | | | | |
| User | | | | | | |
| Jenis Barang | Tahun <input type="text"/> <input type="button" value="Cari"/> <input type="button" value="Cetak"/> | | | | | |
| Barang | | | | | | |
| Pelanggan | ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
| Transaksi | xxx | xxx | xxx | xxx | xxx | xxx |
| Laporan | xxx | xxx | xxx | xxx | xxx | xxx |
| Logout | xxx | xxx | xxx | xxx | xxx | xxx |
| | | | | | | |

Gambar 3.66 Perancangan Menu Laporan Per Tahun

| Laporan Per Tanggal | | | | | |
|---------------------|---------|-----------|-------|--------|-------------|
| ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
| xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx |

Gambar 3.67 Perancangan Cetak Data Laporan Per Tanggal

| Laporan Per Bulan | | | | | |
|-------------------|---------|-----------|-------|--------|-------------|
| ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
| xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx |

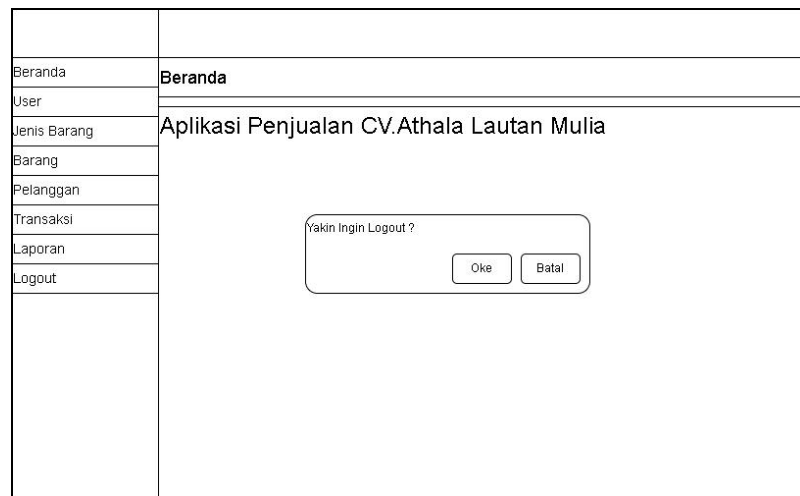
Gambar 3.68 Perancangan Cetak Data Laporan Per Bulan

| Laporan Per Tahun | | | | | |
|-------------------|---------|-----------|-------|--------|-------------|
| ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
| xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx |

Gambar 3.69 Perancangan Cetak Data Laporan Per Tahun

I. Logout

Perancangan antarmuka *logout* ketika *user* ingin keluar dari aplikasi. Perancangan *logout* dapat dilihat pada Gambar 3.70.



Gambar 3.70 Perancangan Logout

3.2.3. *Implementation (Implementasi)*

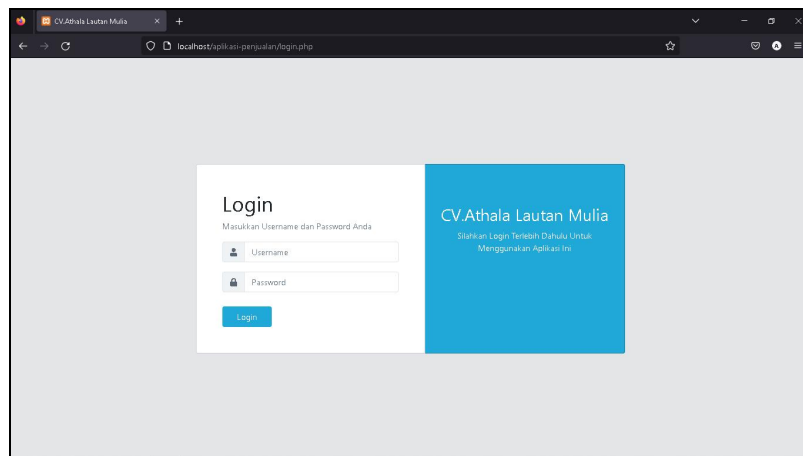
Pada fase ini, mengimplementasikan aplikasi yang telah dibuat dengan melakukan demo aplikasi kepada *user* yang akan menggunakan aplikasi.

a. Halaman Masukan

Halaman masukan merupakan bentuk nyata dari semua perancangan aplikasi yang diusulkan. Halaman masukan terdiri dari halaman login, menu utama, menu user, menu jenis barang, menu barang, menu pelanggan, dan menu transaksi.

A. Halaman Login

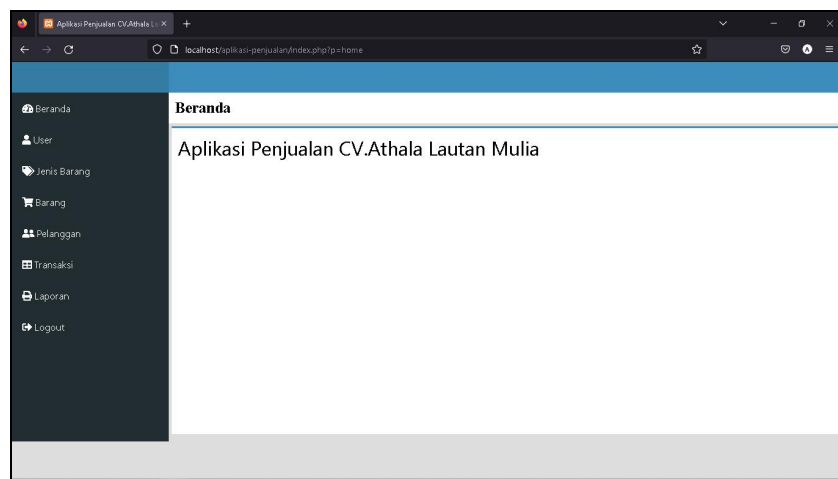
Halaman yang pertama kali digunakan oleh *user* untuk masuk ke aplikasi penjualan pada CV. Athala Lautan Mulia. *User* akan memasukkan *username* dan *password*, jika *username* dan *password* sesuai dengan data yang ada di *database* maka *user* akan berhasil login, jika *username* dan *password* salah maka akan ada peringatan bahwa *username* dan *password* salah. Implementasi halaman login dapat dilihat pada Gambar 3.71.



Gambar 3.71 Implementasi Halaman *Login*

B. Menu Utama

Menu utama adalah menu yang pertama kali muncul ketika user berhasil *login* ke aplikasi. Pada menu utama hanya ada beranda saja dan beberapa pilihan menu yaitu menu *user*, menu jenis barang, menu barang, menu pelanggan, menu transaksi, menu laporan, dan *logout*. Implementasi menu utama dapat dilihat pada Gambar 3.72.

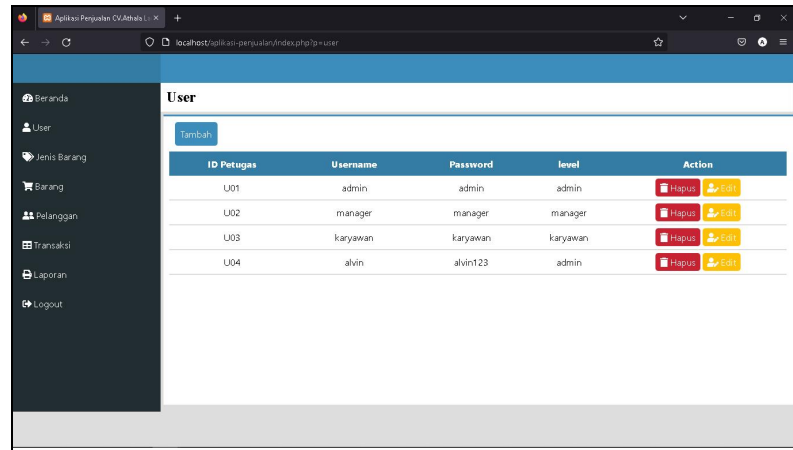


Gambar 3.72 Implementasi Menu Utama

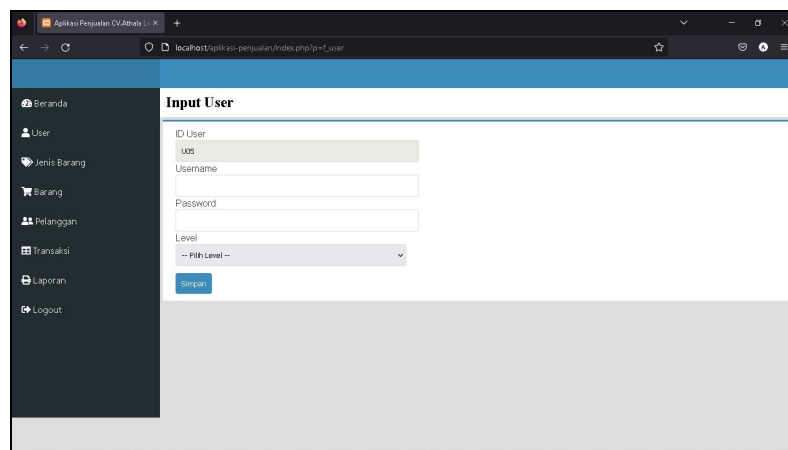
C. Menu User

Menu *user* digunakan oleh *user* untuk melihat data *user*, menambah data *user*, mengedit data *user*, dan menghapus data *user*.

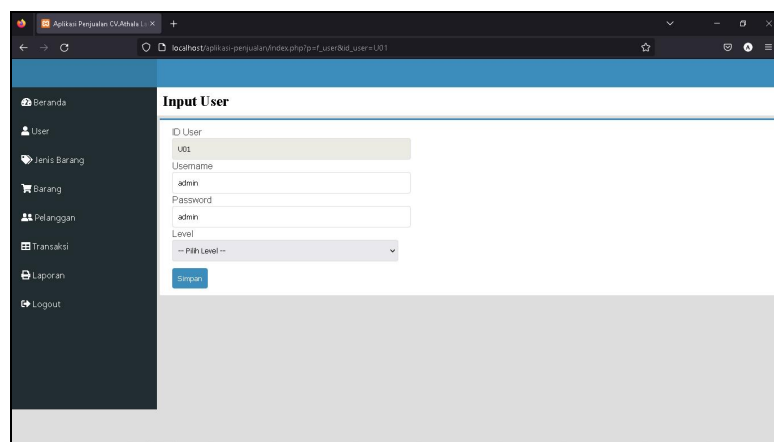
Implementasi menu *user* dapat dilihat pada Gambar 3.73 sampai dengan Gambar 3.76.



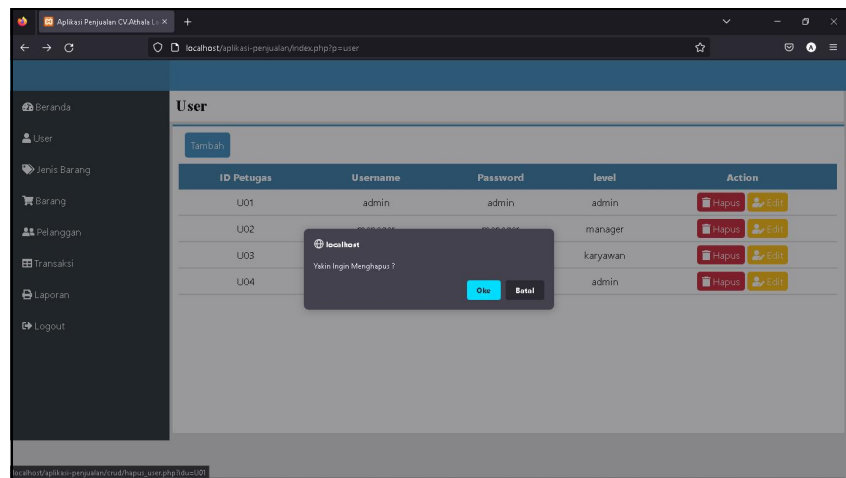
Gambar 3.73 Implementasi Menu *User*



Gambar 3.74 Implementasi Tambah Data *User*



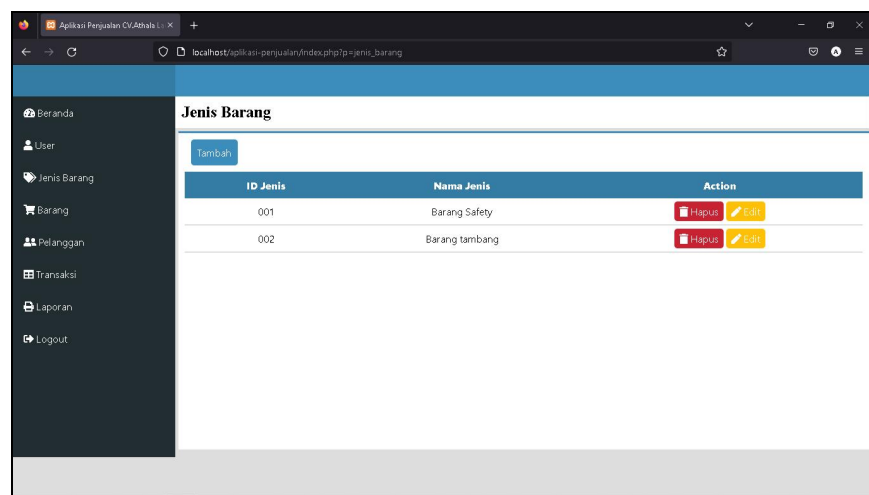
Gambar 3.75 Implementasi Edit Data *User*



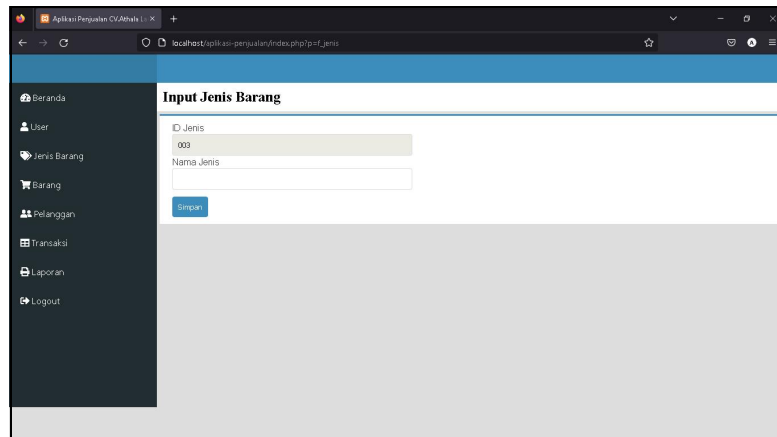
Gambar 3.76 Implementasi Hapus Data User

D. Menu Jenis Barang

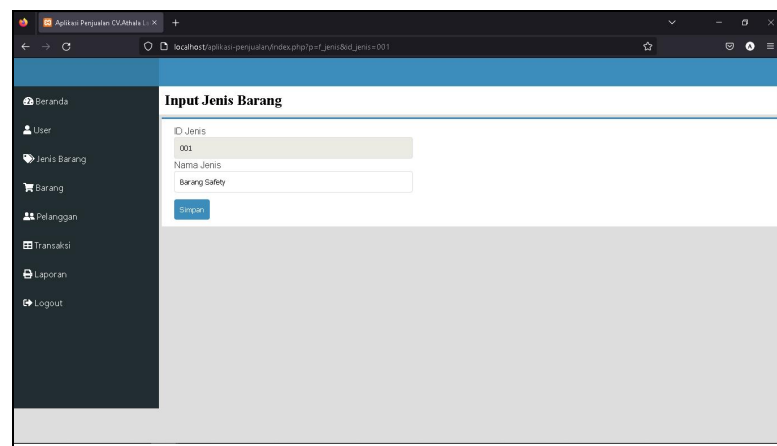
Menu jenis barang digunakan oleh *user* untuk melihat data jenis barang, menambah data jenis barang, mengedit data jenis barang, dan menghapus data jenis barang. Implementasi menu jenis barang dapat dilihat pada Gambar 3.77 sampai dengan Gambar 3.80.



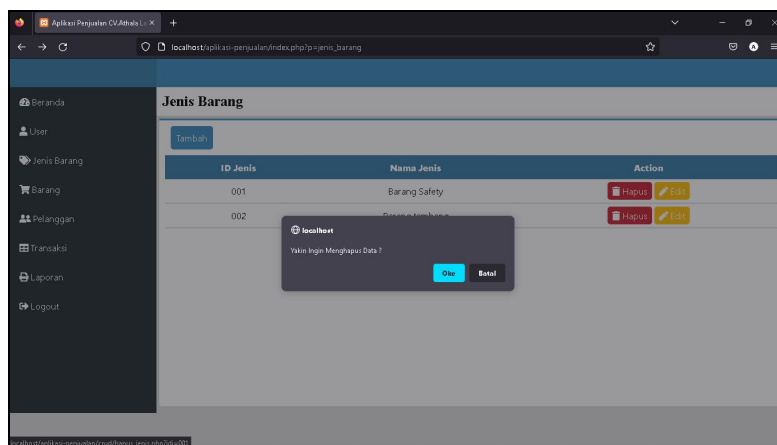
Gambar 3.77 Implementasi Menu Jenis Barang



Gambar 3.78 Implementasi Tambah Data Jenis Barang



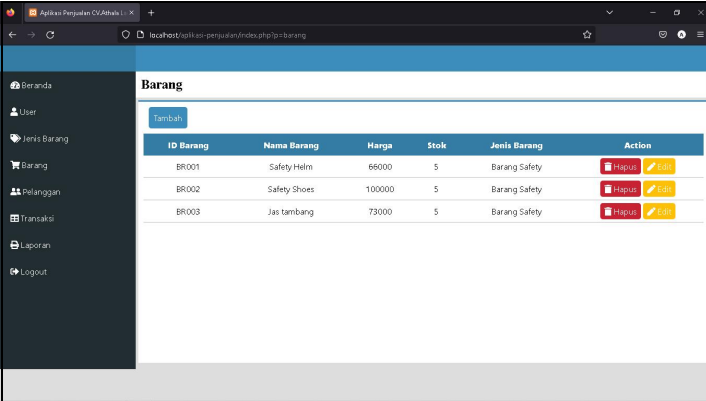
Gambar 3.79 Implementasi Edit Data Jenis Barang



Gambar 3.80 Implementasi Hapus Data Jenis Barang

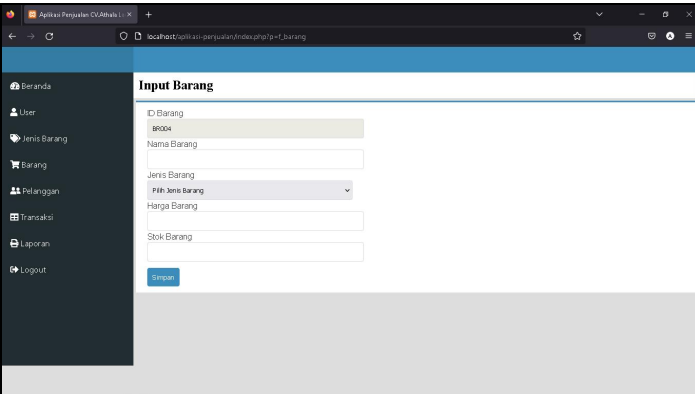
E. Menu Barang

Menu barang digunakan oleh *user* untuk melihat data barang, menambah data barang, mengedit data barang, dan menghapus barang. Implementasi menu barang dapat dilihat pada Gambar 3.81 sampai dengan Gambar 3.84.

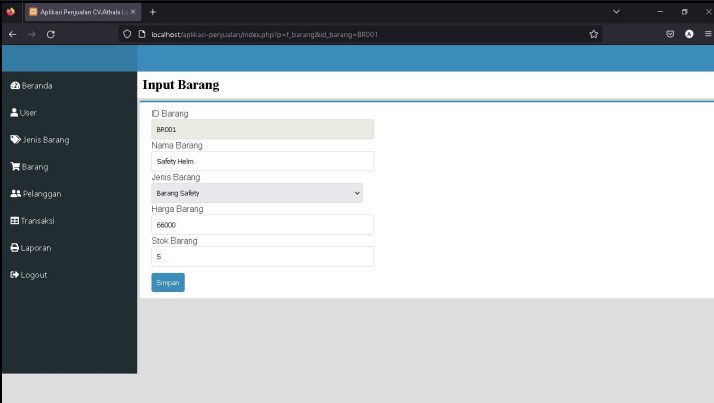


| ID Barang | Nama Barang | Harga | Stok | Jenis Barang | Action |
|-----------|--------------|--------|------|---------------|------------|
| BR001 | Safety Helm | 66000 | 5 | Barang Safety | Hapus Edit |
| BR002 | Safety Shoes | 100000 | 5 | Barang Safety | Hapus Edit |
| BR003 | Jas tambang | 73000 | 5 | Barang Safety | Hapus Edit |

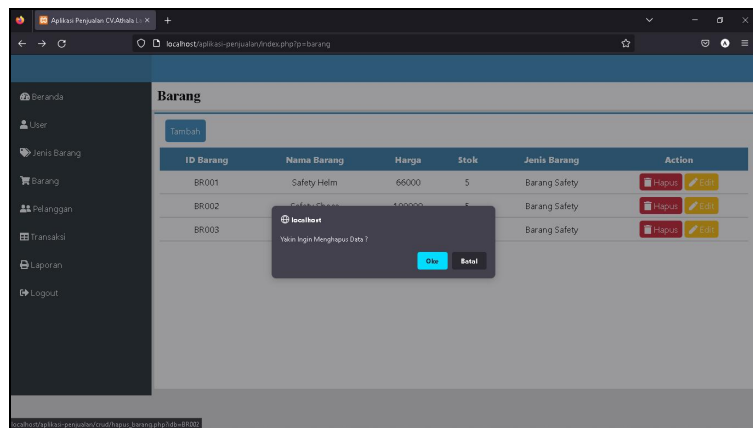
Gambar 3.81 Implementasi Menu Barang



Gambar 3.82 Implementasi Tambah Data Barang



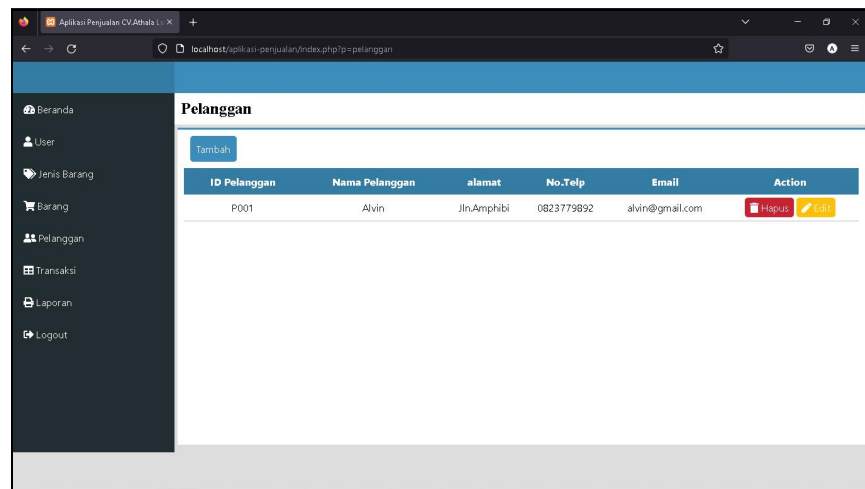
Gambar 3.83 Implementasi Edit Data Barang



Gambar 3.84 Implementasi Hapus Data Barang

F. Menu Pelanggan

Menu pelanggan digunakan oleh *user* untuk melihat data pelanggan, menambah data pelanggan, mengedit data pelanggan, dan menghapus pelanggan. Implementasi menu barang dapat dilihat pada Gambar 3.85 sampai dengan Gambar 3.88.



Gambar 3.85 Implementasi Menu Pelanggan

The screenshot shows a web browser window with the URL `localhost/aplikasi-perjualan/index.php?p=f_pelanggan`. The page title is "Input Pelanggan". On the left is a dark sidebar menu with options: Beranda, User, Jenis Barang, Barang, Pelanggan, Transaksi, Laporan, and Logout. The main content area contains a form with the following fields:

- ID Pelanggan: P002
- Nama Pelanggan: (empty)
- Alamat: (empty)
- No.Telp: (empty)
- E-Mail: (empty)

 A blue "Simpan" button is located at the bottom of the form.

Gambar 3.86 Implementasi Tambah Data Pelanggan

The screenshot shows the same web browser window with the URL `localhost/aplikasi-perjualan/index.php?p=f_pelanggan&id_pelanggan=P001`. The form is now filled with the following data:

- ID Pelanggan: P001
- Nama Pelanggan: Alvin
- Alamat: Jln.Amphibi
- No.Telp: 0823779892
- E-Mail: alvin@gmail.com

 The "Simpan" button is still present at the bottom.

Gambar 3.87 Implementasi Edit Data Pelanggan

The screenshot shows the web browser window with the URL `localhost/aplikasi-perjualan/index.php?p=p_pelanggan`. The page title is "Pelanggan". A "Tambah" button is at the top left. Below it is a table with the following data:

| ID Pelanggan | Nama Pelanggan | alamat | No.Telp | Email | Action |
|--------------|----------------|-------------|------------|-----------------|------------|
| P001 | Alvin | Jln.Amphibi | 0823779892 | alvin@gmail.com | Hapus Edit |

 A modal dialog is open in the center of the screen with the text "Yakin Ingin Menghapus Data?" and two buttons: "Oke" (blue) and "Batal" (grey).

Gambar 3.88 Implementasi Hapus Data Pelanggan

G. Menu Transaksi

Menu transaksi digunakan oleh *user* ketika akan melakukan transaksi dengan pembeli terkait barang yang dibeli. Implementasi menu transaksi dapat dilihat pada Gambar 3.89.

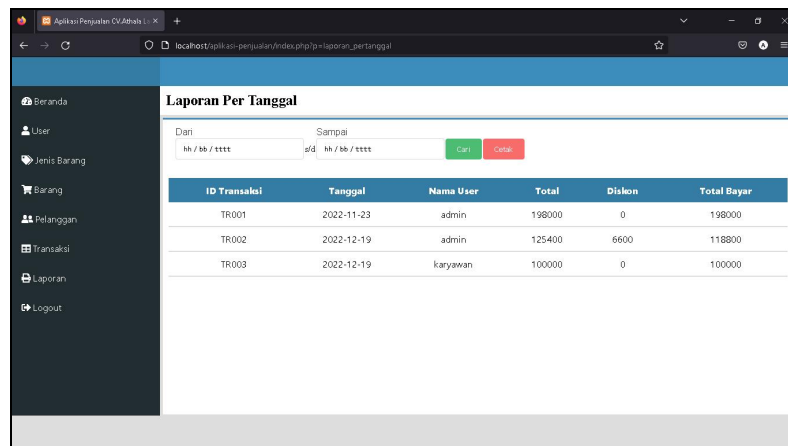
Gambar 3.89 Implementasi Menu Transaksi

b. Halaman Keluaran

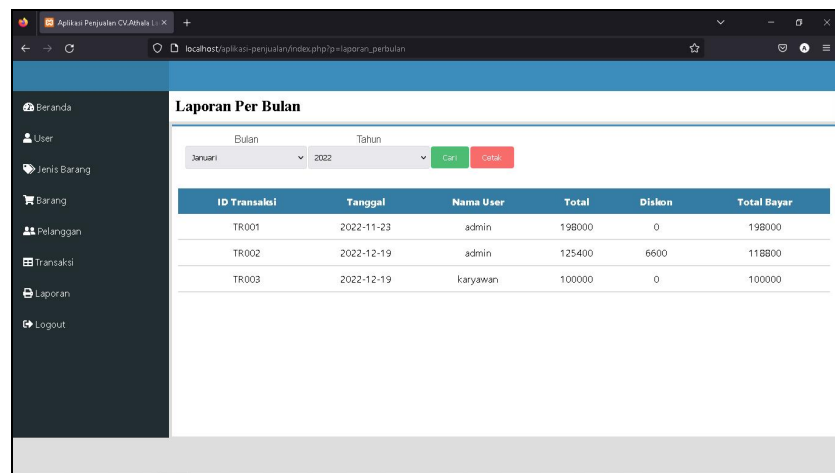
Halaman keluaran adalah bentuk nyata yang dikeluarkan dari aplikasi seluruh perancangan sistem yang telah diusulkan. Halaman keluaran terdiri dari menu laporan.

A. Menu Laporan

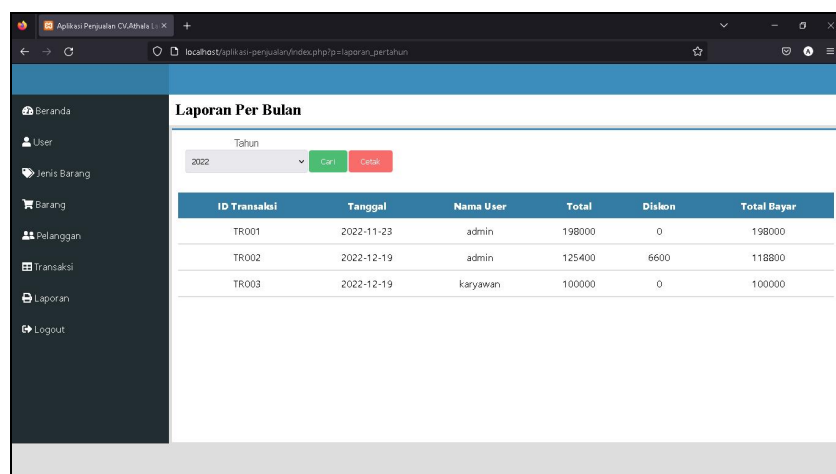
Menu laporan ini terdiri dari beberapa sub-menu yaitu laporan per tanggal, laporan per bulan dan laporan per tahun. Pada menu laporan *user* bisa mencari laporan sesuai dengan tanggal, bulan atau tahun yang dibutuhkan, *user* juga bisa mencetak laporan sesuai dengan tanggal, bulan, atau tahun yang dibutuhkan. Implementasi menu laporan dapat dilihat pada Gambar 3.90 sampai dengan Gambar 3.95.



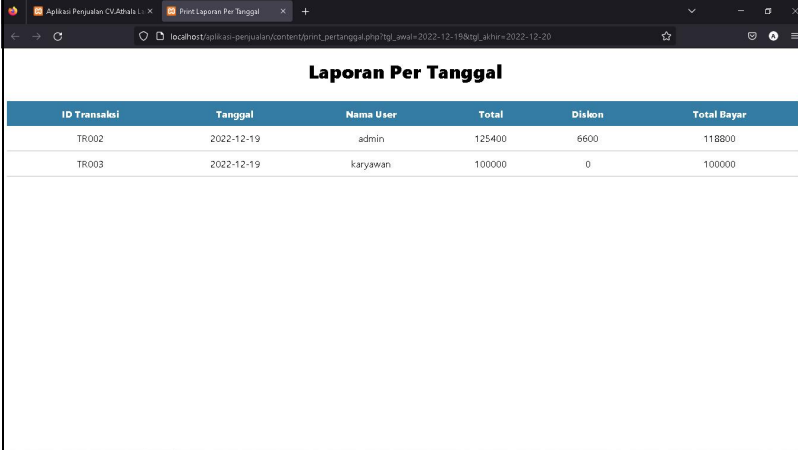
Gambar 3.90 Implementasi Menu Laporan Per Tanggal



Gambar 3.91 Implementasi Menu Laporan Per Bulan

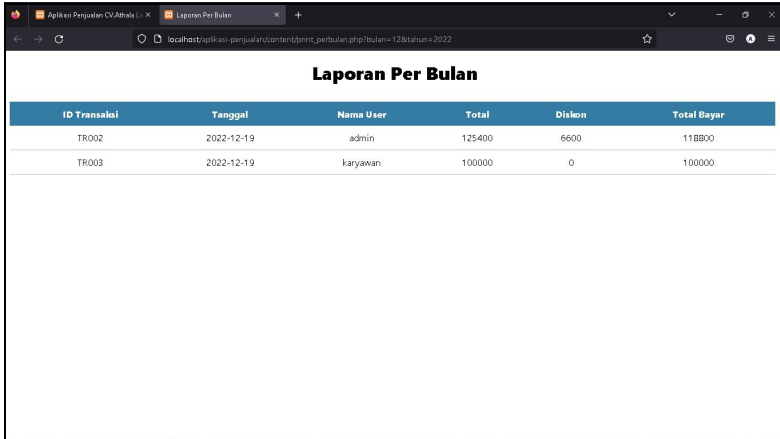


Gambar 3.92 Implementasi Menu Laporan Per Tahun



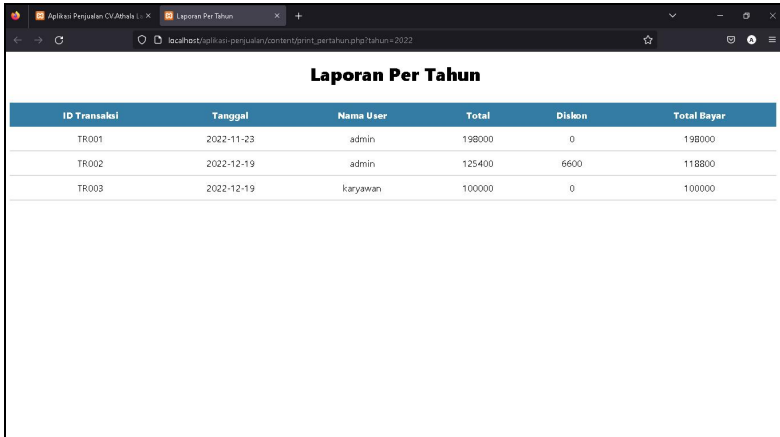
| ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
|--------------|------------|-----------|--------|--------|-------------|
| TR002 | 2022-12-19 | admin | 125400 | 6600 | 118800 |
| TR003 | 2022-12-19 | karyawan | 100000 | 0 | 100000 |

Gambar 3.93 Implementasi Cetak Laporan Per Tanggal



| ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
|--------------|------------|-----------|--------|--------|-------------|
| TR002 | 2022-12-19 | admin | 125400 | 6600 | 118800 |
| TR003 | 2022-12-19 | karyawan | 100000 | 0 | 100000 |

Gambar 3.94 Implementasi Cetak Laporan Per Bulan



| ID Transaksi | Tanggal | Nama User | Total | Diskon | Total Bayar |
|--------------|------------|-----------|--------|--------|-------------|
| TR001 | 2022-11-23 | admin | 198000 | 0 | 198000 |
| TR002 | 2022-12-19 | admin | 125400 | 6600 | 118800 |
| TR003 | 2022-12-19 | karyawan | 100000 | 0 | 100000 |

Gambar 3.95 Implementasi Cetak Laporan Per Tahun

A. Pengujian Sistem

a. Pengujian *Blackbox*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak, dapat diartikan bahwa teknik pengujian *black-box* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program (Pressman, 2012). Berikut tabel-tabel hasil pengujian *black-box* dari aplikasi yang telah dibuat :

Tabel 3.15 Hasil Uji Halaman *Login*

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|---|-----------------------|--|-----------------------------|
| 1. | <i>Username</i> dan <i>password</i> salah | Sistem menolak | Menampilkan peringatan <i>username</i> dan <i>password</i> salah | (✓) Berhasil () Gagal |
| 2. | <i>Username</i> dan <i>password</i> benar | Sistem menerima | Berhasil <i>login</i> dan masuk ke menu utama | (✓) Berhasil () Gagal |

Tabel 3.16 Hasil Uji Menu Utama

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|--------------------------|---------------------------|-----------------------|---|-----------------------------|
| Menu <i>User</i> | | | | |
| 1. | Memilih Menu <i>User</i> | Sistem menerima | Menampilkan menu data <i>user</i> , tambah data <i>user</i> , edit data <i>user</i> , dan hapus data <i>user</i> | (✓) Berhasil () Gagal |
| Menu Jenis Barang | | | | |
| 2. | Memilih Menu Jenis Barang | Sistem menerima | Menampilkan menu data jenis barang, tambah data jenis barang, edit data jenis barang, dan hapus data jenis barang | (✓) Berhasil () Gagal |

Tabel 3.16 Hasil Uji Menu Utama (*Lanjutan*)

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----------------------|--------------------------------------|-----------------------|---|--|
| Menu Barang | | | | |
| 3. | Memilih Menu Barang | Sistem menerima | Menampilkan menu data barang, tambah data barang, edit data barang, dan hapus data barang | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| Menu Pelanggan | | | | |
| 4. | Memilih Menu Pelanggan | Sistem Menerima | Menampilkan menu data pelanggan, tambah data pelanggan, edit data pelanggan, dan hapus data pelanggan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| Menu Transaksi | | | | |
| 5. | Memilih Menu Transaksi | Sistem Menerima | Menampilkan menu transaksi | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| Menu Laporan | | | | |
| 6. | Memilih Menu Laporan | Sistem Menerima | Menampilkan sub-menu laporan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 7. | Memilih sub-menu laporan per tanggal | Sistem Menerima | Menampilkan menu data laporan per tanggal, cari laporan per tanggal, dan cetak laporan per tanggal | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 8. | Memilih sub-menu laporan per bulan | Sistem Menerima | Menampilkan menu data laporan per bulan, cari laporan per bulan, dan cetak laporan per bulan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 9. | Memilih sub-menu laporan per tahun | Sistem Menerima | Menampilkan menu data laporan per tahun, cari laporan per | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

Tabel 3.16 Hasil Uji Menu Utama (*Lanjutan*)

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|----------------|-----------------------|------------------------------------|------------|
| | | | tahun, dan cetak laporan per tahun | |

Tabel 3.17 Hasil Uji Menu *User*

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|---|-----------------------|--|-----------------------------|
| 1. | Tombol tambah data <i>user</i> dan mengisi data pada form tambah data <i>user</i> | Sistem Menerima | Tambah data <i>user</i> berhasil dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu <i>user</i> | (✓) Berhasil () Gagal |
| 2. | Tombol edit data <i>user</i> dan mengisi data pada form edit data <i>user</i> | Sistem Menerima | Edit data <i>user</i> berhasil dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu <i>user</i> | (✓) Berhasil () Gagal |
| 3. | Tombol hapus data <i>user</i> | Sistem menerima | Hapus data <i>user</i> berhasil dan data akan terhapus dari <i>database</i> serta di <i>update</i> pada menu <i>user</i> | (✓) Berhasil () Gagal |

Tabel 3.18 Hasil Uji Menu Jenis Barang

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|---|-----------------------|--|-----------------------------|
| 1. | Tombol tambah data jenis barang dan mengisi data pada form tambah data jenis barang | Sistem Menerima | Tambah data jenis barang berhasil dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu jenis barang | (✓) Berhasil () Gagal |
| 2. | Tombol edit data jenis | Sistem Menerima | Edit data jenis barang berhasil | (✓) Berhasil () Gagal |

Tabel 3.18 Hasil Uji Menu Jenis Barang (*Lanjutan*)

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|--|-----------------------|--|--|
| | barang dan mengisi data pada form edit data jenis barang | | dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu jenis barang | |
| 3. | Tombol hapus data jenis barang | Sistem Menerima | Hapus data jenis barang berhasil dan data akan terhapus dari <i>database</i> serta di <i>update</i> pada menu jenis barang | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

Tabel 3.19 Hasil Uji Menu Barang

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|---|-----------------------|--|--|
| 1. | Tombol tambah data barang dan mengisi data pada form tambah data barang | Sistem Menerima | Tambah data barang berhasil dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu barang | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 2. | Tombol edit data barang dan mengisi data pada form edit data barang | Sistem Menerima | Edit data barang berhasil dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu barang | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 3. | Tombol hapus data barang | Sistem Menerima | Hapus data barang berhasil dan data akan terhapus dari <i>database</i> serta di <i>update</i> pada menu | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

Tabel 3.20 Hasil Uji Menu Pelanggan

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|---|-----------------------|--|--|
| 1. | Tombol tambah data pelanggan dan mengisi data pada form tambah data pelanggan | Sistem Menerima | Tambah data pelanggan berhasil dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu pelanggan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 2. | Tombol edit data pelanggan dan mengisi data pada form edit data pelanggan | Sistem Menerima | Edit data pelanggan berhasil dan data akan tersimpan pada <i>database</i> serta di <i>update</i> pada menu pelanggan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 3. | Tombol hapus data pelanggan | Sistem Menerima | Hapus data pelanggan berhasil dan data akan terhapus dari <i>database</i> serta di <i>update</i> pada menu pelanggan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

Tabel 3.21 Hasil Uji Menu Transaksi

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|--|-----------------------|---|--|
| 1. | Menu transaksi mengambil data dari data pelanggan dan data barang secara otomatis dengan | Sistem Menerima | Ketika diinputkan ID Pelanggan dan Barang data barang dan pelanggan akan secara otomatis muncul | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

Tabel 3.21 Hasil Uji Menu Transaksi (*Lanjutan*)

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----|--|-----------------------|---|--|
| | hanya menginput ID data tersebut | | | |
| 2. | Tombol simpan transaksi untuk menyimpan barang yang telah diinputkan kedalam tabel transaksi | Sistem Menerima | Data berhasil disimpan dalam tabel transaksi | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 3. | Tombol hapus pada tabel transaksi | Sistem Menerima | Data yang ada pada tabel transaksi akan terhapus | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 4. | Tombol transaksi untuk melakukan transaksi | Sistem Menerima | Transaksi telah berhasil dan data transaksi yang terjadi akan langsung masuk ke dalam laporan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

Tabel 3.22 Hasil Uji Menu Laporan

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-------------------------------------|---|-----------------------|--|--|
| Sub-Menu Laporan Per Tanggal | | | | |
| 1. | Tombol cari periode laporan per tanggal | Sistem Menerima | Menampilkan laporan sesuai periode tanggal yang dicari | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 2. | Tombol cetak laporan per tanggal | Sistem Menerima | Mencetak laporan sesuai dengan tanggal laporan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| Sub-Menu Laporan Per Bulan | | | | |
| 1. | Tombol cari periode | Sistem Menerima | Menampilkan laporan sesuai | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

Tabel 3.22 Hasil Uji Menu Laporan (*Lanjutan*)

| No. | Item Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Keterangan |
|-----------------------------------|---------------------------------------|-----------------------|--|--|
| | laporan per bulan | | periode bulan yang dicari | |
| 2. | Tombol cetak laporan per bulan | Sistem Menerima | Mencetak laporan sesuai dengan bulan laporan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| Sub-Menu Laporan Per Tahun | | | | |
| 1. | Tombol cari periode laporan per tahun | Sistem Menerima | Menampilkan laporan sesuai periode tahun yang dicari | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |
| 2. | Tombol cetak laporan per tahun | Sistem Menerima | Mencetak laporan sesuai dengan tahun laporan | (<input checked="" type="checkbox"/>) Berhasil (<input type="checkbox"/>) Gagal |

b. Pengujian *Whitebox*

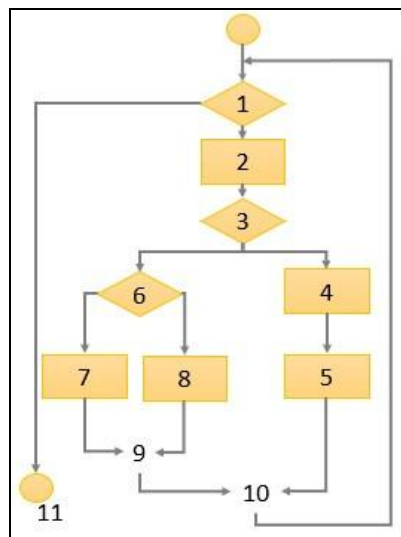
Pengujian *whitebox* disebut juga pengujian kotak kaca (*glass box testing*), merupakan sebuah filosofi perancangan *test case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan peringkat komponen untuk menghasilkan *test case* (Pressman, 2012).

Pengujian *whitebox* yang dilakukan adalah pengujian *basis path*. Metode *basis path* digunakan untuk menentukan ukuran kompleksitas logika dari suatu logika. Metode pengujian *basis path* berguna untuk :

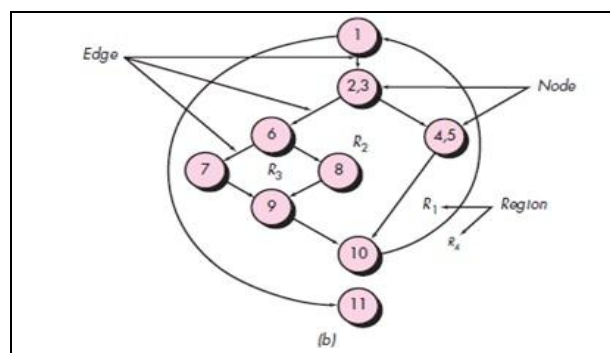
1. Mengukur kompleksitas *logic* dari desain prosedur dan sekaligus sebagai pedoman untuk mendapatkan konsistensi jalur aplikasi.
2. Pengujian yang dilakukan dijamin menggunakan statement dalam program minimal satu kali selama pengujian.

3. Menghitung kompleksitas siklomatik sebagai ukuran kontitif untuk menentukan jumlah *independent path* sebagai jalur yang perlu diuji.

Metode *basis path* ini merupakan salah satu teknik pengujian struktur kontrol untuk mengetahui berapa banyak jalur yang dapat dicari dalam setiap jalur independen yang dapat dilakukan melalui perhitungan kompleksitas siklomatik. Kompleksitas siklomatik (*Cyclomatic Complexity*) adalah metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Sebelum menghitung nilai *cyclomatic complexity* harus diterjemahkan ke desain prosedural ke diagram alir, kemudian dibuat grafik alir (flowgraph), seperti **Gambar 3.92**.



Gambar 3.96 Diagram Alir *Cyclomatic Complexity*



Gambar 3.97 Grafik Alir (*Flow Graph*)

Dari gambar grafik alir (*flowgraph*) didapat :

Path 1 : 1-11

Path 2 : 1-2-3-4-10-1-11

Path 3 : 1-2-3-6-8-9-10-1-11

Path 4 : 1-2-3-6-7-9-10-1-11

Jalur 1 sampai 4 merupakan basis set untuk grafik alir. Selanjutnya mencari jumlah path dalam satu grafik alir menggunakan kompleksitas siklomatik. *Cyclomatic Complexity* dapat dihitung dalam salah satu dari tiga cara berikut ini :

1. Jumlah daerah—daerah grafik alir yang berhubungan dengan *cyclomatic complexity*.
2. *Cyclomatic complexity* $V(G)$ untuk grafik alir G didefinisikan sebagai :

$$V(G) = E - N + 2$$

Dimana :-E : jumlah edge grafik alir

:-N : jumlah node grafik alir

3. *Cyclomatic complexity* $V(G)$ untuk grafik alir G didefinisikan sebagai :

$$V(G) = P + 1$$

Dimana :-P : jumlah node predikat yang terdapat dalam grafik alir

Selanjutnya untuk perhitungan *cyclomatic complexity* yang dirujuk di grafik alir pada **Gambar 3.93** adalah sebagai berikut :

1. Grafik aliran memiliki 4 *region*.
2. $V(G) = 11 \text{ edge} - 9 \text{ node} + 2 = 4$
3. $V(G) = 3 \text{ node predikat} + 1 = 4$

Oleh karena itu *cyclomatic complexity* dari grafik alir pada **Gambar 3.93** adalah 4

Pada aplikasi penjualan akan dilakukan pengujian *white box* pada proses penjualan yang dilihat dari transaksi yang terjadi,

dengan tujuan untuk mengetahui bahwa program yang dihasilkan telah lulus uji dalam proses penjualan. Untuk melakukan pengujian *whitebox* proses penjualan ini dapat dilihat pada *source code* transaksi pada Gambar 3.98.

```

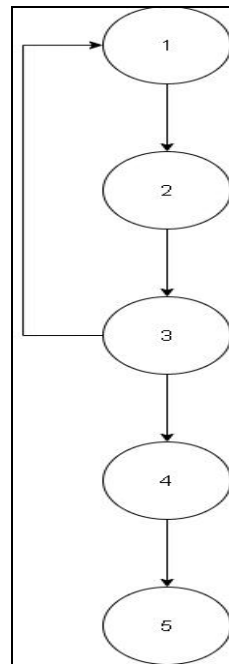
<?php
include "system/proses.php";
$td = date("Y-m-d");
// Nama
$connect = mysqli_connect("localhost", "root", "", "db_penjualan");
$query = "SELECT max(id_transaksi) as maxkode FROM transaksi";
$hasil = mysqli_query($connect, $query);
$kode = mysqli_fetch_array($hasil);
$kodebarang = $data['maxkode'];
$nomor = uniqid($kodebarang, 3);
$nomorutms = $code . "TG";
$nomor = $code . sprintf("%05s", $nomorutms);
$kodeLaba = $code.$nomor.$nomorutms;

?>

<body onload="tuka_tab0">
<div class="tab-content">
<div class="tab-pane active">
<div class="form-content">
<form action="crud/simpan_transaksi.php" method="POST">
<input type="hidden" name="id_user" value="<?php echo $id_ptg; ?>" />
<table>
<tr>
<td><input type="text" name="id_transaksi" id="id_transaksi" class="text disable" value="<?php echo $nomor; ?>" /></td>
</tr>
<tr>
<td><input type="text" name="tanggal" id="tanggal" class="text disable" readonly value="<?php echo $td; ?>" /></td>
</tr>
</table>
<table>
<tr>
<td><input type="text" name="id_kategori" id="id_kategori" onchanges="plgn0">
<option disabled selected="selected" value="<?php echo $id_kategori; ?>" />
</tr>
<tr>
<td><input type="text" name="id_pelanggan" id="id_pelanggan" onkeyup="idpln" class="text disable" autocomplete="off" />
</tr>
<tr>
<td><input type="text" name="nama_pelanggan" id="nama_pelanggan" class="text disable" readonly />
</tr>
</table>
<table>
<tr>
<td><input type="text" name="id_barang" id="id_barang" class="text disable" required="" autocomplete="off" onkeyup="idbd" required="" />
</tr>
<tr>
<td><input type="text" name="nama_barang" id="nama_barang" class="text disable" readonly />
</tr>
<tr>
<td><input type="text" name="harga" id="harga" class="text disable" readonly />
</tr>
<tr>
<td><input type="text" name="jumlah" id="jumlah" class="text disable" required="" autocomplete="off" onkeyup="tdj" />
</tr>
<tr>
<td><input type="text" name="total" id="total" class="text disable" readonly />
</tr>
<tr>
<td><input type="button" name="simpan" class="simpantrans btn btn-primary" value="Simpan" onclick="simpan_detail0" /></td>
</tr>
</table>
</div>
</div>
<div class="tab-pane">
<table>
<tr>
<td><input type="text" name="subtotal" id="subtotal" class="text disable" readonly />
</tr>
<tr>
<td><input type="text" name="diskon" id="diskon" class="text disable" readonly />
</tr>
<tr>
<td><input type="text" name="totalbayar" id="totalbayar" class="text disable" readonly />
</tr>
<tr>
<td><input type="text" name="bayar" id="bayar" class="text disable" onkeyup="bayr0" required="" />
</tr>
<tr>
<td><input type="text" name="kembali" id="kembali" class="text disable" readonly />
</tr>
<tr>
<td><input type="submit" name="simpan" value="Transaksi" class="simpantrans" />
</tr>
</table>
</div>
</div>
</body>

```

Gambar 3.98 Source Code Proses Transaksi



Gambar 3.99 Grafik Alir (*Flowgraph*) Proses Transaksi

Berdasarkan grafik alir (*flowgraph*) diatas, dapat dihitung cyclomatic complexity flowgraph proses transaksi, berikut dengan perhitungan kompleksitas siklomatiknya

$$V(G) = E - N + 2$$

Keterangan : -E : jumlah *edge* grafik alir

: -N : jumlah *node* grafik alir

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

Setelah dihitung nilai *cyclomatic complexity*, didapatkan bahwa $V(G) = 2$, maka dapat disimpulkan bahwa aplikasi memiliki prosedur yang sederhana dan tingkat risiko yang rendah. Tingkat prosedur dan risiko yang didapatkan merujuk pada tabel dibawah ini

Tabel 3.23 Hubungan Cyclomatic Complexity dengan Risiko

| CC | Prosedur | Risiko |
|-------|--|---------------|
| 1-4 | Prosedur yang sederhana | Rendah |
| 5-10 | Prosedur yang terstruktur baik dan stabil | Rendah |
| 11-20 | Prosedur yang lebih kompleks | Sedang |
| 21-50 | Prosedur yang rumit dan mengkhawatirkan | Tinggi |
| >50 | Prosedur yang rawan kesalahan, bermasalah, dan tidak dapat diuji | Sangat Tinggi |

Path yang dihasilkan yaitu :

Tabel 3.24 Path Proses Transaksi yang dihasilkan

| <i>Path</i> | Hasil | Tipe Prosedur dan Risiko |
|-------------|-----------|-----------------------------------|
| 1 | 1-2-3-4-5 | Prosedur Sederhana, risiko rendah |
| 2 | 1-2-3-1 | Prosedur Sederhana, risiko rendah |

Berdasarkan pengujian *white box* yang telah dilakukan maka disimpulkan Aplikasi Penjualan ini memiliki prosedur yang sederhana dan tingkat risiko yang rendah.