

BAB III

METODOLOGI PENELITIAN

3.1 Umum

Pada bab ini akan dijelaskan mengenai pembuatan Rancang Bangun Aplikasi Perencanaan Stok Barang dengan Menggunakan Teori Trafik dari tahap awal perancangan sampai dengan perancangan aplikasi diselesaikan. Tahapan perancangan aplikasi ini dimulai dengan studi pustaka, perancangan aplikasi, serta pengujian dan evaluasi aplikasi. Studi pustaka dilakukan dengan pengumpulan data-data untuk perancangan aplikasi berupa teori perencanaan stok barang dan teori trafik. Perancangan aplikasi ini dilakukan dengan pembuatan *interface* dan pengaplikasian rumus-rumus dengan bantuan aplikasi NetBeans IDE 8.1. Pengujian dan evaluasi dilakukan dengan menginstal aplikasi yang telah dirancang ke perangkat yang mendukung aplikasi ini, setelah itu dilakukan evaluasi kinerja aplikasi tersebut.

3.2 Studi Pustaka

Metode studi pustaka yang dilakukan pada perancangan aplikasi ini, yaitu dengan melakukan studi literatur, diskusi, dan wawancara. Studi literatur dilakukan dengan mempelajari dan mengumpulkan informasi melalui literatur-literatur yang sudah ada berupa buku, jurnal, laporan penelitian, serta literatur pendukung lain yang diperoleh dari internet. Diskusi dilakukan dengan melakukan pembahasan mengenai aplikasi yang dirancang dan pencarian solusi dari masalah yang dihadapi selama perancangan bersama dengan beberapa rekan dari program studi teknik elektro dan dosen pembimbing. Metode wawancara dilakukan dengan menanyakan kepada beberapa pihak yang dapat memberikan informasi yang dapat dipercaya dan dipertanggungjawabkan, seperti pemilik usaha, penyedia barang, dan dosen.

Dari studi pustaka yang telah dilakukan maka diperoleh informasi yang diperlukan untuk membantu perancangan aplikasi ini. Selain itu juga diperoleh informasi mengenai teori yang dapat diterapkan pada aplikasi yang telah dirancang, seperti rumus-rumus yang dapat digunakan pada aplikasi ini, metode pengembangan aplikasi, teknik-teknik perancangan aplikasi, dan lain-lain.

3.3 Kebutuhan Sistem

3.3.1 Alat

Dalam penelitian pada tugas akhir ini dibutuhkan beberapa alat yang digunakan untuk menunjang keberhasilan perancangan aplikasi ini. Alat yang dibutuhkan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*), antara lain:

1. Perangkat Keras

Perangkat keras (*hardware*) yang digunakan dalam perancangan aplikasi ini berupa *Personal Computer* (PC) atau Laptop dengan spesifikasi sebagai berikut:

- a. Sistem operasi Windows 7/8/10.
- b. *Architecture processor 32/64 bit.*
- c. *Random Access Memory* (RAM) 2-8 GB.

2. Perangkat Lunak

Perangkat lunak (*software*) yang digunakan di dalam perancangan aplikasi ini antara lain:

- a. NetBeans IDE 8.
- b. *Windows OS.*
- c. Microsoft Visio.

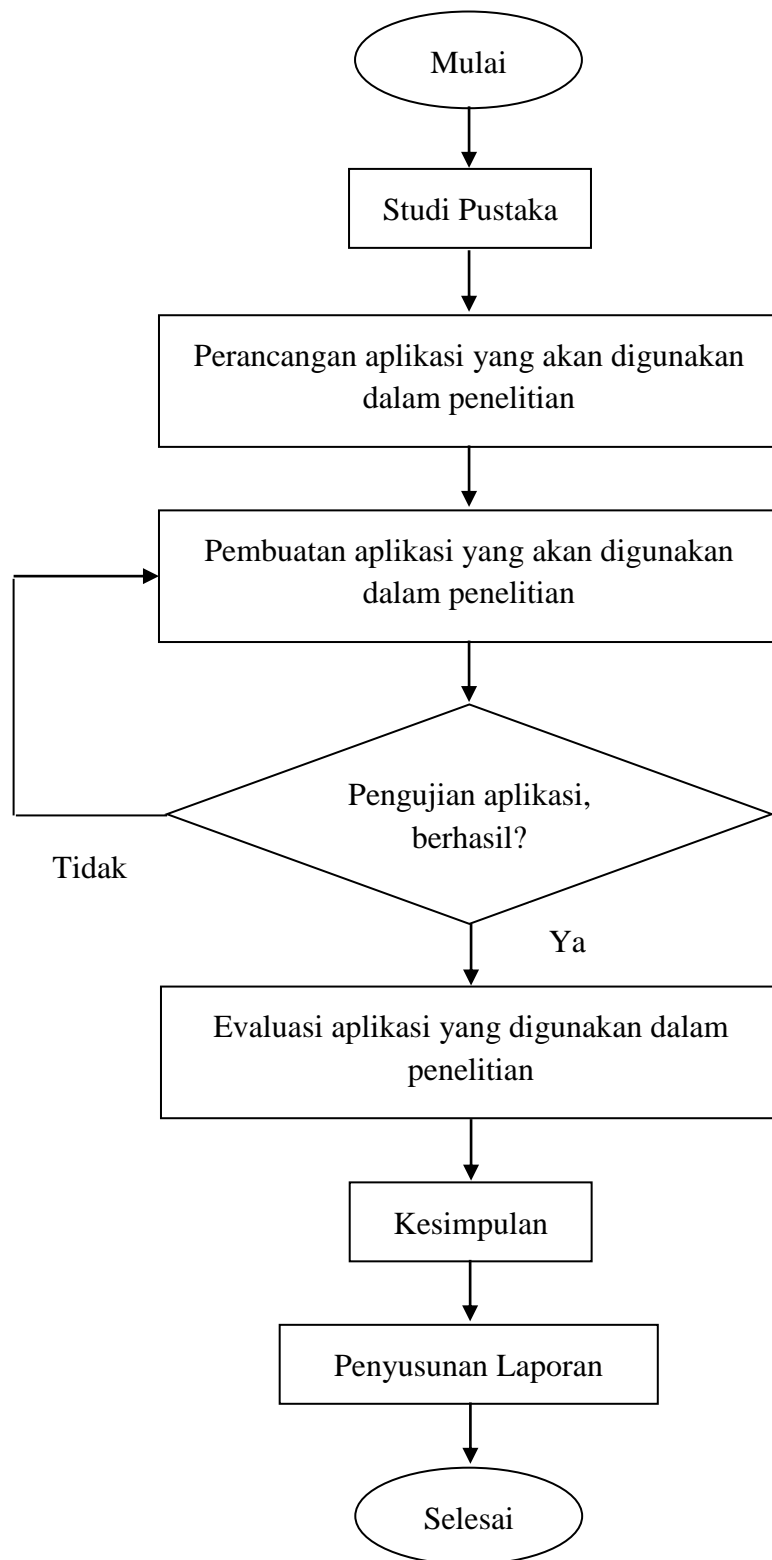
3.3.2 Bahan

Adapun bahan yang dibutuhkan dalam pembuatan aplikasi pada penelitian ini, antara lain:

1. Data yang diperoleh melalui *study literature* berdasarkan pada penelitian sebelumnya. Data yang diperoleh berupa data perancangan yang masih memiliki keterikatan dengan perancangan yang akan dilakukan. Dari data tersebut diperoleh pemahaman mengenai persyaratan dan kebutuhan aplikasi yang akan dirancang.
2. Data arsip barang masuk, stok barang, dan barang yang keluar (baik barang yang terjual maupun yang belum) dari Kantin Fakultas Teknik UMY, serta hasil wawancara dengan pihak manajemen Kantin Fakultas Teknik UMY.
3. Data yang dibuat guna memenuhi kebutuhan pengujian aplikasi. Data yang diperoleh merupakan data acak yang akan diujikan pada aplikasi yang telah dirancang.

3.4 Alur Penelitian

Dalam perancangan aplikasi ini metode yang digunakan ialah metode rekayasa. Menurut Wijaya (2007 dalam Wijaya, 2016:29) mendefinisikan “Kegiatan rekayasa merupakan penelitian yang menerapkan ilmu pengetahuan menjadi satu rancangan guna mendapatkan kinerja sesuai dengan persyaratan yang ditentukan”. Diagram alur penelitian ini dapat dilihat dari gambar 3.1 berikut:



Gambar 3.1 Alur Penelitian

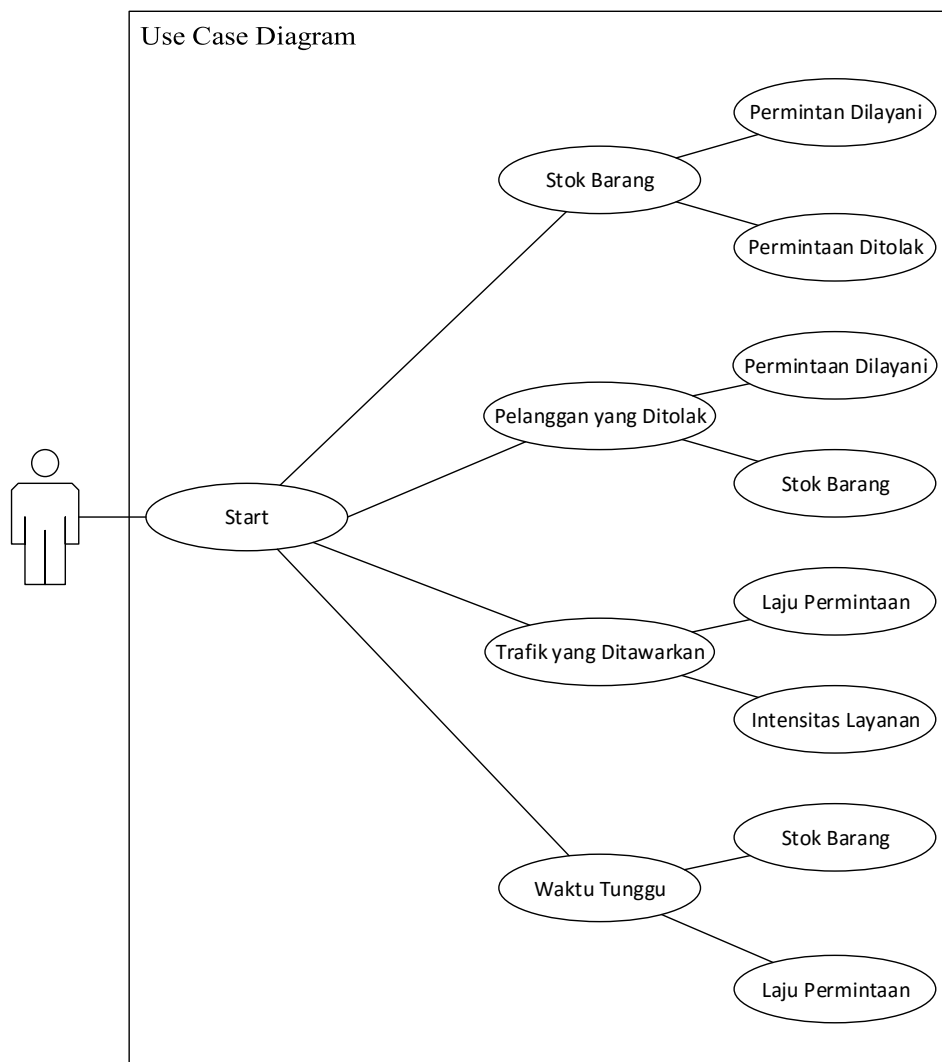
Perancangan penelitian ini dimulai dengan studi pustaka yang bertujuan untuk mengumpulkan data-data yang diperlukan dalam penelitian ini. Proses pengumpulan data yang dilakukan meliputi kegiatan studi literatur, wawancara, dan diskusi. Data yang diperoleh selanjutnya dipelajari dan diolah untuk dikembangkan ke dalam sebuah *software*. Kegiatan selanjutnya adalah kegiatan perancangan aplikasi yang meliputi proses pengumpulan kebutuhan aplikasi, perancangan sistem, antarmuka, algoritma prosedural, alur kerja, dan sebagainya. Perancangan ini dilakukan untuk mempermudah dalam pembuatan aplikasi dan sebagai *blueprint* aplikasi yang akan dibuat. Setelah perancangan selesai, tahap selanjutnya adalah pembuatan aplikasi dengan bantuan *software* NetBeans IDE. Proses pembuatan ini meliputi pembuatan *interface* dan pengodean (*coding*) yang dibuat berdasarkan rancangan yang telah dibuat.

Pengujian aplikasi dilakukan setelah aplikasi selesai dibuat. Pengujian aplikasi ini dibagi dalam dua bentuk pengujian. Pengujian pertama dilakukan untuk mengetahui tingkat keberhasilan pengoperasian aplikasi. Pada pengujian ini aplikasi akan dicoba dijalankan pada beberapa perangkat yang berbeda dan selanjutnya akan dicatat tingkat keberhasilannya. Pengujian kedua adalah pengujian tingkat keakuratan dan ketepatan perhitungan dengan menggunakan aplikasi ini. Pada percobaan kedua data yang digunakan dalam pengujian menggunakan data riil yang diperoleh dari data stok barang Kantin Fakultas Teknik UMY dan data yang dibangkitkan secara acak untuk tujuan pengujian aplikasi. Indikator keberhasilan pengujian, yaitu aplikasi ini dapat dijalankan dalam beberapa perangkat dan tidak terjadi kesalahan pada hasil perhitungan. Setelah pengujian, kegiatan selanjutnya adalah evaluasi.

Evaluasi merupakan kegiatan yang bertujuan untuk menilai tingkat keberhasilan aplikasi yang telah dibuat dengan memperhatikan peluang pengembangan dan kekurangan aplikasi ini. Evaluasi dilakukan berdasarkan data yang diperoleh dari hasil pengujian aplikasi. Setelah evaluasi, tahapan terakhir yang dilakukan ialah tahap pembuatan laporan. Laporan ini berisikan data yang diperoleh selama penelitian dilakukan.

3.5 Use Case Diagram

Use case diagram adalah diagram yang menjelaskan aktor dan interaksi apa saja yang terdapat pada sistem. *Use case diagram* menjelaskan secara singkat hubungan antara *use case*, aktor, dan sistem yang terlibat di dalam sebuah aplikasi. Berikut ini adalah gambar *use case diagram* aplikasi pada sistem yang telah dirancang.



Gambar 3.2 Use Case Diagram

Dari gambar 3.2 di atas diperoleh penjelasan *use case diagram* yang dirancang ini, antara lain:

1. Pada *use case diagram* yang dibuat terdapat satu aktor yang menggunakan sistem yaitu *user/pengguna*.
2. Sistem mulai berjalan saat tombol *start* dipilih.
3. Pengguna pada sistem yang dibuat memiliki wewenang mengoperasikan aplikasi untuk melakukan kegiatan penghitungan stok barang, pelanggan yang ditolak, trafik yang ditawarkan, serta waktu tunggu.

3.6 Perancangan Antarmuka (*Interface*)

Perancangan *user interface* bertujuan untuk menghubungkan pengguna dengan fitur yang disediakan oleh sistem sehingga pengguna mengerti dan dapat menggunakan sistem tersebut. Hasil rancangan *user interface* diharapkan mampu memberikan gambaran mengenai fitur-fitur yang terdapat pada sistem.

3.6.1 Perancangan Antarmuka Tampilan Awal

Tampilan awal adalah halaman yang pertama kali muncul pada saat aplikasi dijalankan. Gambar rancangan antarmuka tampilan awal dapat dilihat pada gambar berikut:

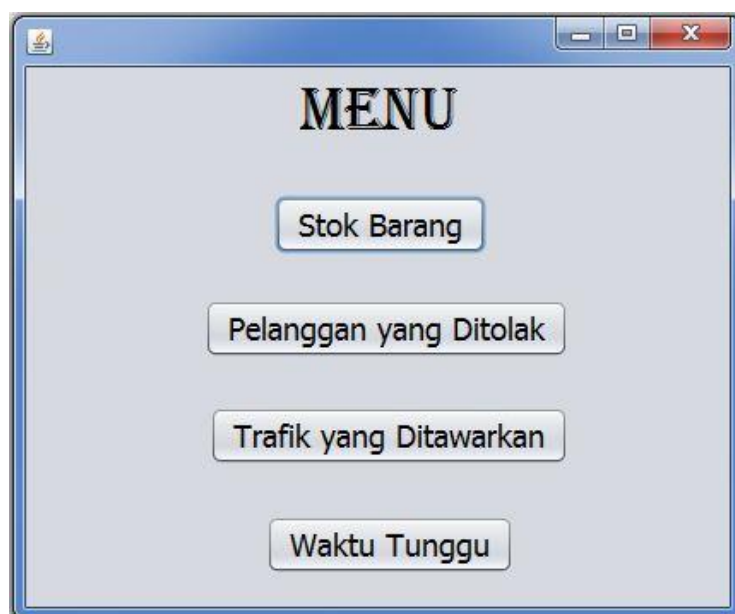


Gambar 3.3 Rancangan Antarmuka Tampilan Awal

Di halaman tampilan awal terdapat tombol 'Start' yang berfungsi untuk memulai menjalankan aplikasi oleh pengguna. Selain itu, pada halaman ini terdapat teks nama aplikasi yang dibuat.

3.6.2 Perancangan Antarmuka Menu Utama

Rancangan antarmuka menu utama adalah halaman yang menampilkan pilihan perhitungan yang tersedia pada aplikasi ini. Gambar rancangan antarmuka yang dibuat terlihat pada gambar 3.4 berikut:



Gambar 3.4 Rancangan Antarmuka Menu Utama

Halaman menu utama bertindak sebagai halaman navigasi yang akan membantu pengguna untuk memilih perhitungan yang diinginkan. Pada halaman ini terdapat beberapa tombol yang berfungsi untuk menampilkan halaman perhitungan. Pilihan perhitungan yang terdapat pada halaman ini berjumlah sebanyak empat model perhitungan. Pilihan perhitungan tersebut, yaitu stok barang, pelanggan yang ditolak, trafik yang ditawarkan, dan waktu tunggu. Dengan adanya menu utama diharapkan dapat mempermudah dalam penggunaan aplikasi ini.

Berikut ini merupakan tabel penjelasan singkat antarmuka pilihan perhitungan yang tersedia pada halaman menu utama beserta dengan masing-masing fungsinya:

Tabel 3.1 Fungsi Rancangan Antarmuka Menu Utama

No.	Tombol	Fungsi
1.	Stok Barang	Menampilkan halaman perhitungan stok barang.
2.	Pelanggan yang Ditolak	Menampilkan halaman perhitungan pelanggan yang ditolak.
3.	Trafik yang Ditawarkan	Menampilkan halaman perhitungan trafik yang ditawarkan.
4.	Waktu Tunggu	Menampilkan halaman perhitungan waktu tunggu.

3.6.3 Perancangan Antarmuka Stok Barang

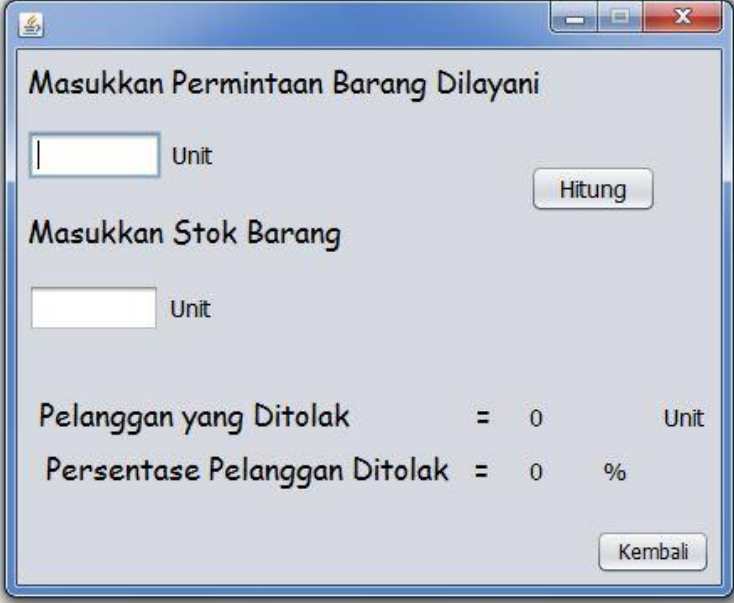
Rancangan antarmuka stok barang adalah halaman antarmuka yang berfungsi sebagai tempat *user* melakukan proses perhitungan stok barang dengan dua *input* dan satu *output*. Rancangan antarmuka yang dibuat dapat dilihat pada gambar 3.5 berikut:

Gambar 3.5 Rancangan Antarmuka Stok Barang

Pada halaman ini terdapat beberapa fasilitas berupa *input*, *output*, dan tombol hitung. *Input* pada halaman ini berupa permintaan barang dilayani dan barang yang ditolak. Untuk *ouput* yang ditampilkan berupa stok barang yang disediakan pengguna. Proses perhitungan terjadi apabila tombol “hitung” ditekan sehingga *input* akan diproses dan hasilnya akan ditampilkan sebagai *output*.

3.6.4 Perancangan Antarmuka Pelanggan yang Ditolak

Rancangan antarmuka pelanggan yang ditolak adalah halaman yang berfungsi sebagai tempat pengguna melakukan proses perhitungan pelanggan yang ditolak dengan dua *input* dan dua *output*. Gambar rancangan antarmuka yang dibuat terlihat pada gambar 3.6 berikut:



The image shows a software window titled "Masukkan Permintaan Barang Dilayani". It contains two input fields, each labeled "Unit". The first input field is empty, and the second is also empty. To the right of the first input field is a button labeled "Hitung". Below the input fields, there are two lines of text: "Pelanggan yang Ditolak = 0 Unit" and "Persentase Pelanggan Ditolak = 0 %". At the bottom right of the window is a button labeled "Kembali".

Gambar 3.6 Rancangan Antarmuka Pelanggan yang Ditolak

Halaman antarmuka pelanggan yang ditolak memiliki beberapa fasilitas yang disediakan berupa *input*, *output*, dan tombol hitung. *Input* pada halaman ini berupa permintaan barang dilayani dan stok barang. Untuk *ouput* yang ditampilkan berupa pelanggan yang ditolak dan persentase pelanggan ditolak. Proses perhitungan terjadi apabila tombol “hitung” ditekan sehingga *input* akan diproses dan hasilnya akan ditampilkan sebagai *output*.

3.6.5 Perancangan Antarmuka Trafik yang Ditawarkan

Rancangan antarmuka trafik yang ditawarkan merupakan halaman yang berfungsi sebagai tempat pengguna untuk melakukan proses perhitungan nilai trafik yang ditawarkan. Pada halaman ini terdapat tiga area dimana dua area digunakan untuk memberi nilai *input* dan satu area digunakan untuk menampilkan nilai *output*. Gambar rancangan antarmuka (*interface*) yang dibuat dapat dilihat pada gambar 3.7 berikut:



The image shows a software window titled "Masukkan Laju Permintaan" (Enter Request Rate). It contains two input fields: the first is labeled "Unit/Hari" and the second is labeled "Hari". A "Hitung" (Calculate) button is positioned to the right of the first input field. Below the second input field, the text "Trafik yang Ditawarkan = 0 Erlang" is displayed. A "Kembali" (Back) button is located at the bottom right of the window.

Gambar 3.7 Rancangan Antarmuka Trafik yang Ditawarkan

Di dalam rancangan antarmuka trafik yang ditawarkan terdapat beberapa fasilitas yang disediakan berupa *input*, *output*, dan tombol hitung. Pada halaman ini terdapat dua variabel *input*, yaitu laju permintaan dan waktu layanan. Berbeda dengan *input* pada halaman lain, *input* pada halaman ini menggunakan tipe data *float*. *Output* yang dihasilkan dari perhitungan di halaman ini adalah nilai trafik yang ditawarkan. Proses perhitungan di halaman ini dapat terjadi jika tombol "hitung" ditekan oleh pengguna sehingga data yang sebelumnya telah dimasukkan sebagai *input* akan diproses oleh sistem dan hasil perhitungan tersebut akan ditampilkan sebagai *output* sistem.

3.6.6 Perancangan Antarmuka Waktu Tunggu

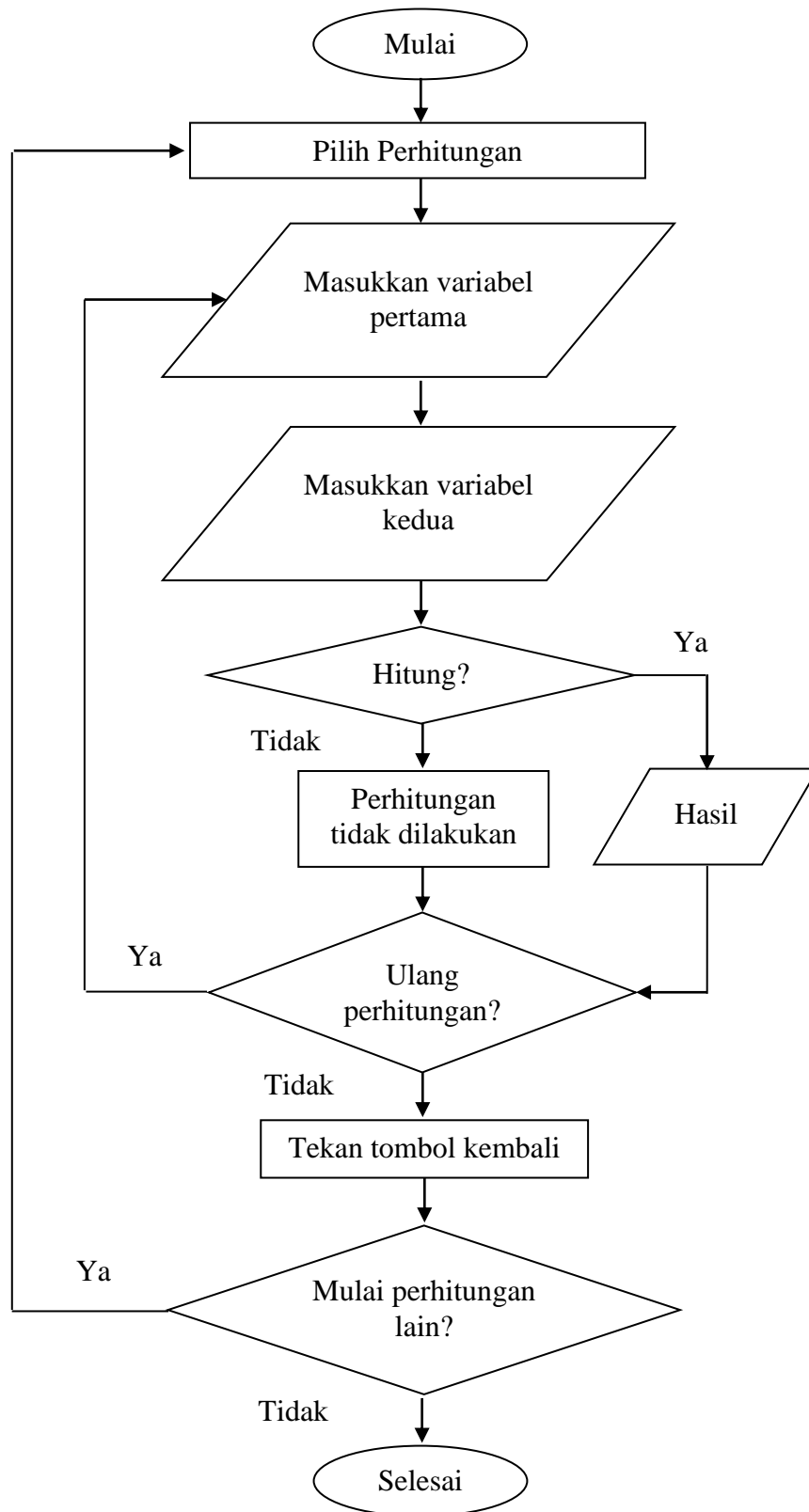
Antarmuka waktu tunggu adalah halaman yang berfungsi sebagai tempat *user* melakukan proses perhitungan waktu tunggu dengan dua *input* dan satu *output*. Gambar rancangan antarmuka yang dibuat terlihat pada gambar berikut:

Gambar 3.8 Rancangan Antarmuka Waktu Tunggu

Halaman antarmuka waktu tunggu memiliki beberapa fasilitas yang telah disediakan berupa *input*, *output*, dan tombol hitung. Pada halaman ini terdapat dua variabel *input*, yaitu laju permintaan dan stok barang. *Output* proses berupa waktu tunggu mendapatkan barang. Proses perhitungan dapat terjadi apabila pengguna menekan tombol “hitung” sehingga sistem selanjutnya akan memproses *input* dan hasil dari proses tersebut akan ditampilkan sebagai *output* perhitungan.

3.7 Flowchart Sistem

Flowchart sistem dirancang untuk menentukan alur kerja dan kebutuhan sistem. Dari *use case diagram* dapat dibuat *flowchart* sistem. *Flowchart* sistem menjelaskan mengenai tahapan yang dialami sistem ketika *user* mulai menjalankan aplikasi ini sampai selesai. Berikut ini adalah *flowchart* sistem yang dirancang pada aplikasi ini.

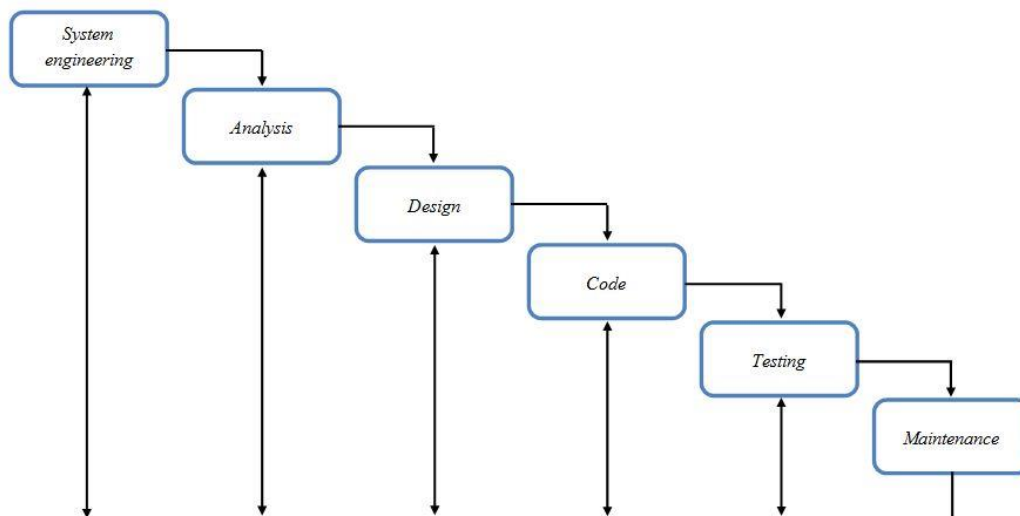
Gambar 3.9 *Flowchart* Sistem

Dari gambar 3.9 dapat dilihat kegiatan yang dilakukan oleh sistem. Pada saat aplikasi mulai dijalankan maka kegiatan pertama yang akan dilakukan oleh sistem adalah proses pemilihan perhitungan. Sistem akan memulai tahap pemasukan variabel *input* untuk memulai proses perhitungan setelah *user* menentukan pilihan perhitungan yang akan dilakukan. Jumlah variabel *input* sebanyak dua variabel. Dengan memasukkan dua variabel *input* maka sistem akan memulai proses perhitungan. Pada proses ini, jika *user* menekan tombol ‘hitung’ maka akan dimunculkan hasil perhitungan dan jika *user* tidak menekan tombol ‘hitung’ maka perhitungan tidak terjadi.

Kegiatan yang diproses sistem selanjutnya adalah proses perhitungan ulang. Pada tahap ini jika *user* ingin melakukan perhitungan ulang maka *user* dapat memasukkan dua variabel baru dan proses perhitungan akan berulang. Akan tetapi jika *user* tidak ingin melakukan perhitungan lagi maka *user* dapat menekan tombol kembali. Aktivitas terakhir yaitu proses pemilihan perhitungan lainnya yang terdapat pada menu utama. Pada saat *user* ingin melakukan perhitungan lain maka sistem akan kembali memulai proses pemilihan perhitungan dan proses akan terjadi seperti yang telah dijelaskan sebelumnya. Namun, jika *user* tidak ingin melakukan perhitungan lain maka aktivitas yang sedang berjalan pada sistem akan selesai (*stop*).

3.8 Model Pengembangan Perangkat Lunak

Dalam perancangan aplikasi ini model pengembangan perangkat lunak yang dilakukan adalah *linear sequential model*. Model ini lebih dikenal dengan sebutan model *the classic life cycle* atau model *waterfall* (air terjun). Model pengembangan perangkat lunak ini merupakan model pengembangan yang paling tua dan yang paling banyak digunakan. Model ini menggunakan pendekatan perangkat lunak yang sederhana, yaitu pendekatan yang sistematis dan urut. Pada model ini pengerjaan dimulai dari tahap analisis, desain, pengodean, pengujian, hingga tahap pemeliharaan.



Gambar 3.10 Model Pengembangan *Waterfall*

Dari gambar di atas diperoleh bahwa pada metode ini kegiatan perancangan aplikasi dilaksanakan secara bertahap dan sistematis. Berikut ini adalah penjelasan mengenai tahapan-tahapan model pengembangan *waterfall*:

3.8.1 *System Engineering* (Rekayasa Sistem)

Rekayasa sistem merupakan tahapan pertama yang dilakukan dalam perancangan aplikasi ini. Pada tahap ini dilakukan pembangunan keseluruhan sistem yang akan dikembangkan. Selain itu, di tahap ini dilakukan pemilihan bagian yang akan dikembangkan pada aplikasi ini.

3.8.2 *Software Requirements Analysis* (Analisis Kebutuhan Perangkat Lunak)

Analisis yang dilakukan pada tahap ini yaitu analisis kebutuhan dari keseluruhan sistem yang dirancang. Kebutuhan ini meliputi kebutuhan antarmuka, fungsi yang dibutuhkan, serta batasan perangkat lunak. Pada tahap ini dilakukan wawancara dan diskusi untuk memperoleh informasi mengenai kebutuhan pengguna (*user*). Pada tahap ini juga dilakukan analisis keadaan-keadaan yang diperkirakan akan muncul selama pengembangan aplikasi ini. Hal ini bertujuan untuk mengantisipasi hal-hal yang sekiranya dapat mengganggu kinerja aplikasi yang dirancang.

3.8.3 *Design* (Desain)

Tahap ini bertujuan untuk membuat sebuah *blueprint* aplikasi. *Blueprint* ini dibuat berdasarkan kebutuhan yang diperoleh pada tahap analisis kebutuhan perangkat lunak sebelum memulai tahapan *coding*. Pada tahapan ini dibuat desain representasi *interface*, diagram kerja sistem, dan algoritma prosedural dari aplikasi yang dikembangkan. Tahapan desain ini menjadi tahapan yang sangat penting karena tahapan ini merupakan tahapan terakhir sebelum dimulainya tahapan *coding*.

3.8.4 *Coding* (Pengodean/Pemrograman)

Tahapan ini merupakan tahapan di mana aplikasi mulai dibuat. Pada tahapan ini dilakukan proses pengodean desain aplikasi yang telah dibuat sebelumnya. Tahapan ini bertujuan agar desain yang telah dibuat dapat diterjemahkan ke dalam sebuah bahasa pemrograman yang dapat dimengerti oleh mesin. Bahasa pemrograman yang digunakan dalam aplikasi ini adalah bahasa pemrograman Java. Tahapan ini dilakukan dengan melakukan penyesuaian dengan tahapan-tahapan yang telah dilakukan sebelumnya. *Coding* pada tahap ini dilakukan dengan bantuan aplikasi NetBeans IDE yang biasa digunakan untuk membuat aplikasi.

3.8.5 *Testing* (Pengujian)

Pengujian dilakukan setelah tahapan rekayasa sistem, analisis kebutuhan perangkat lunak, desain, dan pengodean. Hal ini karena pada tahapan ini akan dilakukan uji coba keberhasilan aplikasi yang telah dirancang pada tahap sebelumnya. Adapun tujuan dari tahapan ini adalah untuk mengetahui unjuk kerja aplikasi. Pada tahap ini dapat diketahui apakah masih ada kesalahan (*error*) dan kendala yang diperoleh pengguna (*user*) pada saat menjalankan aplikasi ini. Dari data tersebut dapat dilakukan pengembangan aplikasi kedepannya guna mengatasi masalah tersebut. Data pada tahap ini merupakan data riil yang diperoleh dari hasil pengujian aplikasi yang telah dibuat secara langsung.

Pengujian aplikasi ini dilakukan dengan mencoba melakukan perhitungan secara manual dan perhitungan secara otomatis dengan menggunakan aplikasi. Data hasil perhitungan yang diperoleh dibandingkan antara perhitungan manual dengan perhitungan menggunakan aplikasi. Dari perbandingan tersebut diperoleh keakuratan dan ketepatan perhitungan yang dilakukan oleh aplikasi ini. Data yang digunakan untuk pengujian ini berupa data yang diperoleh dari Kantin Fakultas Teknik UMY dan data yang dibangkitkan oleh penulis guna untuk melakukan uji coba aplikasi.

3.8.6 *Maintenance* (Pemeliharaan)

Setelah semua tahapan di atas dilakukan, tahapan selanjutnya adalah tahapan pemeliharaan *software*. Tahapan ini dilakukan guna memelihara kinerja dari aplikasi yang telah dibuat agar dapat bekerja secara maksimal. Selain kegiatan pemeliharaan aplikasi, kegiatan pengembangan aplikasi juga termasuk ke dalam tahap ini. Pengembangan aplikasi diperlukan karena aplikasi yang dirancang tidak selamanya dapat memenuhi kebutuhan *user*. Hal ini karena kebutuhan *user* akan semakin meningkat seiring perkembangan zaman. Pengembangan aplikasi ini dapat dilakukan untuk menyesuaikan kebutuhan *user* atau untuk kebutuhan riset. Pengembangan dilakukan dengan harapan agar aplikasi yang dikembangkan dapat mengatasi kesalahan yang terdapat pada aplikasi yang telah dibuat sebelumnya.

3.9 Metode Pengujian Perangkat Lunak

Metode pengujian perangkat lunak merupakan suatu teknik atau cara yang dilakukan untuk menguji sebuah perangkat lunak. Metode pengujian tersebut harus memiliki mekanisme yang tepat dalam menentukan data uji sehingga dapat dilakukan pengujian perangkat lunak secara lengkap dengan kemungkinan kesalahan (*error*) yang paling sedikit. Pengujian yang baik mampu mencakup semua aspek pengoperasian, mencakup sebanyak mungkin jalur yang dirancang, serta tidak terlalu rumit dan tidak terlalu sederhana.

Pengujian yang dilakukan pada penelitian ini menggunakan *black box testing*. *Black box testing* atau disebut juga tes fungsional merupakan pengujian yang berfokus pada persyaratan fungsional dari perangkat lunak dengan melihat tampilan luarnya saja sama dengan pengujian *black box*. Hasil pengujian diperoleh dengan hanya mengamati hasil eksekusi *software* melalui data uji coba. Hasil pengujian tersebut berfungsi untuk mengetahui aplikasi telah sesuai dengan tahap pemodelan atau tidak. Pengujian ini dilakukan dengan mengevaluasi hanya *interface* (tampilan luar) dan fungsionalitas aplikasi tanpa mengetahui apa yang terjadi dalam proses secara detail (hanya mengetahui masukan dan keluaran). Pengujian yang dilakukan melibatkan beberapa orang sebagai pemberi masukan tertentu dan selanjutnya melihat hasil keluaran dari masukan tersebut.

Dalam pengujian yang dilakukan terdapat beberapa hal yang dijadikan sebagai kriteria pengujian. Kriteria tersebut berupa kesalahan yang dijadikan tolak ukur untuk memenuhi tujuan awal pembuatan aplikasi. Kriteria-kriteria tersebut antara lain:

1. Fungsi-fungsi yang *error* dan hilang.
2. Kesalahan *interface* untuk menjalankan fungsinya.
3. Kesalahan kinerja sistem menjalankan fungsinya.
4. Kesalahan sistem untuk memproses masukan (*input*).
5. Kesalahan sistem untuk menampilkan keluaran (*output*).