

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Sejarah Singkat Perusahaan**

CV. Smart Motecare Mandiri adalah sebuah unit usaha yang didirikan pada tanggal 23 juni 2008 oleh Bapak April Lia Hanato, S.Kom., M.Kom. selaku pemilik perusahaan. Perusahaan ini merupakan unit usaha yang bergerak dibidang jasa layanan. Teknologi Informasi(TI) berupa pengembangan software, pengadaan hardware, *maintenance*, *service*, *training*, *networking* dan *IT consultance* beserta aplikasi/perangkat pendukungnya. dengan dukungan tenaga propesional dibidangnya.

#### **3.2 Nama dan Logo Perusahaan**

Nama Perusahaan : **CV. Smart Motecare Mandiri (SMI)**



Gambar 3.1 Logo CV. Smart Motecare Mandiri

#### **3.3 Profil Perusahaan**

CV. Smart Motecare Mandiri adalah sebuah unit usaha yang didirikan pada tanggal 23 juni 2008 oleh Bapak April Lia Hanato, S.Kom., M.Kom. selaku pemilik perusahaan. Perusahaan ini merupakan unit usaha yang bergerak dibidang jasa layanan Teknologi Informasi(TI) berupa pengembangan software, pengadaan hardware, *maintenance*, *service*, *training*, *networking* dan *IT consultance* beserta aplikasi/perangkat pendukungnya. dengan dukungan tenaga propesional dibidangnya. CV. Smart Motecare Mandiri melakukan kerja sama dengan perusahaan - perusahaan industri maupun jasa di kawasan industri yang berada di daerah Karawang, yang meliputi pengadaan barang seperti perangkat komputer, cctv, instalasi jaringan dan *Maintenance IT*.

CV. Smart Motecare Mandiri berusaha mewujudkan visi untuk menjadi perusahaan penyedia jasa total solusi IT bersekala nasional. CV. Smart Motecare Mandiri berkeinginan membangun keinginannya berdasarkan visi yang kuat, tenaga kerja yang tangguh dan kerjasama yang solid dengan mitra-mitranya. Komitmen CV. Smart Motecare Mandiri adalah mencapai keberhasilan dengan memberikan solusi yang dapat digunakan oleh mitra-mitranya dengan pencapaian mereka lebih cepat, efektif dan efisien.

Adapun Perusahaan atau Instansi yang menjalin kerja sama pengadaan Barang dan Jasa *Maintenance* IT dengan CV. Smart Motecare Mandiri adalah sebagai berikut:

1. DINAS PERINDUSTRIAN DAN PERDAGANGAN  
Alamat Kantor Pemerintahan Kabupaten Karawang
2. UNIVERSITAS BUANA PERJUANGAN KARAWANG  
Alamat Jl. H.S. Ronggowaluyo, Teluk Jambe, Karawang.
3. PT. FUJI SPRING INDONESIA  
Alamat Kawasan Karawang Internasional Industri City (KIIC)
4. PT. ISK IINDONESIA  
Alamat Kawasan Karawang Internasional Industri City (KIIC)
5. PT. KOYAMA INDONESIA  
Alamat Kawasan Karawang Internasional Industri City (KIIC)
6. PT. ABE KOGYO INDONESIA  
Alamat Kawasan Karawang Internasional Industri City (KIIC)
7. PT. CHUNPAO STEEL INDONESIA  
Alamat Kawasan Industri Suryacipta - Karawang (SURYACIPTA)
8. PT. ROCK PAINT INDONESIA  
Alamat Kawasan Karawang Internasional Industri City (KIIC)
9. PT. TOTOKU TORYO INDONESIA  
Alamat Kawasan Karawang Internasional Industri City (KIIC)
10. PT. FUJITA INDONESIA  
Alamat Kawasan Karawang Internasional Industri City (KIIC)

### **3.4 Visi dan Misi Perusahaan**

Pengembangan operasional CV. Smart Motecare Mandiri selalu berpedoman pada Visi dan Misi yang membantu perusahaan tetap fokus dalam meraih pencapaian sebuah keberhasilan. Visi dan Misi ini membantu CV. Smart Motecare Mandiri untuk selalu berupaya pencapaian target yang sudah ditentukan dengan mengingatkan manajemen, layanan serta kepuasan terhadap pelanggan.

Berikut Visi dan Misi CV. Smart Motecare Mandiri :

#### **Visi**

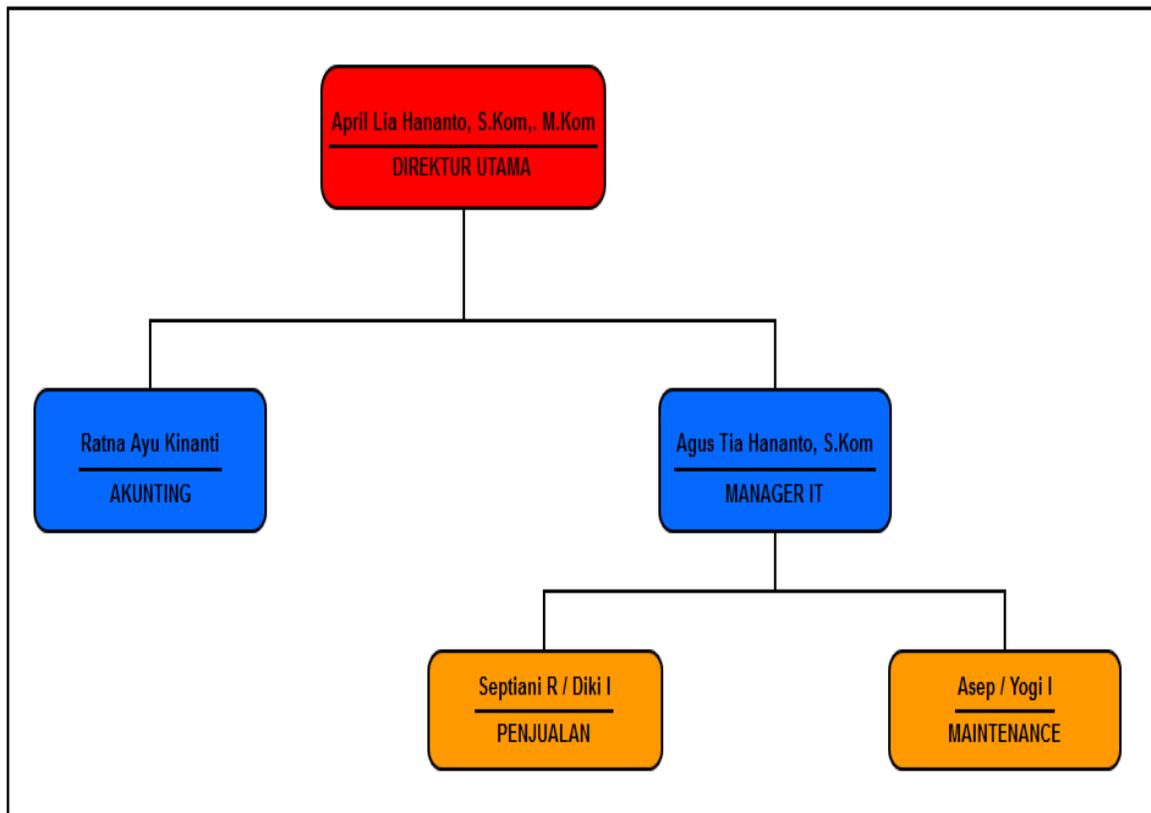
Menjadi setara IT terbesar yang menyediakan jasa layanan Teknologi Informasi dengan kualifikasi dan kompetensi internasional yang berorientasi pada kepuasan pelanggan.

#### **Misi**

1. Menyelenggarakan jasa layanan komputer berorientasi pada kepuasan pelanggan (Customer Satisfaction)
2. Menyediakan sarana dan lingkungan yang kondusif bagi pelaksanaan kegiatan jasa layanan komputer menjaga keterkaitan dan relevansi seluruh kegiatan jasa layanan komputer dengan kebutuhan pasar, serta mengaktispasi maraknya perasingan bisnis dalam globalisasi kehidupan masyarakat.
3. Mengembangkan reset operasi berkesinambungan dan terarah secara jelas untuk meningkatkan kopetensi.
4. Melangsungkan dan meningkatkan kerjasama dengan pihak luar, sehingga mampu meningkatka produktivias dan pelayanan.
5. Meningkatkan *brand image* CV. Smart Motecare Mandiri.

### **3.5 Struktur Organisasi Perusahaan**

Berikut Adalah Struktur Organisasi CV. Smart Motecare Mandiri



Gambar 3.2 Struktur Organisasi CV. Smart Motecare mandiri.

### 3.6 Metode yang dipakai

Dalam menyelesaikan perancangan dan penulisan tugas akhir ini maka peneliti menggunakan beberapa metode penelitian untuk mendapatkan data yang diperlukan, sehingga dapat diperoleh suatu hasil yang sesuai seperti yang diharapkan penulis.

Adapun metodologi penelitian yang digunakan peneliti adalah :

- **Metode Pengumpulan Data**

Data untuk mendapatkan data yang diperlukan dalam rangka penyusunan penelitian laporan skripsi, penulis menggunakan beberapa metode yang digunakan sebagai berikut :

#### A. Metode Interview (wawancara)

Metode yang dilakukan penulis untuk mendapatkan data dengan cara wawancara atau tanya jawab secara tertulis baik itu kepada staff akunting yaitu Ibu Ratna ayu kinanti dan kepada Manager IT yaitu Bapak Agustia Hananto, S.Kom. Untuk mendapatkan data-data yang diperlukan, pertanyaan-pertanyaan yang diajukan tentunya berkaitan dengan perumusan masalah yang ada. Diantaranya yaitu berkaitan bagaimana sistem yang berjalan dan apa saja kelemahannya. Dari kelemahan inilah peneliti dapat mencari solusi untuk merancang sistem yang baru yang sesuai dengan kebutuhan dari CV. Smart Motecare Mandiri.

#### B. Metode Observasi

Suatu metode untuk mendapatkan data dengan jalan pengamatan dan untuk mencari sumber informasi dengan melakukan evaluasi terhadap masalah yang ada dan melaksanakan pencatatan secara sistematis terhadap unsur-unsur yang diteliti. Observasi dilakukan secara langsung di CV. Smart Motecare Mandiri dengan mengamati dan menganalisa tentang apa saja yang berkaitan dengan sistem pengolahan data dalam pembayaran pembuatan laporan. Data yang diperoleh dari hasil pengamatan yang dilakukan secara langsung akan menghasilkan data yang nyata dan akurat karena sesuai dengan apa yang dilihat dan yang diamati.

#### C. Study Pustaka (literature review)

Suatu metode untuk mendapatkan data dengan cara mempelajari jurnal-jurnal yang berkaitan dengan judul yang diambil. Sehingga penulis mendapatkan gambaran secara teoritis yang berguna untuk perancangan maupun penulisan penelitian ini. Jurnal yang diambil tentunya yang berkaitan dengan penagihan pembayaran invoice. Selain itu juga studi pustaka diperoleh dengan literature review.

#### D. Metode Analisa

Berdasarkan dari analisis yang dilakukan penulis dapat diambil kesimpulan bahwa sistem penagihan dan pembayaran invoice yang sedang berjalan saat ini di CV. Smart Motecare Mandiri belum berjalan dengan baik

karena tidak adanya pencatatan data yang sistematis sehingga informasi menjadi tidak terdokumentasi dengan baik. Dengan adanya sistem informasi pembuatan invoice ini, diharapkan dapat menyajikan aplikasi yang mudah digunakan agar dapat bermanfaat bagi staff dalam mendokumentasikan informasi, dan lebih efektif dalam melakukan monitoring pembayaran dan penagihan *invoice*.

#### E. Metode Perancangan

Metode perancangan yang digunakan dalam penelitian ini adalah metode perancangan yang berorientasi objek yang melalui tahapan pembuatan diagram UML dan prototype yang disesuaikan dengan dokumen yang telah disetujui oleh stakeholder.

#### F. Metode Testing

Salah satu metode yang digunakan dalam penelitian ini adalah metode testing blackbox dengan menggunakan metode ini peneliti dapat menemukan kesalahan fungsional aplikasi saat dioperasikan apakah input diterima dengan benar dan output telah sesuai dengan yang sudah diharapkan peneliti.

### **3.7 Alat Yang Digunakan**

Dalam perancangan ini penulis menggunakan beberapa alat untuk kelancaran penyusunan Tugas Akhir ini. Berikut adalah alat-alat yang digunakan :

#### 1. Sublime Text

Aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API.

Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatfleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages.

Sublime Text bukanlah aplikasi opensource dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan

mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.



Gambar 3.3 Logo Sublime Text.

## 2. Google chrome

Sistem operasi sumber terbuka yang dirancang oleh Google Inc untuk bekerja secara eksklusif dengan aplikasi web.

Google Chrome OS diumumkan pada tanggal 7 Juli 2009 dan versi stabilnya akan diluncurkan [umum](#) pada paruh kedua tahun 2010. Sistem operasi ini berbasis Linux dan hanya akan berjalan pada pada perangkat keras yang dirancang khusus. Antar muka penggunanya dirancang mengambil pendekatan minimalis, seperti penjelajah web Google Chrome. Sistem operasi ini ditujukan bagi mereka yang menghabiskan sebagian besar waktunya di [Internet](#).



Gambar 3.4 Logo Google Chroom.

## 3. XAMPP

Adalah seperangkat paket PHP dan MySQL yang memiliki basis open source dimana software ini dapat digunakan sebagai alat pembantu untuk mengembangkan aplikasi berbasis PHP. XAMPP ini berupa satu paket aplikasi yang berisi beberapa perangkat lunak yang berbeda-beda.



Gambar 3.5 Logo Xampp.

### 3.8 Metode Pengembangan

Penulis dalam metode pengembangannya menggunakan Metode Water Fall.

#### 3.8.1 Pengertian Water Fall

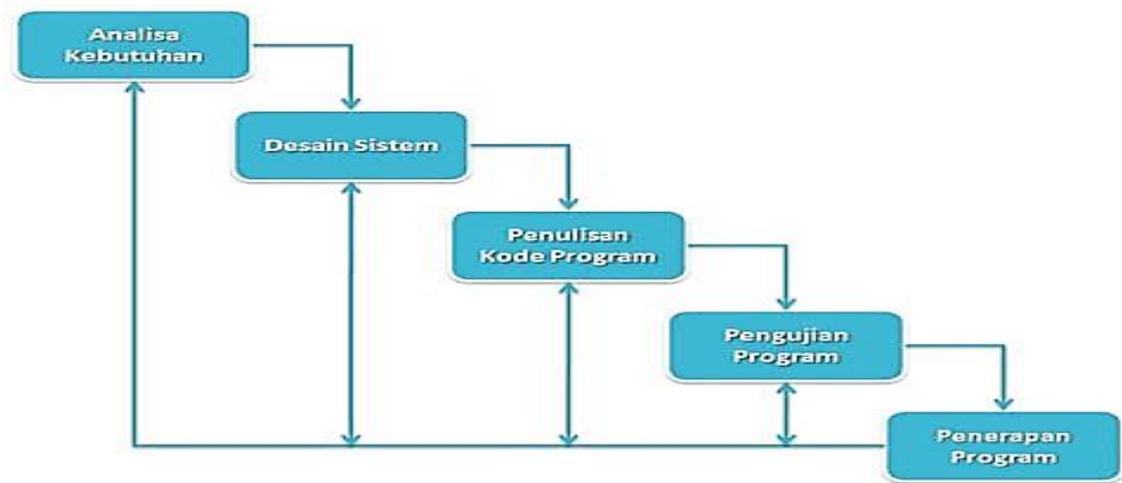
Model waterfall sering disebut dengan classic life cycle melakukan pendekatan secara sistematis dan berurutan karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

1. Sommerville (2011:29-30) mengatakan “Waterfall model adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan”.
2. Pressman (2010) mengatakan bahwa model waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun software.

Dari pengertian diatas dapat disimpulkan bahwa waterfall adalah sebuah proses perencanaan yang bersifat sistematis berurutan dalam membangun software karena pendekatan tahap demi tahap sampai selesai.

#### 3.8.2 Tahap Pemodelan Water Fall

Tahapan model waterfall menurut Sommerville dan Pressman dapat digambarkan sebagai berikut :



Gambar 3.6 Contoh Tahap Model Waterfall

Langkah-langkah pada metode waterfall adalah sebagai berikut :

A. Analisis Kebutuhan Perangkat lunak

Memahami kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh software yang akan dibangun.

B. Desain

Proses multi langka yang berfokus pada empat atribut sebuah program yang berbeda, struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) procedural. Proses desain menerjemahkan syarat/kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum mulai pemunculan kode. Proses software design untuk mengubah kebutuhan- kebutuhan diatas menjadi representasi ke dalam bentuk “blueprint” software sebelum coding dimulai.

C. Implementasi dan pengujian unit

Perancangan lunak direalisasikan sebagai serangkaian program atau unit program, Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

#### D. Integrasi dan pengujian system

Unit program atau program individual diinterasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Tujuannya untuk memastikan bahwa semua pernyataan telah diuji dan memastikan juga bahwa input yang digunakan akan menghasilkan output yang sesuai.

#### E. Pemeliharaan

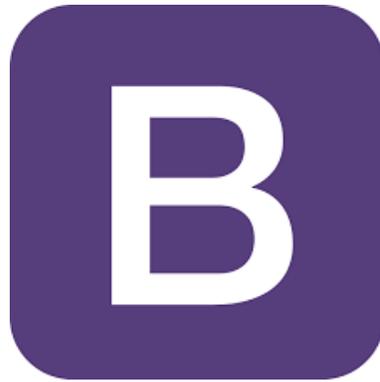
Ini merupakan fase siklus hidup yang paling lama, sistem diinstal dan dipakai. Proses ini dilakukan setelah piranti lunak telah digunakan oleh user. Perubahan akan dilakukan jika terdapat kesalahan oleh karena itu piranti lunak harus disesuaikan lagi untuk menampung perubahan kebutuhan yang diinginkan konsumen.

### 3.9 CSS Framework

CSS Framework adalah pustaka CSS yang dimana sudah dibuat dan siap untuk digunakan. Dengan CSS Framework proses design website nantinya hanya tinggal menggunakan class-class yang sudah disediakan masing-masing CSS Framework. Berikut adalah CSS Framework yang digunakan oleh penulis untuk membantu proses design sebuah website.

- **Bootstrap**

Salah satu CSS Framework yang cukup populer adalah Bootstrap salah satu CSS Framework open source untuk membuat design website dan design aplikasi web. Slogan dari framework ini adalah “Sleek, intuitive, and powerful front-end framework for faster and easier web development”, yang berarti kita dapat mendesain sebuah website dengan lebih rapi, cepat dan mudah. Bootstrap menyediakan komponen berupa layout tampilan, form, button dan typography.



Gambar 3.7 Logo Bootstrap

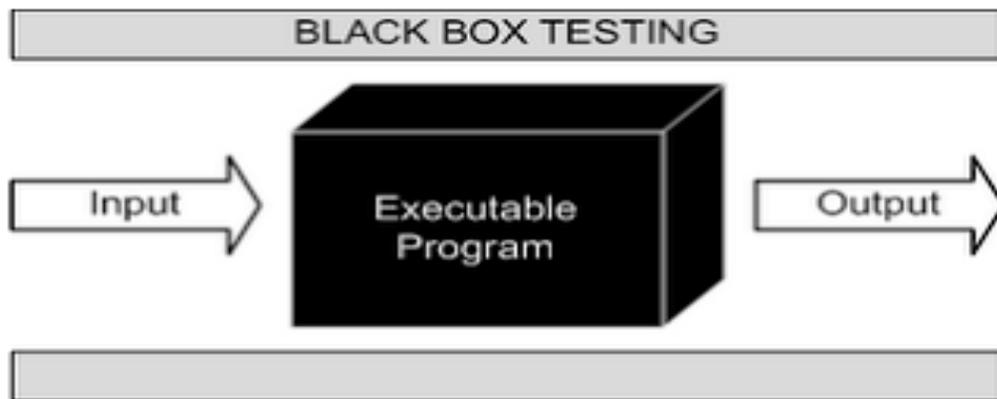
### **3.10 Konsep Dasar Black Box Testing**

#### **3.10.1 Pengertian Black Box Testing**

Pengujian menggunakan sekumpulan aktifitas validasi, dengan pendekatan black box testing.

Menurut Shalahuddin dan Rosa (2011), black box testing adalah “menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian black box testing harus dibuat dengan kasus benar dan kasus salah”.

Menurut Pressman (2010) black box testing juga disebut “pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian black box memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performansi, kesalahan inisialisasi dan akhir program”.



Gambar 3.8 Black Box Testing

### 3.10.2 *Equivalence Partitioning* Black Box testing

*Equivalence Partitioning* merupakan metode *black box testing* yang membagi domain masukan dari program kedalam kelas-kelas sehingga test case dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence Partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi *boolean*.

Kesetaraan kelas dapat didefinisikan menurut panduan berikut (Pressman, 2001) :

1. Jika masukan kondisi menentukan kisaran, satu sah dan dua diartikan tidak valid kesetaraan kelas.
2. Jika masukan membutuhkan nilai, kondisi tertentu satu sah dan dua tidak valid kesetaraan kelas diartikan.
3. Jika masukan kondisi menentukan anggota dari set, satu sah dan satu tidak valid kesetaraan kelas diartikan.
4. Jika kondisi yang input, *boolean* satu sah dan satu tidak valid kelas diartikan.

### 3.10.3 Kelebihan Black Box testing.

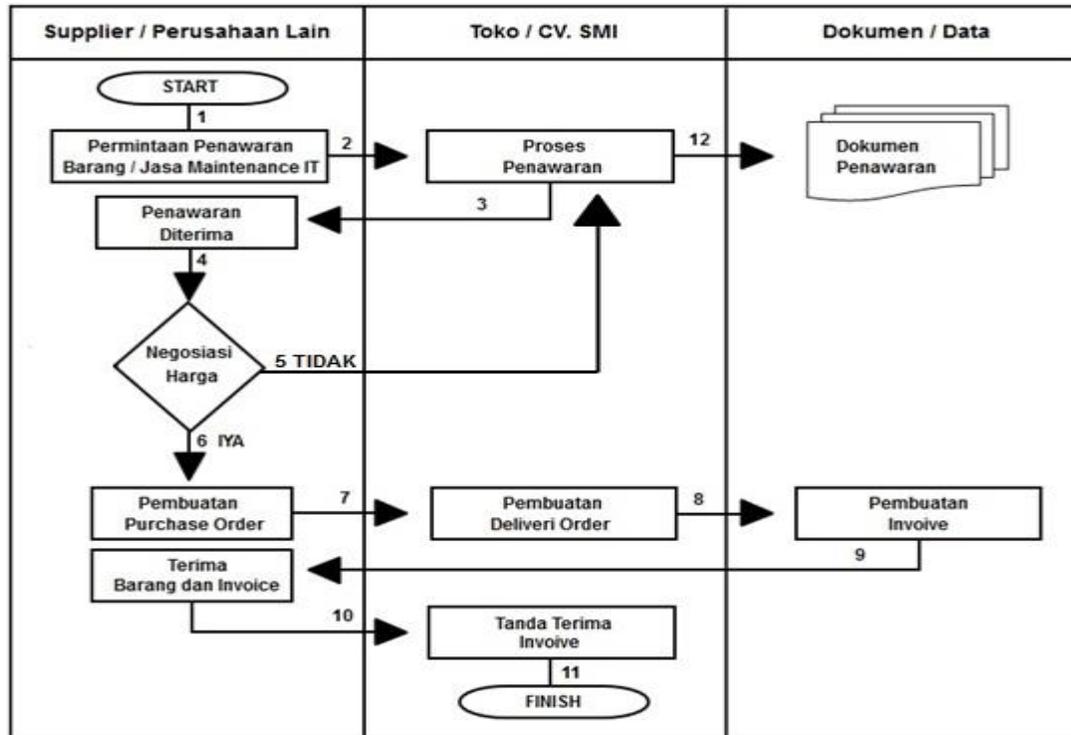
1. Tidak perlu melihat source code secara detail hanya cukup dengan menjalankan modul-modul pada aplikasi apakah ada modul yang tidak berfungsi atau tidak.
2. Mendeteksi kealahan *design* / user interface dari sebuah *Software* atau *website*.
3. Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk dianalisa dan diperbaiki.
4. Seorang tester tidak harus programmer karena pengujian hanya cukup menjalankan aplikasi tanpa harus melihat codingan secara mendalam.

### 3.10.4 Kekurangan Black Box Testing

1. Ketergantungan dengan dokumen dan desain tersebut.
2. Tidak sampai level code, sehingga tester tidak mengetahui level security dari software tersebut.

### 3.11 Sistem *Invoice* Berjalan

Flowchart Sistem *Invoice* berjalan



Gambar 3.9 Flowchart Sistem *Invoice* berjalan.

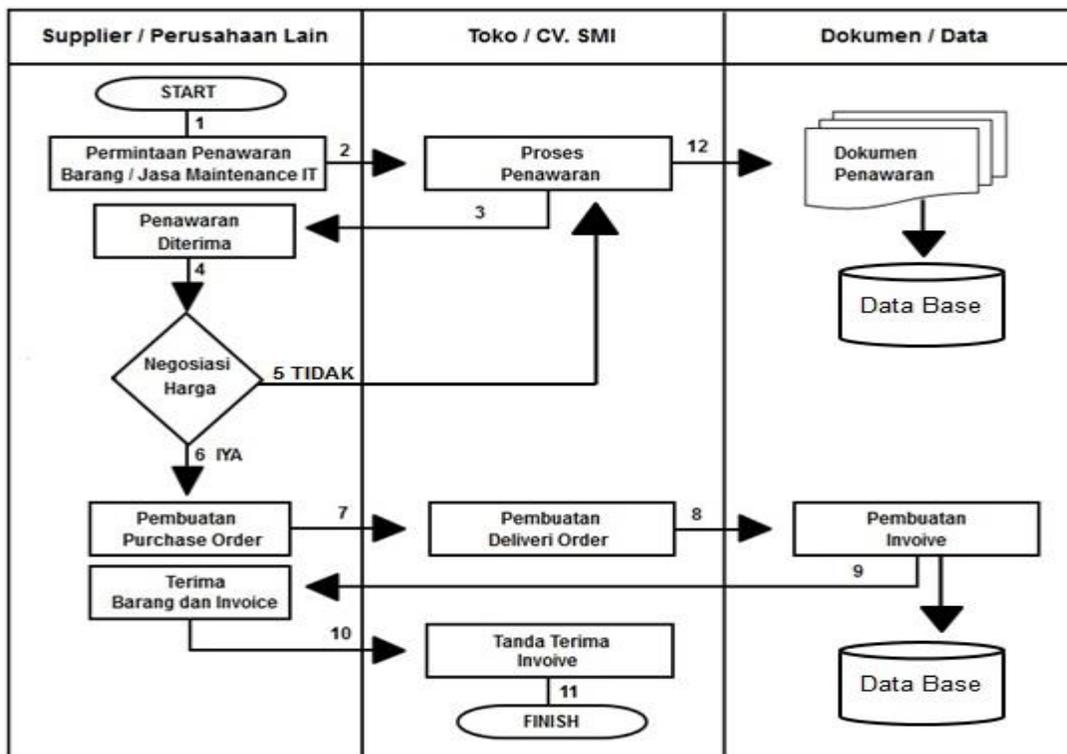
Keterangan:

1. Supplier membuat permintaan penawaran barang atau jasa *maintenance* kepada CV. SMI.
2. Lalu CV. SMI menerima dan melakukan proses penawaran tersebut.
3. Setelah penawaran diproses CV. SMI mengirimkan pengajuan harga penawaran dan Supplier menerima harga penawaran tersebut.
4. Dan Supplier melakukan bernegosiasi harga dari penawaran tersebut.
5. Jika harga TIDAK sepakat maka proses kembali lagi ke proses penawaran.
6. Dan jika harga IYA sepakat maka proses berlanjut Supplier membuat PO.
7. Setelah PO dibuat lalu di kirim kepada CV. SMI untuk pembuatan DO.
8. Setelah DO dibuat lalu pihak CV. SMI membuat Invoice.

9. Setelah DO dan Invoice dibuat lalu kirimkan ke Supplier. Dan Supplier menerima permintaan Barang serta Invoice dari CV. SMI.
10. Setelah DO Permintaan Barang serta Invoice diterima. Supplier mengirim Tanda Terima Invoice kepada CV. SmartMI.
11. CV. SMI menerima Tanda Terima Invoice tersebut.
12. CV. SMI setelah proses penawaran dibuat, lalu penawaran tersebut di simpan di Dokumen / Data.

### 3.12 Perancangan Sistem Usulan

#### 3.12.1 Flowchart Sistem yang diusulkan



Gambar 3.10 Flowchart Sistem Usulan

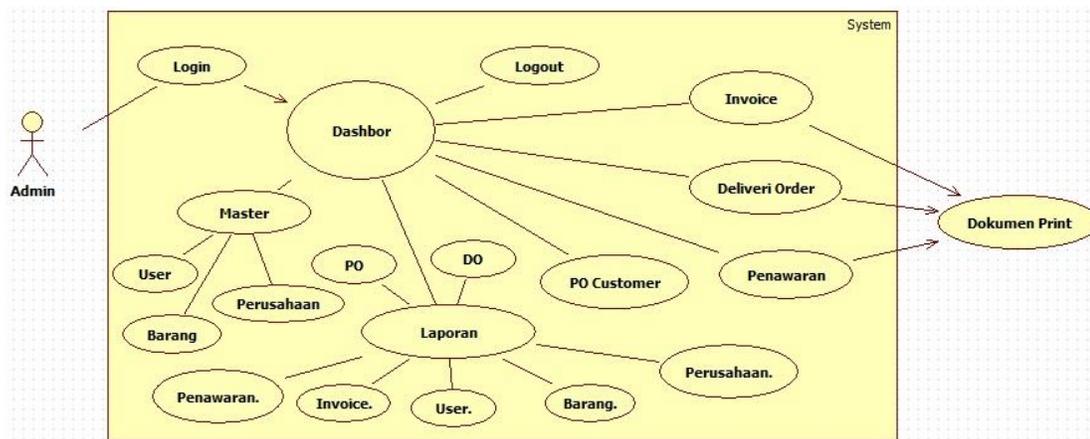
Keterangan :

1. Supplier membuat permintaan penawaran barang atau jasa *maintenance* kepada CV. SMI.
2. Lalu CV. SMI menerima dan melakukan proses penawaran tersebut.

3. Setelah penawaran diproses CV. SMI mengirimkan pengajuan harga penawaran dan Supplier menerima harga penawaran tersebut.
4. Dan Supplier melakukan bernegosiasi harga dari penawaran tersebut.
5. Jika harga TIDAK sepakat maka proses kembali lagi ke proses penawaran.
6. Dan jika harga IYA sepakat maka proses berlanjut Supplier membuat PO.
7. Setelah PO dibuat lalu di kirim kepada CV. SMI untuk pembuatan DO.
8. Setelah DO dibuat lalu pihak CV. SMI membuat Invoice dan Dokumen / Data disimpan dalam database.
9. Setelah DO dan *Invoice* dibuat lalu kirimkan ke Supplier. Dan Supplier menerima permintaan Barang serta Invoice dari CV. SMI.
10. Setelah DO Permintaan Barang serta *Invoice* diterima. Supplier mengirim Tanda Terima *Invoice* kepada CV. SmartMI.
11. CV. SMI menerima Tanda Terima Invoice tersebut.
12. CV. SMI setelah proses penawaran dibuat, lalu penawaran tersebut disimpan Dokumen / Data kedalam database.

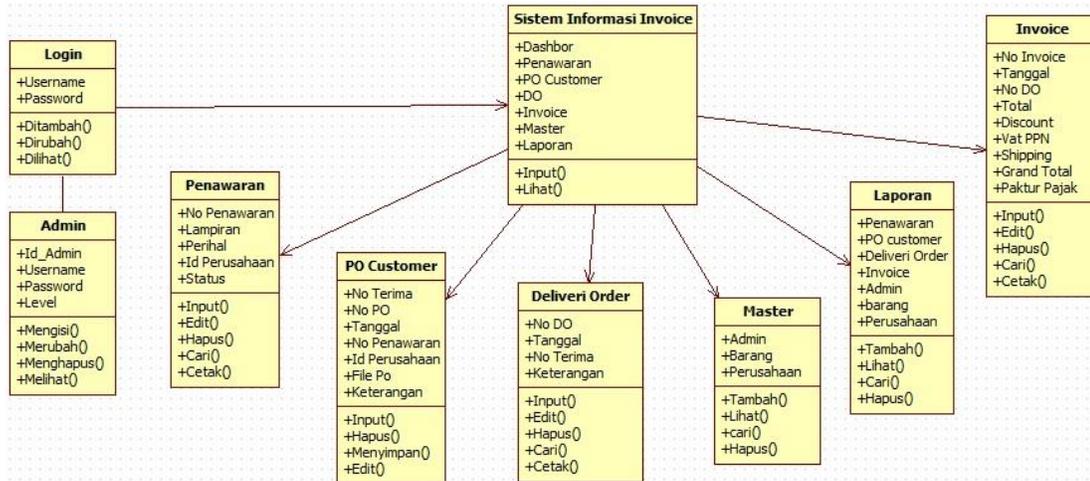
### 3.12.2 UML

#### A. Use Case Diagram Usulan



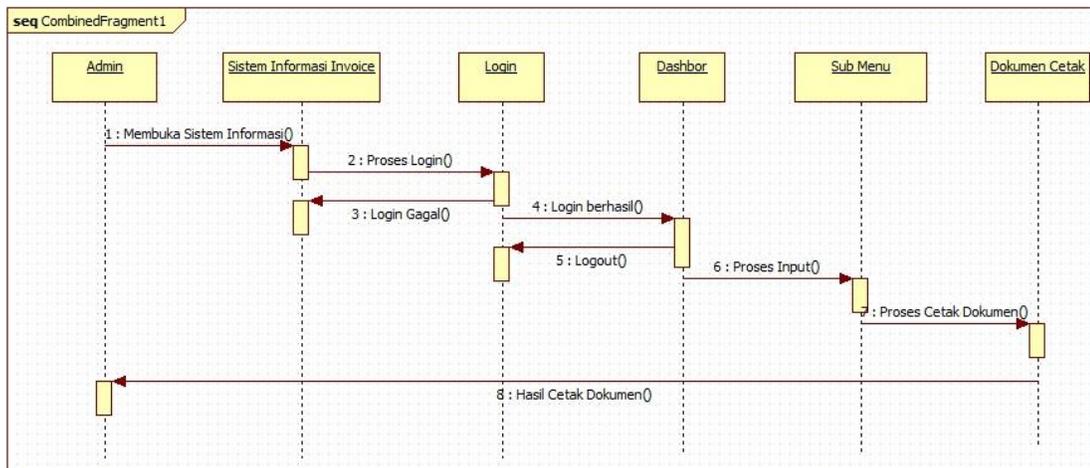
Gambar 3.11 Use Case Diagram Sitem Usulan

## B. Class Diagram



Gambar 3.12 Class Diagram Sistem Usulan

## C. Squence Diagram



Gambar 3. 13 Squence Diagram Sistem Usulan