BAB III

OBJEK DAN METODOLOGI PENELITIAN

3.1 Objek Penelitian

Penelitian ini dilakukan di PT. Pratama Salaras Mandiri (*Invisible Corporation*) Bandung yang beralamat di jalan Arcamanik Endah (Peluasan) Ruko 4 No. 17 Bandung

3.1.1 Sejarah Singkat PT. Pratama Selaras Mandiri (*Invsible Corporation*)

Invisible Corporation merupakan salah satu pelaku bisnis kreatif anak muda yang telah berdiri sejak tahun 2010, dan sampai saat ini kami sudah memiliki beberapa *brand* sepatu dan pakaian yang sudah terkenal diseluruh Indonesia yaitu GIANT FLAMES, MM SHOES dan TRUMPH FOOTWEAR. Pemasaran kreatif kami lakukan secara *online* dan *offline* yang didukung oleh sumber daya manusia rata-rata anak muda yang memiliki kreativitas tinggi. Berangkat dari latar belakang sesama anak muda kreatif kota Bandung, kami terdorong untuk menawarkan kerjasama dalam berbagai hal guna mendukung kegiatan positif anak muda yang diselenggarakan, khususnya di Sekolah atau Universitas di kota Bandung.

Invisible Corporation yang berada dibawah naungan PT. Pratama Selaras Mandiri merupakan jenis usaha yang berkecimpung di bidang *fashion* khususnya sepatu dan pakaian. Dalam naungan nama *Invisible Corporation*,

produk asli hasil kreativitas anak muda Indonesia, telah memiliki tiga *brand* yang memiliki ciri khas dan kekuatan produk masing-masing. *Invisible Corporation* memiliki segmentasi pasar 80% anak muda seluruh Indonesia, pria dan wanita.

Invisible Corporation memiliki pasar online yang sangat luas sampai mencakup seluruh Indonesia. Selain pasar online yang luas, *Invisible Corporation* juga memiliki *store* yang terletak di Bandung. *Invisible Corporation* berkapasitas produksi 5000 pasang sepatu per bulannya, dan masih akan bertambah lagi sesuai permintaan pasar, dengan bahan material yang berkualitas dan sistem produksi yang sangat terjaga kualitasnya. *Invisible Corporation* memiliki program garansi seumur hidup untuk semua produk-produknya yang akan menjamin kepuasan konsumen.

3.1.2 Visi dan Misi PT. Pratama Salaras Mandiri (*Invisble Corporation*)

Visi dan misi PT. Pratama Salaras Mandiri (*Invisble Corporation*) adalah sebagai berikut:

3.1.2.1 Visi

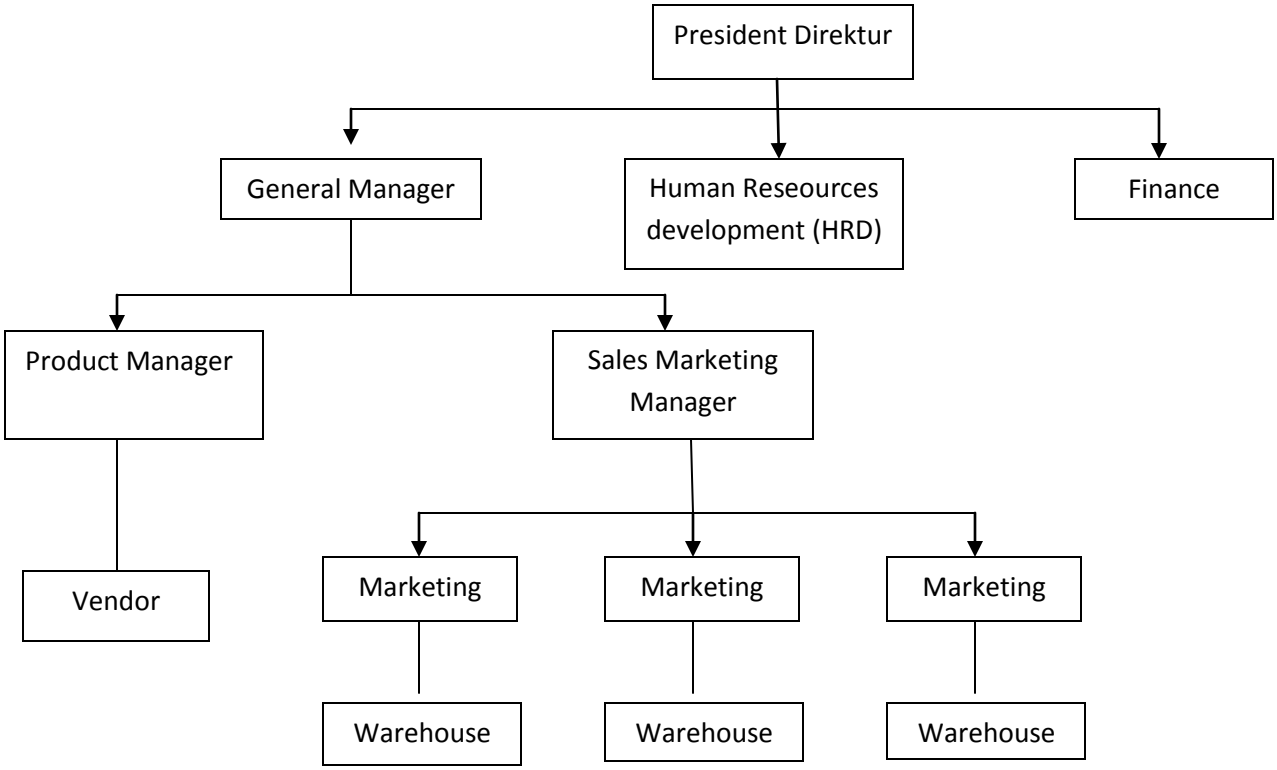
Ingin memenuhi kebutuhan alas kaki untuk kurang lebih 250 juta pasang kaki di Indonesia, baik pria dan wanita yang mempunyai produk yang tinggi, tahan lama, nyaman dipakai, dan mempunyai model yang ekslusif dan limited, sehingga dapat meningkatkan kepercayaan diri bagi para pemakainya.

3.1.2.2 Misi

1. Meningkatkan kuantitas produk mencapai 10.000 pasang perbulan serta meningkatkan kualitas produk.

2. Menumbuhkan rasa keinginan untuk mempersembahkan produk yang bermanfaat bagi semua orang.
3. Menggeser produk luar negeri di kalangan masyarakat Indonesia agar lebih tertarik untuk menggunakan produk dalam negeri .

3.1.3 Struktur Organisasi PT. Pratama Salaras Mandiri (*Invisible Corporation*)



Gambar 3.1 Struktur Organisasi PT. Pratama Salaras Mandiri (*Invisible Corporation*)

3.1.4 Job Deskription

1. **President Direktur**

   Bertanggung jawab terhadap kemajuan perusahaan dan mengkoordinasikan seluruh kegiatan perusahaan dan juga memimpin perusahaan dan mengawasi kelancaran perusahaan sesuai dengan tujuan dan kebijakan yang telah di telah ditetapkan .

2. **General Manager**

   Membatu kinerja presiden direktur khususnya yang terkait dengan kegiatan usaha yang di jalankan oleh PT. Pratama Selaras mandiri (*Invisible Corporation*) serta menjalankan bagian tugasnya secara baik.

3. **HRD (*Human Resources Development*)**

   Bertanggung jawab terhadap kegiatan yang berhubungan dengan rekapitulasi absensi karyawan,perhitungan gaji,tunjangan dan bonus .

4. **Finance**

   Melakukan pengaturan keuangan perusahaan dan membuat laporan manajemen kepada induk perusahaan.

5. **Product Manager**

   Bertanggung jawab terhadap seluruh produksi,membuat marketing plan dan membuat analisa dan evaluasi produk.

6. **Vendor**

   Bertanggung jawab dalam pembuatan sepatu dengan model-model terbaru .

7. **Sales Marketing Manager**

   Bertanggung jawab terhadap keseluruhan aspek sales,distribusi marketing,brand manajemen perusahaan,termasuk mengelola semua merk dan produk-produk perusahaan, dalam menunjang pencapain target yang telah di tentukan.

8. **Marketing**

   Bertanggung jawab mengenalkan produk perusahaan baik melalui promosi langsung,iklan,kegiatan pemasaran langsung di suatu tempat dan media lainnya.

9. **Warehouse**

   Menjaga kestabilan stock barang yang ada dan melakukan packing barang.

3.2 Metode Penelitian

Metode penelitian adalah kegiatan yang dilakukan untuk memperoleh fakta-fakta dari suatu pengetahuan dengan cara mengumpulkan, mencatat dan menganalisa data yang dikerjakan secara sistematis berdasarkan ilmu pengetahuan.

3.2.1 Desain Penelitian

Metode penelitian yang digunakan oleh penulis adalah metode penelitian deskriptif dan *Action*. Menurut Erlina (2008:20), penelitian deskriptif adalah penelitian terhadap fenomena atau populasi tertentu yang diperoleh oleh peneliti dari subjek berupa individu, organisasional, industri atau perspektif yang lain. Penelitian

deskriptif dilakukan untuk menjawab pertanyaan tentang: apa, siapa, kapan, dimana, dan bagaimana yang berkaitan dengan karakteristik populasi atau fenomena tersebut.

Metode *Action* atau tindakan merupakan penelitian dimana penulis berupaya untuk memecahkan masalah dalam dunia nyata dengan mengkaji pengalaman-pengalaman dalam memecahkan masalah tersebut.

3.2.2 Jenis dan Metode Pengumpulan Data

Data yang diperlukan dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer yang diperlukan dalam penelitian ini dikumpulkan dengan melakukan penelitian lapangan yaitu datang langsung ke perusahaan yang bersangkutan. Data sekunder dikumpulkan melalui penelitian kepustakaan.

3.2.2.1 Sumber Data Primer

Sumber data primer yaitu sumber data yang berasal dari perorangan, kelompok, panel atau sumber terselubung. Dalam memperoleh data primer penulis melakukan wawancara dan observasi di tempat penelitian.

1. **Pengamatan Langsung (Observasi)**

   Proses untuk mendapatkan data dengan mengamati objek yang akan diteliti secara langsung di tempat penelitian yaitu di PT. Pratama Selaras Mandiri (*Invsible Corporation*). Yang meliputi bagaimana proses pemesanan barang yang berjalan di PT. Pratma Selaras Mandiri (*Invisible Corporation*).

2. **Wawancara (Interview)**

   Wawancara proses untuk mendapatkan data dengan melakukan dialog langsung kepada orang yang dianggap bisa memberikan informasi yang

dibutuhkan oleh penulis yaitu kepada salah satu staff PT. Pratma Selaras Mandiri (*Invisible Corporation*).

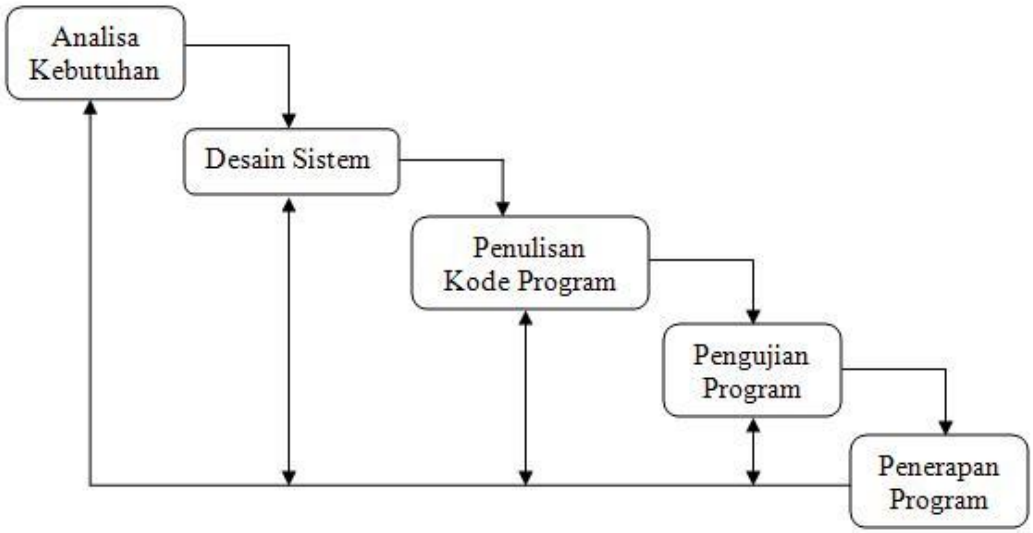
3.2.2.2 Sumber Data Sekunder

Sumber data sekunder adalah data yang digunakan untuk mendukung data primer. Jenis data primer merupakan jenis data yang sudah diolah terlebih dahulu oleh pihak pertama. Pengumpulan data yang dilakukan penulis dengan cara membaca dan mempelajari buku-buku yang berhubungan dengan masalah yang diteliti untuk melengkapi dan menunjang data-data yang telah dipelajari dalam laporan. Data sekunder yang diperoleh dari instansi terkait seperti struktur organisasi, uraian tugas dan fungsi dari sruktur organisasi *(job deskription)*.

3.3 Metode Pendekatan Sistem

Metode pendekatan yang digunakan adalah metode pendekatan yang berorientasi objek dengan menggunakan metode pemodelan UML, yang mencakup analisis dan desain yang disebut *OOAD* (*Object Oriented Analysis and Design*). *Object oriented analysis* adalah metode analisis yang memeriksa *requirements* (syarat atau keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan. Sedangkan *Object oriented analysis* adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

3.4 Metode Pengembangan Sistem



Gambar 3.2 Metode Waterfall

Sumber: http://nurichsan.blog.unsoed.ac.id/2010/11/19/metode-pengembangan-waterfall-prototyping/

Pada perkembangan sistem informasi ini penulis menggunakan metode *waterfall* sebagai alternatif pengembangan sistem informasi dengan tujuan mendapatkan kualitas dari sistem yang dihasilkan akan baik.

Metode *waterfall* merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut : Analisa, *Design, Code* dan *Testing,* Penerapan.

1. **Analisa**

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa malakukan sebuah penelitian, wawancara atau study literatur. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirment* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan aplikasi penerimaan barang. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menterjemahkan ke dalam bahasa pemprograman.

2. **Design**

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirment*. Dokumen inilah yang akan digunakan *proggrammer* untuk melakukan aktivitas pembuatan sistemnya.

3. **Coding & Testing**

Coding merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menterjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam

mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

4. **Penerapan**

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem atau rancangan. Setelah melakukan analisa, design dan pengkodean maka sistem atau rancangan yang sudah jadi akan digunakan oleh *user*.

3.5 Alat Bantu dan Pengembangan Sistem

1. Use case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang di perbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu misalnya login sistem, meng-*create* sebuah daftar belanja dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain

sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di *include* akan di panggil setiap kali *use case* yang meng-*included* eksekusi secara normal.

Sebuah *use case* dapat di *include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behavior*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. Activity Diagram

Activity Diagram atau yang di sebut Diagram Aktivitas digunakan untuk mendokumentasikan alur kerja pada sebuah *system,* yang di mulai dari pandangan business level hingga ke *operational level,* adapaun penjelasanya antara lain :

1. *Activity* diagram merupakan cara lain memodelkan aliran kejadian.
2. Kita gunakan *activity* diagram untuk menggambarkan *workflow*/aliran kerja proses bisnis.
3. *Activity* diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang di rancang bagaimana masing – masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

4. *Activity* diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

3. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan antar *class.*

4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari class. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary, controller* dan *persistent entity.*

5. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan *(dependency)* di antaranya. Komponen piranti lunak adalah modul berisi *code,* baik berisi *source code* maupun *binary code,* baik *library* maupun *executable,* baik yang muncul pada *compile time, link time,* maupun *run time.* Umumnya komponen terbentuk dari beberapa *class* dan/atau *package,* tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface,* yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

6. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi

server, dan hal-hal lain yang bersifat fisikal . Sebuah *node* adalah *server, workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

3.6 Pengujian Software

Pengujian Software (perangkat lunak) adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Metode pengujian yang digunakan oleh penulis adalah black-box testing. Black Box Testing digunakan untuk menguji fungsi-fungsi dari perangkat lunak yang dirancang. Pengujian black box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujuan black box memungkinkan perekayasa perangkat lunak mendapat serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian black box merupakan pendekatan komlementer yang kemungkinan besar mampu mengungkap kelas kesalahan. Pengujian black box berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi yang tidak benar atau hilang.
2. Kesalahan antar muka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja dan Inisialisasi kesalahan terminasi.