BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Dalam penelitian ini, yang menjadi objek penelitian adalah *CV. Racing Line The Ultimate Clothing* Bandung.

3.1.1 Sejarah Singkat Perusahaan

Racing Line The Ultimate Clothing merupakan perusahaan yang bergerak dalam bidang penjualan t-shirt, tas, sweater, dan jaket yang dikemas dalam trend remaja. *Distro* ini berdiri pada 23 Februari 2011 oleh Martin Hidayat beliau adalah Pemilik sekaligus Pimpinan Perusahaan. Berawal dari jualan-jualan kepada teman-teman, kerabat dan saudara usaha ini sekarang sudah sampai keluar pulau Jawa, diantaranya Medan, Bali, Makassar, Samarinda, Palu dan Papua.

Dengan berjalanya waktu melihat usaha distro lebih menjanjikan maka dibangunlah sebuah *distro* dengan berbagai desain – desain menarik membuat para calon pembeli tertarik untuk membeli, dan inilah yang membuat para calon pembeli tertarik untuk datang ke *distro* ini dan ingin memilikinya.

Racing Line The Ultimate Clothing menjual pakaian, tas, sweater, dan jaket untuk pria dan perempuan namun prioritas desain yang tersedia di distro ini untuk pria karena desain khusus otomotif. *Racing Line The Ultimate Clothing* berlokasi Jl. A.H. Nasution No 49 Bandung.

3.1.2 Visi dan Misi Perusahaan

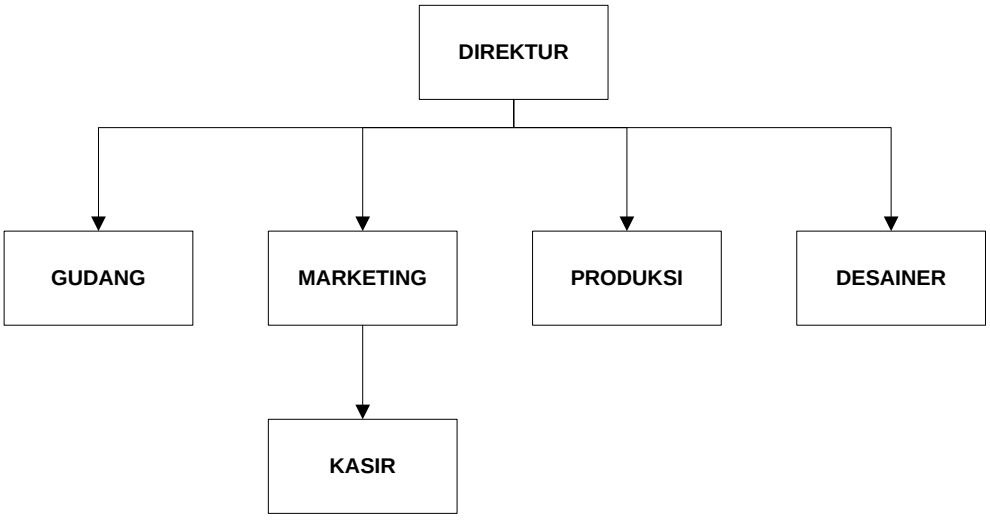
Visi

Menjadi yang terdepan dan terbaik dalam menyediakan jasa printing & clothing yang terpadu.

Misi :

a. Menyediakan layanan yang cepat, tanggap dan bersahabat bagi para pelanggan.
b. Menyediakan layanan yang terpadu di bidang printing dan clothing.
c. Menyediakan layanan yang mudah dijangkau bagi kalangan mahasiswa pada khususnya dan masyarakat pada umumnya.
d. Menjadi perusahaan yang terpercaya dalam ketepatan waktu dan kualitas barang.
e. Selalu memberikan keamanan kepada para pelanggan.

3.1.3 Sturuktur Organisasi Perusahaan



Gambar 3.1 Struktur Organisasi pada *Racing Line Clothing*

3.14 Deskripsi Tugas

Deskripsi jabatan digunakan untuk mengetahui tugas, wewenang dan tanggung jawab dari masing-masing bagian. Adapun deskripsi tugas yang ada pada *Racing Line Clothing* adalah sebagai berikut :

1. Direktur/Owner *Racing Line Clothing*

   Mengawasi jalannya sistem prosedur operasional *Racing Line Clothing* secara keseluruhan untuk menjaga konsisitensi kerja dalam mencapai tujuan perusahaan secara efektif dan efisien dan melaksanakan pemeriksaan yang meliputi seluruh aspek kegiatan manajemen keuangan operasional agar pengelolaan dapat dilaksanakan secara efektif dan efisien.

2. Gudang

   Bagian Gudang bertugas mengendalikan persediaan bahan baku dan barang, bertanggung jawab penuh terhadap pembelian bahan baku dan penjualan barang, mengendalikan stok.

3. Marketing

   Bagian Marketing bertugas untuk meningkatkan pejulaan dengan mengelala penjulalan, dan menyusun kegiatan pemasaran unuk memenuhi kegiatan dan kepuasan pelanggan.

4. Produksi

   Bagian Produksi bertugas melaksanakan atau membuat produk sesuai pola (Desain gambar) yang sudah ditentukan oleh Bagian Desainer.

5. Desainer

   Bagian Desainer bertugas membuat pola (Desain gambar) atau *prototype* dari sebuah produk yang akan diluncurkan atau diproduksi.

6. Kasir

   Bagian Kasir bertugas mencatat dan mengaudit penjualan barang agar lebih teratur dan efisien.

3.2 Metode Penelitian

Metode penelitian yang akan dilakukan dalam penelitian ini adalah metode yang menghasilkan produk juga melibatkan individu sebagai objek penelitian.

3.2.1 Desain Penelitian

Desain penelitian ini termasuk kedalam penelitian yang bersifat deskriptif. Penelitian deskriptif adalah penelitian yang bertujuan untuk memperoleh ciri-ciri variabel, dimana dalam penelitian ini adalah untuk memperoleh gambaran tentang kinerja program dirancang dan diimplementasikan kepada pengguna (*user*) pada *CV. RACING LINE THE ULTIMATE CLOTHING* BANDUNG.

3.2.2 Jenis dan Metode Pengumpulan Data

Dalam pengembangan sistem informasi tentunya diperlukan jenis data yang akurat sesuai dengan sistem yang akan dikembangkan, oleh karena itu dibutuhkan suatu teknik pengumpulan data yang tepat. Jenis data yang digunakan penulis dalam tahap analisa dan tahap perancangan adalah data primer dan data sekunder.

3.2.2.1 Sumber Data Primer

Sumber data primer, yaitu data yang diperoleh langsung dari sumbernya, diamati dan dicatat untuk pertama kalinya. Data diperoleh secara langsung dalam kegiatan yang berlangsung pada *CV. Racing Line The Ultimate Clothing* Bandung. Adapun metode penelitian yang dilakukan oleh penulis untuk pengumpulan data dan informasi berupa :

1. *Observasi*

Metode ini merupakan teknik pengumpulan data yang dilakukan dengan mengamati secara langsung dan ikut serta kedalam kegiatan yang berlangsung pada *Racing Line The Ultimate Clothing* Bandung khususnya bagian gudang, marketing dan designer guna memperoleh gambaran yang berhubungan dengan sistem informasi pembelian dan penjualan**.**

2. Wawancara (*Interview*)

Dalam hal ini penulis melakukan tanya jawab secara langsung mengenai permasalahan yang akan diteliti kepada pihak yang bersangkutan di bagian gudang, design dan marketing yaitu mengenai hal-hal yang menyakut pembelian bahan baku, pengajuan bahan baku untuk produksi dan penjualan, serta persediaan atau stok baik itu bahan baku atau barang. Wawancara ini bertujuan untuk memperjelas dan meyakinkan atas fakta atau informasi yang diperoleh melalui observasi.

3.2.2.2 Sumber Data Skunder

Sumber data sekunder, yaitu data yang diperoleh dari sumber lain seperti catatan perusahaan, arsip-arsip, dokumen-dokumen, file-file yang ada. Penulis mengumpulkan dokumen-dokumen yang dibutuhkan dalam proses analisis dan desain pengembangan sistem yang akan dibuat. Studi Dokumentasi, melakukan pengumpulan data yang berkaitan dengan masalah pembelian bahan baku dan penjualan barang dari dokumen yang diberikan oleh perusahaan berupa dokumentasi form *purchase order (*PO), pengajuan bahan baku, faktur penjualan, faktur pembelian bahan baku, dan laporan – laporan .

3.2.3 Metode Pendekatan dan Pengembangan Sistem

Adapun metode pendekatan dan pengembangan sistem akan dijelaskan seperti dibawah ini antara lain yaitu:

3.2.3.1 Metode Pendekatan Sistem

Metode pendekatan yang digunakan adalah metode pendekatan berorientasi objek (*Object Oriented).* Menurut Adi Nugroho (2005:4) Pendekatan berorientasi objek merupakan

cara berfikir baru serta berlogika dalam menghadapi masalah-masalah yang akan di atasi dengan bantuan komputer. OOP mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu. Pendekatan berorientasi objek terdiri dari analisis berorientasi objek *(OOA)* dan desain berorientasi objek *(OOD)*.

Analisis berorirentasi objek *(OOA)* dimulai dengan menyatakan suatu masalah, analisis membuat suatu model situasi dari dunia nyata, menggambarkan sifat yang penting. Sedangkan Desain berorientasi objek (*OOD)* merupakan tahap lanjutan setelah *(OOA)*, dimana tujuan sistem diorganisasikan kedalam subsistem berdasarkan struktur analisis dan arsitektur yang dibutuhkan.

3.2.3.2 Metode Pengembangan Sistem

Adapun metode pengembangan yang digunakan adalah *prototype*, dalam buku Kendal (2003:221) *prototyping* sistem informasi yaitu suatu teknik yang sangat berguna untuk mengumpulkan informasi tertentu mengenai syarat-syarat informasi pengguna secara cepat. *Prototyping* yang efektif seharusnya dilakukan pada awal-awal siklus pengembangan sistem, yakni selama fase penetapan syarat-syarat. *Prototyping* adalah suatu teknik yang sangat rumit karena memerlukan pengetahuan mengenai siklus hidup pengembangan sistem secara keseluruhan.

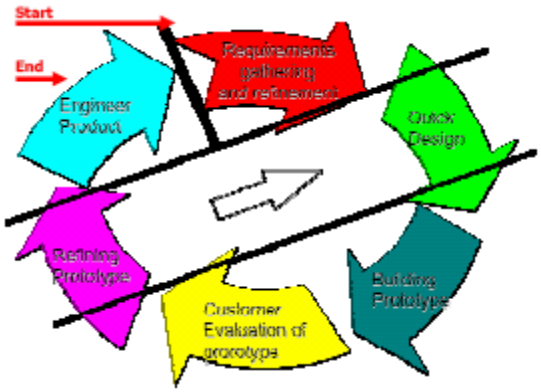
Sistem *prototype* merupakan bagian operasional dari sistem secara keseluruhan yang akan dibangun. *Prototype* merupakan suatu cara yang baik untuk mendapatkan umpan balik mengenai sistem yang diajukan dan mengenai bagaimana sistem tersebut tersedia untuk memenuhi kebutuhan informasi pengguna.

Ada empat petunjuk yang harus diamati saat mengintegrasikan *prototype* kedalam fase penetapan siklus hidup pengembangan sistem, yakni:

1. Bekerja sesuai modul
2. Membangun *prototype* dengan cepat

3. Memodifikasi *prototype* dengan iterasi yang berurutan

4. Menekankan antar muka pengguna.



Gambar 3.2 Model Prototype

Sumber : blog.unsri.ac.id/userfiles/09071003013.doc

Tahapan-tahapan dalam *Prototype* adalah sebagai berikut:

1. Pengumpulan kebutuhan

   Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun *prototyping*

   Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).

3. Evaluasi *protoptyping*

   Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginann pelanggan. Jika sudah sesuai maka

langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulangu langkah 1, 2 , dan 3.

4. Mengkodekan sistem

   Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji sistem

   Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan pengujian Black Box.

6. Evaluasi Sistem

   Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan . Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

7. Menggunakan sistem

   Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

3.2.3.3 Alat Bantu Analisis dan Perancangan

Alat bantu yang digunakan analisis dan perancangan sisitem yaitu menggunakan *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu.

UML mendeskripsikan OOP (*Object Oriented Programming*) dengan beberapa diagram, diantaranya:

1. *Use case diagram*

   Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

2. *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 satu sama lain seperti *containment*, pewarisan, *asosiasi*, dan lain-lain.

3. *Activity diagram*

   Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

   Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak

menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

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 atar 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.

5. *Collaboration diagram*

   Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari *level* tertinggi memiliki nomor 1. *Messages* dari *level* yang sama memiliki *prefiks* yang sama.

6. *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.

3.2.4 Pengujian Software

Pengujian *Software* adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan.

Pengujian *Black-box* berfokus pada struktur tampilan kontrol program. Test case dilakukan untuk memastikan bahwa semua statement pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji.

Pengujian *Black-box* berfokus kepada persyaratan fungsional perangkat lunak. Pengujian *Black-box* memungkinkan perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *Black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang

kemungkinan besar mampu mengungkap kesalahan-kesalahan pada metode *white-box*.