BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Dalam menentukan objek penelitian, penulis melakukan penelitian pada Distro Blackjack yang beralamat di Jl.Trunojoyo No 34 Bandung.

3.1.1 Sejarah Singkat Perusahaan

Pada era awal di mulai berdirinya distro Black Jack di Bandung,si pemilik (Ronald) bekerja ditempat saudara (Paman-nya), sejak itu pemilik ter-inspirasi untuk membuat usaha distro sendiri. Nama Distro Black Jack ini diambil dari tanggal lahir pemilik balck jack, yang mana dalam kartu *black jack* berjumlah 21(dua puluh satu), sesuai dengan tanggal lahir pemilik.Awal tujuan dari didirikan distro ini tak lain adalah sebagai usaha memajukan usaha clothing di bandung, dan sebagai tempat yang menarik untuk belanja ataupun berkunjung untuk melihat produk yang baru di Distro BlackJack Distro ini didirikan pada tahun 2004 yang beralamatkan di Jl. Sulanjana Bandung. kemudian pada tahun 2006 distro pindah ke Jl Trunojoyo 34 Bandung sampai sekarang.

3.1.2 Visi dan Misi Perusahaan

Visi distro adalah Memperluas dan mempertahankan usaha yang sedang di jalankan, serta mencari keuntungan sebesar mungkin dengan modal seminim mungkin. Sedangkan Misi distro adalah mepertahankan produk asli dan membuat produk sesuai dengan perkembangan jaman.

3.1.3 Struktur Organinsasi Perusahaan

Struktur organisasi merupakan salah satu komponen penting dalam pembentukan sebuah perusahaan, agar dapat mampu bersaing dan mencapai tujuan perusahaan yang diinginkan. Melalui sistem organisasi tersebut akan diatur tugas dan wewenang kepada masing-masing personil perusahaan. Tanpa adanya struktur organisasi yang baik, tidak mungkin perusahaan akan bekerja baik sesuai dengan apa yang diinginkan.

Ini berarti bahwa struktur organisasi akan sangat berperan penting terhadap kelangsungan hidup perusahaan. Struktur organisasi Distro Blackjack dapat digambarkan sebagai berikut:



Gambar 3.1 Struktur Organisasi

3.1.4 Deskripsi Tugas

1. Pemilik

   Adalah pemlilk yang mempunyai hak untuk membuat keputusan di dalam

2. Kasir

   Mempunyai tugas dan melayani konsumen di bidang transaksi keuangan dalam penjualan barang.

3. Gudang

   Mempunyai tugas di gudang untuk mengecek barang dan mengatur segalamacam aktifitas gudang.

4. Bagian Produksi

   Mempunyai tugas untuk memproduksi barang, apabila stok barang di distro sudah mencapai pada stok minimum atau habis.

3.2 Metode Penelitian

Suatu penelitian tidak akan berjalan dengan baik apabila tidak dilakukan dengan metode-metode yang baik untuk mendukung penelitian. Metode yang dilakukan penulis yaitu metode *Action.* Metode ini merancang dan mendesain program berdasakan analisis sistem yang diusulkan sesuai dengan masalah yang ada di dunia lapangan.

3.2.1 Jenis dan Metode Pengumpulan Data

Untuk memperoleh data yang diperlukan, penulis menggunakan metode sebagai berikut :

3.2.1.1 Sumber Data Primer

Data primer yaitu data yang diperoleh langsung dari sumber tempat yang diteliti yaitu Distro Blackjack. Untuk mendapatkan data primer, dilakukan hal sebagai berikut :

a. Observasi, yaitu teknik pengumpulan data dan informasi yang dilakukan dengan cara pengamatan langsung terhadap tempat penelitian. Sehingga penulis dapat mengetahui alur sistem yang dilakukan pada Distro Blackjack.
b. Wawancara, yaitu teknik pengumpulan data dengan cara tanya jawab langsung dengan para pegawai dan pemilik Distro Blackjack. Hal ini bertujuan untuk mendapatkan data serta informasi yang benar dan akurat dengan masalah yang diteliti penulis. Sehingga penulis dapat mengetahui sistem yang sedang berjalan di Distro Blackjack.

3.2.1.2 Sumber Data Sekunder

Cara yang digunakan untuk mengumpulkan data sekunder adalah metode Dokumentasi, yaitu dengan mengumpulkan data dan informasi yang diperlukan dari sumber-sumber kebanyakan dari materi sejenis dokumen yang berkenaan dengan masalah yang diteliti. Faktur penjualan merupakan dokumen yang digunakan pada

sistem penjualan di Distro Blackjack saat ini. Metode ini digunakan untuk pengumpulan data yang berhubungan dengan sejarah, tujuan, kegiatan dan struktur organisasi.

3.2.2 Metode Pendekatan dan Pengembangan Sistem

Adapun metode pendekatan dan pengembangan sistem yang digunakan oleh penulis adalah sebagai berikut :

3.2.2.1 Metode Pendekatan Sistem

Metode yang digunakan adalah dengan metode pendekatan terstruktur. Cara pendekatan terstruktur ini adalah dengan melihat sistem dimulai dari masalah utama yang ada secara *global*, dan setelah didapatkan proses utama, maka setiap proses yang ditemukan tesebut dipecah menjadi beberapa proses atau masalah secara detail dan jelas, dan begitu seterusnya hingga tahapan proses yang terakhir.

3.2.2.2 Metode Pengembangan Sistem

Metode pengembangan sistem menggunakan metode *prototype* yang merupakan metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat suatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai. Kadang-kadang pemakai atau *user* hanya memberikan beberapa kebutuhan umum *software* tanpa detail *input*, proses atau detail *output*.

Di lain waktu mungkin dimana tim pembangun (*developer*) tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form *user interface*. Ketika situasi seperti ini terjadi model *prototype* sangat membantu proses pembangunan *software*. Seperti pada semua metode, *prototype* dimulai dari pengumpulan kebutuhan.

Dengan perencanaan yang cepat akan dibentuk konstruksi dari *prototypenya*. *Prototype* ini dievaluasi oleh pelanggan dan digunakan untuk mengelola kembali kebutuhan dari perangkat lunak yang dikembangkan. Suatu proses iterasi terjadi, setelah *prototype* disesuaikan dengan kebutuhan pelanggan, sementara pihak pengembang makin mengerti keinginan pemakai.

Tahapan proses model *prototype*, bisa dijelaskan sebagai berikut:

1. Pengumpulan kebutuhan : *developer* dan pelanggan bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.
2. Perancangan : perancangan dilakukan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
3. Evaluasi *prototype* : pelanggan mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan software. Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype*

dibuat untuk memuaskan kebutuhan pelanggan dan untuk memahami kebutuhan pelanggan lebih baik.

3.2.2.3 Alat Bantu Analisis dan Perancangan Berorientasi Objek

Analisis berorientasi objek (OOA) adalah metode analisis untuk memeriksa kebutuhan-kebutuhan dari perspektif kelas-kelas dan objek-objek yang ditemukan di *domain* masalah. Selama analisis dan tahap perancangan, pengembangan mempunyai dua tugas, yaitu :

1. Identifikasi kelas-kelas dan objek-objek pembentuk pada *domain* masalah, (disebut *abstraksi* masalah).
2. Menemukan struktur dimana himpunan-himpunan objek yang bekerja sama menyediakan perilaku-perilaku yang memenuhi kebuthuan-kebutuhan masalah (disebut mekanisme *implementasi*).

Dua bagian penting dari pengertian perancangan berorientasi objek adalah:

1. Menuntut menuju dekomposisi berorientasi objek
2. Menggunakan notasi-notasi berbeda untuk mengekspresikan model-model perancangan logik berbeda (struktur kelas dan objek) dan perancangan fisik (arsitektur modul dan proses), serta juga model statis dan dinamis sistem yang sedang dirancang.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi pendukung OOP *(Object Oriented Programming*) mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya dan pendekatan OOP lebih mudah dikembangkan dan dirawat. Dalam UML (*Unified Modelling Language*) yang dapat dikelompokan berdasarkan sifatnya yaitu statis atau dinamis jenis diagramnya adalah sebagai berikut :

1. Diagram Kelas (*Class Diagram*) bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi- kolaborasi serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan berorientsi objek.
2. Diagram Objek (*Object Diagram*) bersifat statis. Diagram ini memperlihatkan objek-objek serta relasi-relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai di dalam diagram kelas.
3. Diagram Use-Case (*Use-Case Diagram*) bersifat statis. Diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram Sekuen (*Sequence Diagram*) bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu.
5. Diagram Kolaborasi (*Collaboration Diagram*) bersifat dinamis. Diagram berkolaborasi adalah menggambarkan interaksi antar objek seperti sequence

diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian pesan (*message*).

6. Diagram Statechart (*Statechart diagram*) bersifat dinamis. Diagram state ini memperlihatkan state–state pada sistem, memuat sstate, transisi, event, serta aktifitas. Diagram ini penting untuk memperlihatkan sifat dinamis dari antar muka , kelas, kolaborasi, dan terutama penting pada pemodelan sistem–sistem yang reaktif.
7. Diagram aktivitas, (*Activity Diagram*) bersifat dinamis. Diagram ini adalah diagram tipe khusus dari digram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas yan lainnya dalam suatu sistem. Diagram ini penting dalam pemodelan fungsi-fungsi dalam satu sistem dan memberi tekanan pada aliran kendali antar objek.
8. Diagram Komponen (*Componen Diagram*) bersifat statis. Diagram ini memperlihatkan organisasi serta ketergantungan system atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tifikal dipetakan kedalam satu atau lebih kelas-kelas.
9. Diagram Penyebaran (*Deployment Diagram*) bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. Deployment diagram berhubungan dengan diagram komponen dimana deployment diagram memuat satu atau lebih komponen-komponen.

3.2.2.4 Pengujian Software

Dalam pembuatan suatu program baik berupa sistem informasi atau bukan, membutuhkan suatu teknik pengujian yang berguna dalam mengevaluasi program yang dibuat. Menurut Janner Simarmata (2010:300) Adapun teknik pengujian program dibagi menjadi dua, yaitu teknik pengujian perangkat lunak *white box* dan teknik pengujian black box.

a. Kotak Putih (*White Box*)

Kadang disebut pula pengujian *glass box*, adalah metoda *desain test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case* atau dengan kata lain bahwa pengujian dilakukan untuk memastikan bahwa operasi internal bekerja sesuai dengan spesifikasi dan semua komponen internal telah diamati dengan baik. Dengan menggunakan metoda pengujian ini perekayasa sistem dapat melakukan *test case* yaitu :

1. Memberi jaminan bahwa semua jalur independent pada suatu modul telah digunakan paling sedikit satu sekali.
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*
3. Mengeksekusi semua loop sesuai dengan batasan
4. Menggunakan struktur data internal untuk menjamin validasi

b. Kotak Hitam (*Black Box*)

Konsep Kotak Hitam (*Black Box*) digunakan merepresentasikan sistem cara kerja di dalamnya tidak tersedia untuk diinspeksi. Di dalam Kotak Hitam (*Black Box)*, item-item yang diuji dianggap ”gelap” karena logiknya tidak diketahui, yang diketahui hanya apa yang masuk dan apa yang keluar dari kotak hitam.

Pada pengujian Kotak Hitam (*Black Box*), kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sendiri mungkin di proses pengembangan perangkat lunak. Teknik pengujian konvensional yang termasuk pengujian Kotak Hitam (*Black Box*) adalah sebagai berikut :

1. *Graph based testing* (pengujian yang didasarkan pada grafik)
2. *Equivalence partitioning* (menyekat persamaan)
3. *Comparison testing* (pengujian perbandingan)
4. *Orthogonal testing* (pengujian orthogonal)