BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitiaan

Objek penelitian merupakan lokasi dimana penelitian berlangsung yaitu di Toko DC (*Dream Computer)*. Adapun sejarah singkat perusahaan, visi, misi, tujuan, struktur organisasi dan deskripsi tugas adalah sebagai berikut:

3.1.1 Sejarah Perusahaan

Pada awalnya pemilik toko DC (*Dream Computer*) ini adalah seorang pegawai di salah satu perusahan tekstil. Pada saat beliau bekerja waktu itu ada beberapa komputer yang mengalami kerusakan dan harus cepat di perbaiki, untuk menunggu teknisi dari perusahaan tersebut memerlukan waktu, sedangkan kerusakan komputer tersebut harus cepat ditanggulangi. Berhubung beliau mengerti bagaimana memperbaiki komputer. kemudian beliau mengambil inisiatif untuk memperbaiki komputer tersebut. Dari kejadian tersebut kemudian beliau berfikir untuk merintis jasa *service* komputer sendiri di sela-sela waktu luangnya, dan ternyata cukup menjanjikan. Dari situ kemudian beliau bermimpi untuk membuat toko sendiri, sedangkan untuk membuat toko dibutuhkan modal yang tidak sedikit serta keseriusan dalam mengelolanya. Dan waktu itu beliau kemudian keluar dari tempat kerja sebelumnya dengan sedikit modal sisa dari hasil pekerjaan sebelumnya. Karena beliau merasa yakin kalau usaha dibidang komputer prospek kedepannya akan bagus. DC (*Dream Computer*) selain

menerima jasa *service* juga menerima jasa *sales, upgrate, dan networking*. Karena perkembangan teknologi semakin lama semakin berkembang maka DC (*Dream Computer*) tidak fokus hanya di bidang jasa saja. sekarang DC (*Dream Computer*) merambah ke penjualan notebook karena banyak permintaan dari konsumennya. Pengetahuan komputer yang dimiliki awalnya belajar secara otodidak karena beliau hobi dalam mengotak-ngatik komputer dan memiliki keyakinan serta keteguhan hati, sehingga beliau saat ini sudah memiliki toko yang diimpikannya. DC (*Dream Computer*) mulai beroperasi dari pukul 09.00 – 18.00.

DC (*Dream Computer*) adalah sebuah toko yang menjual notebook baru da bekas serta jasa. Seperti : *sales, service, upgrate,* dan *networking*. Yang berlokasi di ITC Kebon Kelapa Lt.3 Blok F01 No.5 Bandung. Awal berdirinya DC (*Dream Computer*) itu sendiri pada tahun 2005.

3.1.2 Visi dan Misi Perusahaan

Visi DC (*Dream Computer*) :

"Membangun kepercayaan yang solid antara toko DC (*Dream Computer*) dengan konsumen"

Misi DC (*Dream Computer*):

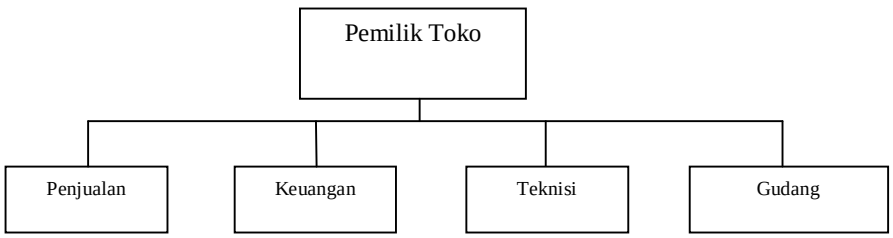
- Mengelola toko DC (*Dream Computer*) secara profesional, untuk dapat memberikan layanan yang terbaik bagi konsumen.
- Memberikan produk yang bermutu tinggi, dengan tetap memberikan harga yang bersaing kepada konsumen.

- Memberikan jasa konsultasi bagaimana memilih notebook yang baik sesuai dengan kebutuhan konsumennya.

3.1.3 Struktur Organisasi Perusahaan

Struktur organisasi merupakan salah satu komponen dasar, pembentukannya sesuai dengan kebutuhan perusahaan dan diatur sedemikian rupa sehingga mampu mendukung upaya untuk mencapai tujuan perusahaan yang diinginkan. Melalui sistem organisasi tersebut akan diatur mengenai tugas dan kewenangan masing-masing personil perusahaan serta hubungan serta pola kerja antar personil. Kejelasan mengenai hal tersebut akan sangat besar artinya guna menghindari adanya kesalahan penanganan maupun kesalahan dalam menangani suatu pekerjaan. Tanpa adanya struktur organisasi yang baik, tidak mungkin suatu perusahaan akan berjalan dengan baik sesuai yang diharapkan.

Ini berarti bahwa sedikit banyak, sistem organisasi akan berpengaruh terhadap faktor dan kemampuan manajerial. Struktur organisasi DC (*Dream Computer*) dapat digambarkan sebagai berikut:



Gambar 3.1 Struktur Organisasi Toko DC (*Dream Computer*)

3.1.4 Deskripsi Tugas

1. Pemilik Toko

   Pemilik Toko merupakan pengurus perusahaan yang bertanggung jawab penuh dalam segala hal mengenai perusahaan, yang mewakili perusahaan terhadap pihak luar, yang berhak menandatangani atas nama perusahaan, menghubungkan perusahaan dengan pihak luar dan sebaliknya serta melakukan segala tindakan baik mengenai pengurusan maupun mengenai pemilikan.

2. Penjualan

   Bagian penjualan bertanggung jawab untuk melayani komsumen yang datang ke toko DC (*Dream Computer*) dan memenuhi target yang telah ditetapkan.

3. Keuangan

   Bertugas untuk mengatur keuangan di toko DC (*Dream Computer*). Baik itu pendapatan dari hasil penjualan, dan juga pengeluaran DC (*Dream Computer*), baik itu tentang sewa gedung, listrik, kebersihan dan juga perihal tentang beban dan biaya operasional DC (*Dream Computer*).

4. Teknisi

   Bertugas melakukan perakitan komputer dari awal sampai akhir, melakukan pengecekan dan review barang. Melakukan service

terhadap hardware yang rusak secara langsung, baik penggantian komponen sebagian ataupun keseluruhan.

5. Gudang

   Bertugas melakukan pengecekan barang masuk dan keluar, melakukan pemeliharaan selama barang ada di gudang.

3.2 Metode Penelitian

Suatu penelitian tidak akan berjalan dengan baik apabila tidak dilakukan dalam suatu proses yang teratur dan terarah. Oleh karena itu, diperlukan suatu metode pendekatan/penyelesaian untuk melaksanakan suatu penelitian. Metode yang digunakan oleh penulis pada penelitian ini adalah metode *Action*, yaitu metode dengan merancang dan mendesain program berdasarkan analisis sistem yang diusulkan sesuai dengan masalah yang ada di dunia aktual (lapangan).

3.2.1 Jenis dan Metode Pengumpulan Data

Untuk memperoleh data dan keterangan yang diperlukan, dalam hal ini penulis menggunakan cara sebagai berikut :

3.2.1.1 Sumber Data Primer

Data primer yaitu data yang diperoleh langsung dari sumber yang akan diteliti, dalam hal ini adalah karyawan DC (*Dream Computer*). Untuk memperoleh data primer dapat dilakukan hal sebagai berikut :

Studi Lapangan yaitu penelitian dengan cara :

a. Observasi, yaitu teknik pengumpulan data dan informasi dengan cara pengamatan langsung terhadap operasi perusahaan, tetapi peneliti tidak ikut langsung terhadap kegiatan perusahaan.
b. Wawancara, yaitu teknik pengumpulan data dengan cara tanya jawab secara langsung dengan para pegawai yang berhubungan dengan penelitian ini. Wawancara dilakukan dengan pimpinan yang dapat memberikan data yang relevan dengan masalah yang diteliti oleh penulis.

3.2.1.2 Sumber Data Sekunder

Data sekunder merupakan cara pengumpulan data dengan cara mempelajari data yang telah tersedia atau dikumpulkan terlebih dahulu oleh pihak lain seperti buku-buku.

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. Metode ini digunakan untuk pengumpulan data yang berhubungan dengan sejarah, tujuan, kegiatan dan struktur organisasi.

Menurut Abdul Kadir (2003 : 414) pada tahapan implementasi, dokumentasi yang dibuat dapat dibagi menjadi tiga jenis, yaitu :

a. Dokumentasi pengembangan, yaitu menjabarkan sistem secara lengkap, mencakup deskripsi sistem, bentuk keluaran, bentuk masukan, bentuk basis data, bagan alir program, hasil pengujian dan lembar penerimaan

pemakai.

b. Dokumentasi operasi, yaitu mencakup jadwal pengoperasian, cara pengoperasian peralatan, faktor-faktor keamanan dan masa berlakunya suatu berkas.
c. Dokumentasi pemakai, yaitu berisi petunjuk untuk menggunakan masing-masing program.

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 pendekatan yang digunakan oleh penulis adalah dengan menggunakan pendekatan sistem berorientasi objek. Menurut Bambang Hariyanto (2004 : 86) keunggulan pendekatan berorientasi objek adalah sebagai berikut :

1. Bekerja yang mendekati kognisi manusia.
2. Menghasilkan sistem yang dibangun di atas bentuk-bentuk antara yang stabil dan dengan demikian lebih mampu untuk mengikuti perubahan.
3. Dapat digunakan tidak hanya pada perancangan perangkat lunak tapi juga seluruh proses pengembangan perangkat lunak.
4. Mereduksi resiko pengembangan sistem-sistem kompleks terutama karena pemaduan terjadi menyebar pada siklus kehidupan tidak terjadi dalam sekejap sebagaimana kejadian ledakan besar (*big bang*).

5. Pendekatan ini membantu mengeksploitasi keampuhan bahasa pemrograman berbasis objek.

Pendekatan berorientasi objek adalah salah satu faktor untuk menuntun menuju ke solusi perangkat lunak, untuk itu kadang harus menyeimbangkan dengan pertimbangan- pertimbangan pendekatan yang lain. Menurut Booch, prinsip-prinsip yang mendasari pengembangan berorientasi objek adalah :

1. Abstraksi
2. Pengkapsulan
3. Modularitas
4. Hirarki
5. *Typing*
6. Kongkurensi
7. *Persistence*

Prinsip-prinsip ini bekerja/diterapkan secara sinergi (bersama-sama bekerja sama). Prinsip-prinsip tersebut merupakan prinsip-prinsip utama, dimana keempat prinsip pertama adalah wajib sementara tiga prinsip berikutnya adalah *optional* (boleh digunakan ataupun tidak).

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 *prototype*nya. *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.

Seluruh metode pengembangan sistem memiliki kelebihan dan kekurangan. Berikut adalah kelebihan dan kekurangan dari metode *prototype*. Abdul Kadir (2003 : 418) :

1. Kelebihan
    a. Pendefinisian kebutuhan pemakai menjadi lebih baik karena keterlibatan pemakai yang lebih intensif.
    b. Meningkatkan kepuasan pemakai dan mengurangi resiko pemakai tidak menggunakan sistem mengingat keterlibatan mereka yang sangat tinggi sehingga sistem memenuhi kebutuhan mereka dengan lebih baik.

c. Memperkecil kesalahan disebabkan pada setiap versi *prototype* kesalahan segera terdeteksi oleh pemakai.

d. Pemakai memiliki kesempatan yang lebih banyak dalam meminta perubahan-perubahan.

    e. Mempersingkat waktu dalam mengembangkan sistem secara keseluruhan.
    f. Menghemat biaya (menurut penelitian, biaya penghematan dapat mencapai 10% hingga 20% dibandingkan dengan menggunakan metode SDLC tradisional).

2. Kekurangan

a. Sistem akan baik jika pemakai sungguh-sungguh meluangkan waktunya untuk menggarap *prototype*.
b. Dokumentasi sering terabaikan karena pengembang lebih berkonsentrasi pada tahap pengujian dan pembuatan *prototype*.
c. Waktu yang singkat dapat menghasilkan sistem yang tidak lengkap dan kurang teruji.
d. Jika terlau banyak proses pengulangan dalam membuat *prototype*, ada kemungkinan pemakai menjadi jenuh dan memberikan respon yang negatif.
e. Apabila *prototype* tidak dikelola dengan baik dapat mengakibatkan *prototype* tidak pernah berakhir karena usulan perubahan terlalu sering dan mudah untuk dipenuhi.

3.2.2.3 Alat Bantu Analisis dan Perancangan Berorientasi Objek

Menurut Bambang Hariyanto (2004 : 252) Analisis Sistem menggunakan pemodelan agar :

1. Fokus pada fitur-fitur sistem penting sambil meremehkan fitur-fitur kurang penting
2. Membahas perubahan-perubahan dan koreksi - koreksi kebutuhan-kebutuhan pembeli dengan ongkos yang murah dan resiko minimal.
3. Memferivikasi pemahaman analis sistem terhadap lingkungan pemakai dan didokumnetasikan dengan cara yang diperlukan oleh perancangan dan pemrograman membangun sistem.

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 state, 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 diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas yang 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.

3.2.3 Pengujian Software

Langkah terakhir sebelum sistem/perangkat lunak diserahkan kepada user adalah melakukan pengujian (*testing*) terhadap sistem/perangkat lunak tersebut.

3.2.3.1 *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)

Pada pengujian Kotak Hitam (*Black Box*), kita mencoba beragam masukan dan memeriksa keluaran yang dihasilkan. Kita dapat mempelajari apa yang dilakukan kotak, tapi tidak mengetahui sama sekali mengenai cara konversi dilakukan.

Teknik pengujian Kotak Hitam (*Black Box*) juga dapat digunakan untuk pengujian berbasis skenario, dimana isi dalam sistem mungkin tidak tersedia

untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan use-case dan informasi analisi yang lain.