BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

3.1.1 Sejarah Singkat *FCC*

Lembaga kursus ini bergerak di bidang bahasa asing (Inggris) yang terbentuk sejak dari tahun 2008 tepatnya pada tanggal 20 Juni. Yang didirikan oleh 9 orang pengajar di sebuah lembaga pendidikan yang membeli waralaba dari lembaga tersebut. Dimana lembaga ini bernama La-Ciel yang beralamatkan di Jalan Parakan Resik No. 3 Soekarno Hatta Bandung. Setelah berjalan kurang lebih 1 tahun, dikarenakan sesuatu hal lembaga ini memisakan diri dengan lembaga sebelumnya dan mengganti nama lembaga mereka sendiri dengan nama 9e (*Nine Education*) yang beralamatkan di Jalan Ir. H. Djuanda No. 230 dimana pemilik saham yang sempat berpindah tangan dengan suatu alasan. Dan 9e ini bergerak sampai di penghujung tahun 2010.

Para pemilik saham bersepakat untuk mendidirikan sebuah yayasan yang tentunya masih bergerak di bidang jasa khususnya pada bidang bahasa asing. Dan di tahun yang baru para pemilik saham sepakat untuk merubah *brand* pada lembaga tersebut. Di bawah naungan Yayasan yang bernama Yayasan Sinergi Pemuda Mandiri lembaga ini mengganti *brand* dengan nama FCC (*Foreign Language Course & Club*).

3.1.2 Visi dan Misi Perusahaan

Visi FCC (*Foreign Language Course & Club*) adalah SUCCESS :

1. *Simple*
2. *Unique*
3. *Complete*
4. *Comfortabl*e
5. *Effective*
6. *Sensation*
7. *Satisfying*

Misi :

1. *Method in transfering any kind of knowledge*
2. *Become the only one institution that equalize between social and bussi*nes
3. *Give full servise, facilities, and hospitality*
4. *Give homey & enjoy atmosphere*
5. *Apply the preciseaction*
6. *Always put a innovative variable*

3.1.3 Sturktur Organisasi Perusahaan

Berikut adalah struktur organisasi dari lembaga kursus FCC (*Foreign Language Course and Club*) :



Gambar 3.1 *Struktur Organisasi Lembaga Kursus FCC*

3.1.4 Deskripsi Tugas

1. Direktur
    a. Bertanggung jawab atas seluruh elemen yang ada di FCC.
    b. Bertanggung jawab akan urusan dengan pihak luar perusahaan.
    c. Menentukan target jangka panjang dan jangka pendek.
    d. Mengambil keputusan dan kebijakan perusahaan.
2. Divisi Keuangan
    a. Menyiapkan data-data keuangan dan anggaran pengeluaran perusahaan.
    b. Bertanggung jawab menangani pemasukan kas yang diperoleh dari penerimaan biaya kursus.
    c. Membuat laporan aliran kas baik itu pemasukan ataupun pengeluaran perusahaan.

3. Kepala Sekolah
    a. Memastikan progres perkembangan setiap member.
    b. Bertanggung jawab atas pengadaan sumber daya manusia.
    c. Mengawasi, merencanakan, mengarahkan, mengkoordinir dan mengevaluasi seluruh kegiatan yang berhubungan dengan member.
4. Divisi Pemasaran

    Bertugas melakukan promosi-promosi guna mendapatkan member yang akan belajar sesuai dengan target yang telah ditetapkan oleh atasan.
5. Bagian Kurikulum

    Bertugas untuk membuat, merancang dan mengembangkan metode pembelajaran guna mencapai metode pembelajaran yang terbaik.
6. Administrator
    a. Menyusun penjadwalan baik itu jadwal kelas, pengajar, shift pergawai dan lain-lain.
    b. Bertanggung jawab atas pengadaan alat tulis kantor.
    c. Melayani member.
7. Pengajar
    a. Bertanggung jawab secara penuh terhadap proses belajar mengajar dengan member.
    b. Bertanggung jawab atas perkembangan member.

3.2 Metode Penelitian

3.2.1 Desain Penelitian

Menurut Jonathan Sarwono (2006 : 79), desain penelitian bagaikan sebuah peta jalan bagi peneliti yang menuntun serta menentukan arah berlangsungnya proses penelitian secara benar dan tepat sesuai dengan tujuan yang telah ditetapkan. Desain penelitian ini termasuk kedalam penelitian yang bersifat deskriptif. Penelitian deskriftif adalah penelitian yang bertujuan untuk memperoleh ciri-ciri *variable*, dimana dalam penelitian ini adalah untuk memperoleh gambaran tentang kinerja program yang dirancang dan di implementasikan kepada pengguna.

3.2.2 Jenis dan Metode Pengumpulan Data

3.2.2.1 Sumber Data Primer (Wawancara dan Observasi)

1. Wawancara yaitu penulis melakukan wawancara langsung dengan pegawai FCC dan mencatat segala data yang diperlukan terutama dalam kegiatan akademik.
2. Observasi yaitu mengadakan pengamatan langsung untuk mendapatkan sistem informasi akademik yang lebih akurat terkait dengan masalah-masalah yang diteliti.

3.2.2.2 Sumber Data Sekunder (Dokumentasi)

Data sekunder adalah data yang digunakan untuk mendukung data primer, merupakan jenis data yang sudah diolah terlebih dahulu oleh pihak pertama, data sekunder diambil secara tidak langsung dari objek penelitian misalnya data ini diperoleh dari buku-buku, jurnal, tutorial, internet dan

artikel-artikel ilmiah yang dapat dikaji sebagai bahan rujukan dan landasan teoritis dalam pemecahan masalah. Adapun dokumentasi dari *FCC (Foreign Language Course & Club*) adalah :

1. Formulir pendaftaran
2. Absensi kehadiran member
3. Laporan penilaian
4. Kwitansi pembayaran kursus

3.2.3 Metode Pendekatan dan Pengembangan Sistem

3.2.3.1 Metode Pendekatan Sistem

Metode pendekatan yang digunakan adalah metode pendekatan berorientasi objek (*Object Oriented*). Menurut 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 sub-sistem berdasarkan struktur analisis dan arsitektur yang dibutuhkan.

3.2.3.2 Metode Pengembangan Sistem

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

Prototyping merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode prototyping ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dikehendakinya tanpa menyebutkan secara detail output apa saja yang dibutuhkan, pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang memperhatikan efisiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer.

Kunci agar model *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa *prototype* dibangun untuk mendefinisikan kebutuhan. *Prototype* akan dihilangkan sebagian atau seluruhnya dan

perangkat lunak aktual aktual direkayasa dengan kualitas dan implementasi yang sudah ditentukan.

3.2.3.3 Alat Bantu Analisis dan Perancangan

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. *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.

3. *Sequence* diagram

   Sequence diagram digunakan untuk memberikan gambaran detail dari setiap *use case* diagram yang telah dibuat sebelumnya. Setiap objek yang terlibat dalam sebuah use case digambarkan dengan garis putus-putus vertikal, kemudian pesan yang dikirim oleh objek digambarkan dengan garis horizontal secara kronologis dari atas ke bawah.

4. *Collaboration* diagram

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

5. *Class* diagram

   Class diagram menggambarkan class dan hubungan antar *class* di dalam sistem. *Class* diagram dibangun berdasarkan *use case* diagram, *sequence* diagram yang telah di buat sebelumnya.

   Class memiliki tiga area pokok :

   Nama (*stereotype*), Atribut, dan Metoda

   Atribut dan metoda dapat memiliki salah satu sifat berikut :

   1. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan.
   2. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.
   3. *Public*, dapat dipanggil oleh siapa saja.

6. *Component* Diagram

   Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

7. *Deployment/physical* Diagram

   Deployment atau *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.

3.2.4 Pengujian Software

Pengujian suatu system merupakan suatu bagian yang sangat penting untuk membangun suatu Software atau perangkat lunak yang bertujuan untuk menjamin bahwa Software yang dibangun memiliki kualitas yang sangat baik. Metode Pengujian software yang digunakan oleh penulis adalah suatu metode pengujian *BlackBox*. Pengujian *BlackBox* berfokus pada persyaratan fungsional perangkat lunak yang dibuat. Metode pengujian *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Pengujian *BlackBox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan performa.
5. Kesalahan inisialisasi dan terminasi.