BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

3.1.1 Sejarah singkat Perusahaan

S28 merupakan terobosan bidang usaha milik PT. Shafira Laras Persada. PT. Shafira Laras Persada sendiri merupakan sebuah sanggar busana muslim yang bernama SHAFIRA yang didirikan di Bandung pada tanggal 8 Januari 1989 di Jln. Ir. H.Juanda No.52 Lt.II yang dirintis oleh Feny Mustafa.

Seriring berkembangnya PT. Shafira Laras Persada di bidang *fashion* bernuansa muslim, PT. Shafira Laras Persada kembali merintis bidang bisnis baru di bidang kuliner yang diberi nama 28.

S2 didirikan pada tanggal 13 Maret 2010 yang terletak di Jln. Sulanjana No.28 Bandung, Jawa Barat. Nama S28 itu sendiri diambil dari nama alamat dimana S28 beroperasi, yang berarti Sulanjana 28. S28 memiliki tiga bidang bisnis yang masing-masing bidang bergerak di bisnis produksi yang berbeda. Adapun ketiga bidang tersebut dibagi menjadi tiga bagian, yaitu ;

1. S28 Cafe

   Sama halnya dengan cafe-cafe lainnya, disini S28 menyajikan berbagai sajian wisata kuliner yang dapat dinikmati oleh pengunjung. Sajian yang disajikan berupa menu nasional dan menu internasional dengan mengutamakan kepuasan pengunjung serta tersedia berbagai fasilitas lainnya yang yang dapat dinikmati oleh semau pengunjug, seperti lounge, dan hostpot internet.

2. S28 *Foodmart*

   Dalam bidang *foodmart*, S28 memiliki berbagai macam produk makanan dan minuman tradisional khas Jawa Barat. Bagi pengunjung yang datang langsung ke lokasi S28, dapat mencoba terlebih dahulu produk khas tradional sebelum membeli. Produk yang dicoba disajikan secara gratis oleh S28 *Foodmart*.

3. S28 *Bookstore*

   Dalam bidang *bookstore*, S28 menyediakan beraneka ragam jenis buku yang dapat dibaca langsung ditempat dan juga dapat dibeli oleh konsumen.

3.1.2 Visi dan Misi Perusahaan

Visi S28 Cafe :

Mewujudkan S28 Cafe sebagai pusat pembelanjaan oleh-oleh khas Jawa Barat

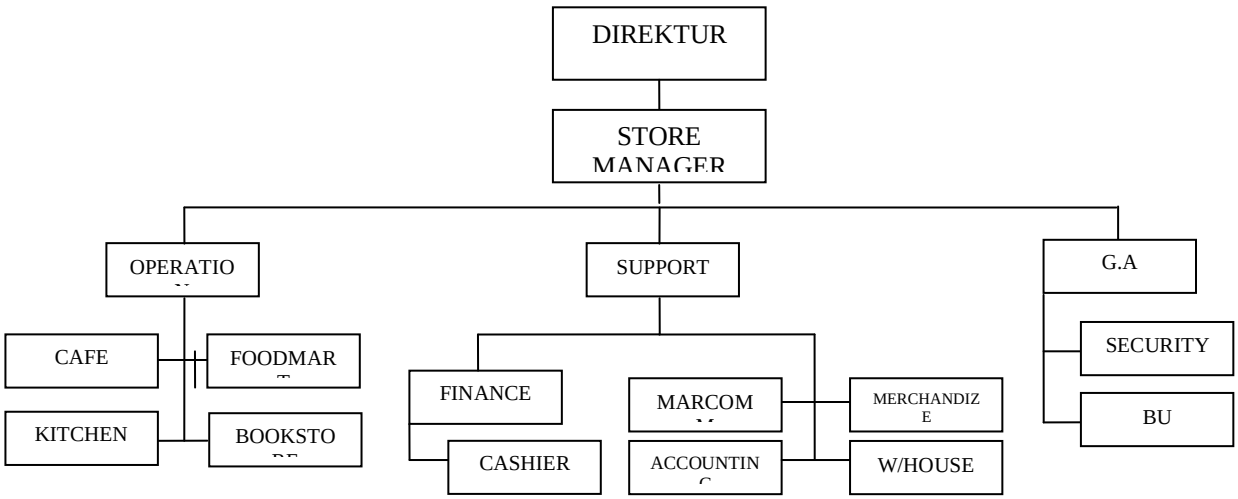
Misi S28 Cafe :

1. Mengutamakan kualitas dalam hal apapun yang dilakukan (pelayan) dan yang disajikan (produk).
2. Memberikan produk yang bermutu tinggi, dengan tetap memberikan harga yang bersaing kepada konsumen.
3. Mengembangkan inovasi-inovasi yang terbaik dalam produk maupun pelayanan dengan tetat pada koridor tradisional.

3.1.3 Struktur Organisasi

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 personel 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.

Struktur organisasi S28 CAFE dapat digambarkan sebagai berikut:



Gambar 3.1 struktur organisasi

3.1.4 Deskripsi Tugas

1. Direktur.

   Direktur merupakan pengurus perusahaan yang bertanggung jawab penuh dalam segala hal yang 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. *Store* Manager

   Store Manager bertugas dalam mengelola sumber daya yang ada diperusahaan secara efektif dan efisien untuk mencapai target yang telah ditetapkan oleh perusahaan. Dalam hal ini, *store* manager manager mempunyai tanggung jawab mengorganisir oulet yang ada, melakukan pelayanan dan pemasaran pasca jual, merancang strategi jangka panjang dan pendek serta merealisasikan target penjualan. *Store* manager akan mengkoordinir langsung dengan manager operasional, *support* dan *general affair*.

3. *Operation*

   Manager Operasional *(operation)* bertugas mengkordinasikan tugas harian kepada semua personel bidang operasional, dalam hal ini bidang yang koordinir adalah personel cafe, dapur *(kitchen)*, *foodmart,* dan *bookstore.* Juga melakukan evaluasi keluar dan masuknya barang, menetapkan harga, dan juga pengambil keputusan bilamana diperlukan. Bekerjasama dengan pihak luar dalam penyelenggaraan event.

4. *Support*

   Manager Pendukung *(support)* bertugas dalam memenuhi kebutuhan pendukung bagi jalannya operasional harian dengan melakukan koordinasi langsung dengan manager operasional. Manager pendukung juga

melakukan koordinasi tugas harian kepada personel bidang *Finance, Marketing Communication, Accounting, Merchandize*, dan *W/house*.

5. *General Affair*

   Manager urusan umum *(General Affair)* bertugas dalam melakukan koordinasi langsung dengan *security* dan staff bidang umum. *General affair* juga meng*handle* pekerjaan umum, seperti pengurusan ijin iklan promosi. Dan juga melakukan koordinasi dengan manager terkait.

6. Cafe

   Personel bidang cafe bertugas dalam mengatur kelayakan jalannya operasional harian cafe, seperti mengantar pesanan kepada konsumen, mengatur tata letak dekorasi cafe dan berhubungan langsung dengan manager operasional demi lancarnya aktifitas operasional cafe.

7. *Kitchen*

   Personel bidang dapur (*kitchen)* bertugas dalam menjalankan operasional didapur dalam hal ini menyiapkan segala pesanan konsumen, mengontrol jalannya aktifitas semua pengerjaan didapur, melakukan *loading supply* dan berhubungan langsung dengan manager operasional demi lancarnya aktifitas operasional cafe.

8. *Foodmart*

   Personel bidang *foodmart* bertugas dalam mengatur jalannya operasional di bidang penjualan makanan dan minuman tradisional. Dalam hal ini, personel *foodmart* melakukan koordinasi langsung dengan manager operasional dalam memenuhi semua kebutuhan operasional harian.

9. *Bookstore*

Personel bidang bookstore bertugas dalam mengelola aktifitas yang ada di bagian bookstore, seperti menyusun tata letak buku berdasarkan kategori, mengatur dan menjaga area bookstore agar selalu terlihat nyaman dan bersih. Personel bidang *bookstore* juga melakukan koordinasi langsung dengan manager operasional dalam mengontrol jalannya operasional *bookstore* sebagaimana mestinya.

10. Finance

Bertugas untuk mengatur keuangan dalam S28. Baik itu pendapatan dari hasil penjualan, dan juga pengeluaran S28, baik itu tentang sewa gedung, listrik, kebersihan dan juga perihal tentang beban dan biaya operasional S28. Dan melakukan koordinasi langsung dengan cashier dalam hal keuangan.

11. Cashier

Bertugas untuk melayani konsumen yang melakukan transaksi, dalam hal ini, masing-masing bidang usaha memiliki cashier masing-masing demi memudahkan pengaturan pendapatan masing-masing bidang. Cashier juga berkoordinasi langsung dengan personel bidang keuangan dalam memberikan laporan keuangan.

12. *Marketing Communication*

Bertugas untuk menangani komunikasi internal dan external perusahaan. Juga bertanggung jawab dalam bidang komunikasi brand, melalukan pemantauan distribusi, sales dan pengembangan produk baru. *Marketing*

communication juga melakukan koordinasi langsung dengan manager yang terkait dengan kelancaran bidang marketing.

13. *Acconting*

Bertugas untuk melakukan pencatatan keuangan terhadap kebutuhan pengeluaran dan pemasukan, accounting juga bertugas melakukan audit keuangan setiap bulan dan membuat laporan pencatatan terhadap semua yang berkaitan dengan keuangan.

14. *Merchandize*

Bertugas dalam memberikan pelayan kepada konsumen yang menyangkut dengan cindera mata *(marchandize)*. Merancang dan memproduksi cindera mata perusahaan.

15. *Ware House*

Ware house (gudang) bertugas dalam hal pengecekan barang masuk dan barang keluar, melakukan pemeliharaan barang yang berada digudang, melakukan pencatatan terhadap jumlah barang masuk dan barang keluar.

16. *Security*

Bertugas untuk menjaga keamanan terhadap perusahaan, mengatur urusan perparkiran dan menjaga kendaraan pegawai S28 dan konsumen, juga menjaga loker para pegawai S28

17. *BU*

Bidang umum bertugas untuk melakukan hal-hal yang berkaitan dengan pemasangan spanduk promosi, menjaga kebersihan area S28 dan melakukan pencatatan setiap ada tamu yang ingin bertemu dengan pengelola S28.

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 dengan merancang dan mendesain program berdasarkan analisis sistem yang diusulkan sesuai dengan masalah yang ada di dunia aktual (lapangan) dan sesuai dengan kebutuhan system untuk di implementasi nantinya.

3.2.1 Jenis dan Metode Pengumpulan

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

3.2.2.1 Sumber Data Primer

Data primer yaitu data yang diperoleh langsung dari sumber yang akan diteliti, dalam hal ini adalah *support* manager pada S28 Cafe. 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 secara langsung terhadap operasional perusahaan, tetapi peneliti tidak ikut langsung terhadap kegiatan perusahaan.
b. Wawancara, yaitu teknik pengumpulan data dengan cara tanya jawab secara langsung dengan para karyawan yang berhubungan dengan penelitian ini. Wawancara dilakukan dilakukan dengan *support* manager yang dapat memberikan data yang relevan dengan masalah yang diteliti oleh penulis.

3.2.2.2 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, visi dan misi, 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 pendekatan yang digunakan oleh penulis adalah dengan menggunakan pendekatan sistem berorientasi objek. Didalam melakukan penelitian ini, penulis melakukan penelitian terhadap proses pembelian produk *foodmart* yang sedang berjalan pada S28. Didalam proses pembelian penulis mengamati objek dan prosedur yang sedang berjalan. Adapun objek yang diamati adalah actor pelaku sistem yaitu *sales counter* dan *customer* karena kedua actor itu sangat berpengaruh terhadap jalannya proses penjualan produk *foodmart* pada S28. Adapun alat-alat yang digunakan untuk melakukan pendeketan dalam analisis dan pemograman berorientasi objek yaitu dengan notasi UML dengan membuat tujuah diagram yaitu, *Use case diagram, Activity Diagram, Sequence*

diagram, Collaboration diagram, Class diagram, Component diagram, Deployment diagram.

1. *Use Case*

   Use Case adalah untuk mendeskripsikan bagaiman sistem akan terlihat dimata user. *Use case* terdiri dari sekumpulan *scenario* yang di lakukan oleh *actor*, sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna. Didalam *use case* terdapat <*include*> yang berfungsi untuk menyatakan sebuah *use case* adalah bagian dari *use case* lain. <*extend*> digunakan untuk membuat *use case* baru dengan menambahkan langkah-langkah pada *use case* yang sudah ada.

2. *Sequence Diagram*

   Sequence diagram menggambarkan perilaku pada sebuah *scenario*. Program ini menunjukka sejumlah objek dan *message* (pesan) yang diletakkan diantara objek-objek didalam *use case 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.

3. *Activity Diagram*

   Activity diagram adalah sebuah teknik untuk mendeskripsikan sebuah logika procedural proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart* akan tetapi perbedaan dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku parallel, sedangkan *flowchart* tidak bisa. *Activity diagram* bisa

digunakan untuk menunjukkan siapa yang mengerjakan apa dengan teknik partision.

4. *Collaboration Diagram*

   Collaboration diagram adalah bentuk lain dari *sequence diagram*. Bila *sequence* diorganisir menurut waktu, maka *collaboration* diorganisir menurut ruang/*spase. Collaboration diagram* merupakan asosiasi diantara objek parallel didekat gasis asosiasi yang menunjukkan *message*, sedangkan konten *message* ditunjukkan dengan label. *Collaboration diagram* memungkinkan untuk memodelkan pengiriman sebuah *message* kebanyak objek pada *class* yang sama. Demikian juga halnya untuk menunjukkan adanya objek aktif yang mengendalikan aliran *message.*

5. *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.

6. *Component Diaram*

   Component diagram merepresentasikan dunia *real* item yaitu *component software*. *Component diagram* menetap dikomputer bukan dibenak analis. *Component* bisa diakses melalui *interfacenya* yaitu operasi-operasi relasi antar *component* dan *interfacenya* disebut *realization.* Suatu komponen

mengakses s*ervices-service* yang ada di *component* dengan cara *import interface*. Sedangkan *component* yang menyediakan *service* menggunakan *export interface.*

7. *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.

Menurut Bambang Hariyanto (2004 : 86) keunggulan pendekatan berorientasi objek adalah sebagai berikut :

1. Bekerja yang mendekati kondisi 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).

3.2.3.2 Metode Pengembangan Sistem

Prototyping adalah proses iteratif dalam pengembangan sistem di mana kebutuhan diubah ke dalam sistem yang bekerja (*working system*) yang secara terus menerus diperbaiki melalui kerjasama antara pengguna dan analis. *Prototype* juga bisa dibangun melalui beberapa *tool* pengembangan untuk menyederhanakan proses, Hanif Al Fatta (2007:36).

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.

Adapun alasan penulis mengambil metode *prototype* dalam penelitian ini adalah mengacu pada kepuasan pengguna (*user)* dan mendapatan *feedback* langsung dari pengguna, sehingga proses dalam pembuatan sistem bisa lebih kompleks dengan menambah dan memperbaiki kekurangan dari prototype yang diajukan pertama kali.

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.

3.2.3.3 Alat Bantu Analisis dan Perancangan Berorientasi Objek

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.

1. 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.

2. Diagram Sekuen (*Sequence Diagram*) bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu.
3. 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

memeuat simpul-simpul beserta komponen-komponen yang ada didalamnya. *Deployment diagram* berhubungan dengan diagram komponen. dimana *deployment diagram* memuat satu atau lebih komponen-komponen.

3.2.4 Pengujian Software

Terdapat dua metode pengujian dalam melakukan pengujian terhadap sebuah sistem. Adapun metode tersebut adalah :

1. Metode pengujian *Black Box*

   Metode pengujian *Black Box* digunakan untuk menguji fungsi-fungsi khusu perangkat lunak yang dirancang. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut. Dari keluaran yang dihasilkan dalam memenuhi kebutuhan pengguna dapat diukur dan dapat diketahui kesalahannya

2. *White Box*

   Metode pengujian *White Box* digunakan untuk mengetahui cara kerja perangkat lunak secara internal. Pengujian dilakukan untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang.

   Dalam penelitian ini penulis menggunakan metode pengujian dengan menggunakan *Black Box* karena pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak (software) yang dibuat.

Dengan demikian, pengujian *Black Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses databse eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi.