BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Adapun prmbahasan mengenai Objek Penelitian dapat dilihat pada penjelasan di bawah ini.

3.1.1 Sejarah Singkat Perusahaaan

Cikal bakal Yayasan Darul Hikam dirintis oleh KHE. Hasbullah Hafidzi pada tahun 1942, setelah beliau selesai belajar di pesantren Allanah Cianjur. Kegiatan pertama yang dilaksanakan adalah menyelenggarakan Madrasah Islam di Kampung Cisitu Girang Bandung. Pada Revolusi fisik, Madrasah ini dihentikan karena rakyat harus mengungsi meninggalkan bandung. Sekembalinya dari mengungsi, pada tahun 1949, Madrasah dibuka kembali dan dinaikkan statusnya menjadi Sekolah Dasar Islam dengan nama Sekolah Rakyat Muslimin. Alhamdulillah pada tahun 1951, sekolah sudah memiliki bangunan milik sendiri di atas lahan pinjaman milik Ustadz Abdusalam.

Pada tahun 1953 dibuka SMP dengan waktu belajar siang hari dan tidak lama kemudian SMP ini dipindahkan ke Jalan Puyuh No.5 dengan nama SMP Muslimin. Sementara itu pada tahun 1953 bangunan SD Islam yang berlokasi di Cisitu mendapat musibah yaitu ambruk tertimpa angin. Para murid dipindahkan ke sekolah-sekolah sekitarnya. Setelah bangunan di Cisitu hancur, pada tahun 1964 dibuka Taman

Kanak-Kanak Islam bertempat di rumah Hj. Dedeh Ruyati Hasbullah di Jalan. Ir. H. Juanda No. 212 Bandung. Seusai kegagalan Pemberontakan G 30 S/PKI, atas izin Allah Swt dan perjuangan keras KHE Hasbullah Hafidzi disertai dukungan istri dan keluarganya, dibangun Masjid Darul Hikam berukuran 12 X 18 di Jalan. Ir. H. Juanda No. 285 Bandung (lokasi sekarang).

Pada tahun 1980, berdirilah SMA Darul Hikam berdasarkan Keputusan Menteri Pendidikan tanggal 11 September 1980 Nomor 0222h/0/1980 dan Nomor 0241/0/1980 dengan status “Terdaftar”. Perkembangan yang baik dari tahun ke tahun menjadikan SMA Darul Hikam mendapatkan status “Diakui” pada tahun 1984. Dengan perbaikan mutu yang terus menerus SMA Darul HIkam mendapat status “Disamakan” pada tahun 1994 sampai 2005. Dan akhirnya SMA Darul Hikam berstatus Terakreditasi "A".

3.1.2 Visi dan Misi SMA Darul Hikam

Adapun visi dan misi SMA Darul Hikam adalah:

3.1.2.1 Visi SMA Darul Hikam

Visi adalah harapan tentang masa depan organisasi yang realistis. Bisa dicapai dan menarik, atau penjabaran tujuan kemana organisasi harus menuju masa depan yang lebih baik.

Visi SMA Darul Hikam adalah : ” *Menjadi sekolah Islam terbaik di Jawa Barat melalui Budaya (Jatidiri, Ciri Khas, dan Keunggulan) Berakhlak Berprestasi”.*

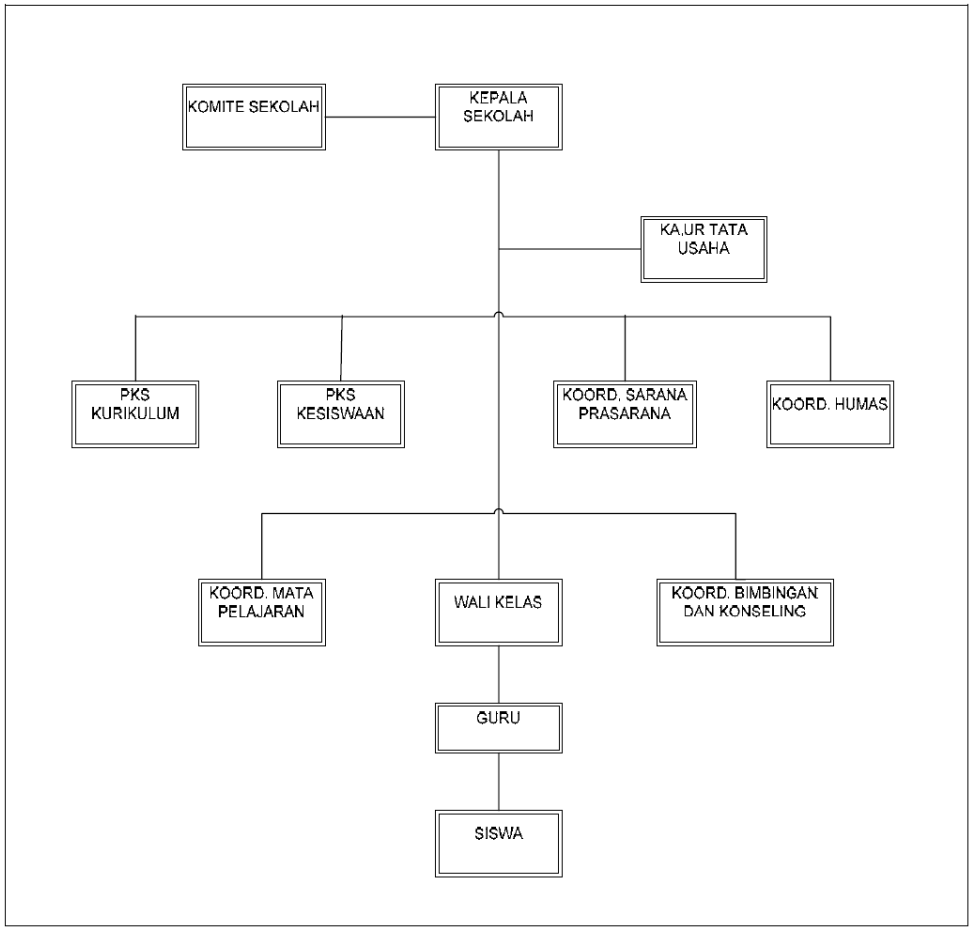
3.1.2.2 Misi SMA Darul Hikam

Misi SMA Darul Hikam yaitu:

1. Melaksanakan pendidikan Islam secara utuh, terpadu, dan sempurna untuk membangun akhlak karimah Siswa dan semua civitas akademika.
2. Melaksanakan pendidikan Islam secara utuh, terpadu, dan sempurna untuk meraih prestasi Siswa dan civitas akademika dalam berbagai bidang kependidikan.
3. Membangun citra baik sekolah Islam sebagai bagian dari system pendidikan nasional.
4. Membangun silaturahim dan kerjasama dengan orang tua siswa dalam proses pendidikan Islam bagi putra dan putri mereka

3.1.3 Struktur Organisasi SMA Darul Hikam

Adapun struktur organisasi SMA Darul Hikam dapat dilihat pada gambar di berikut ini.



Gambar 3.1

Struktur Organisasi SMA Darul Hikam

3.1.4 Deskripsi Tugas

1. Komite Sekolah

   SMA Darul Hikam berada dibawah naungan Yayasan Darul Hikam yang memiliki dewan komite sekolah yang bertugas untuk mengawasi dan memantau setiap perkembangan dari lembaga-lembaga pendidikan yang bernaung dibawah Yayasan Darul Hikam.

2. Kepala Sekolah

   SMA Darul HIkam dipimpin oleh seorang kepala sekolah yang bertanggung jawab terhadap komite sekolah. Kepala sekolah berfungsi dan bertugas sebagai edukator, manajer, administrator, supervisor, leader, inovator dan motivator serta memastikan kelangsungan SMA Darul Hikam ke sasaran yang telah ditetapkan.

3. Kepala Urusan Tata Usaha betugas untuk menyusun program tata usaha sekolah, mengurus administrasi ketenagaan dan siswa, membina dan pengembangan karier pegawai tata usaha sekolah, menyusun administrasi perlengkapan sekolah, menyusun dan penyajian data statistik sekolah, mengkoordinasikan dan melaksanakan K6 (Keamanan, Kebersihan, Ketertiban, Keindahan, Kenyamanan dan Kekeluargaan), membuat laporan kegiatan tata usaha.
4. Pembantu Kepala Sekolah Bidang Kurikulum bertugas menyusun program pengajaran, pembagian tugas guru dan jadwal pelajaran, jadwal ulangan dan evaluasi, kriteria kenaikan, ketidaknaikan atau kelulusan, membina lomba akademis, dan MGMP (Musyawarah Guru Mata Pelajaran).
5. Pembantu Kepala Sekolah Kesiswaan bertugas menyusun program pembinaan OSIS, melaksanakan pembimbingan dan pengarahan kegiatan OSIS, melaksanakan koordinasi K6, pemilihan siswa teladan atau penerima

beasiswa, mutasi siswa, program ekstra kurikuler, dan membuat laporan kegiatan kesiswaan secara berkala.

6. Koordinator Sarana Prasarana bertugas menyusun rencana kebutuhan sarana dan prasarana, mengkoordinasikan pendayagunaan sarana dan prasarana, pengelola pembiayaan alat-alat pengajaran, dan menyusun laporan pelaksanaan urusan sarana dan prasarana secara berkala.
7. Koordinator Humas bertugas mengatur dan menyelenggarakan hubungan sekolah dengan orang tua atau wali siswa, membina hubungan antar sekolah, komite sekolah, lembaga dan instansi terkait, dan membuat laporan pelaksanaan hubungan masyarakat secara berkala.
8. Koordinator Mata Pelajaran bertugas mengatur dan menyusun mata pelajaran sesuai dengan kurikulum yang ada, menyusun bahan ajar bagi tiap-tiap mata pelajaran dan mengatur jadwal mata pelajaran yang akan diterapkan.
9. koordinator Bimbingan dan Konseling bertugas merencanakan, melaksanakan dan mengevaluasi Program Kerja Bimbingan dan Konseling di Sekolah, melaksanakan Konseling dengan siswa, melaksanakan konsultasi, koordinasi dengan guru dan orang tua siswa.

3.2. Metode Penelitian

Metode penelitian adalah cara ilmiah untuk mengumpulkan data dengan tujuan dan kegunaan tertentu.

3.2.1 Desain Penelitian

Penelitian yang akan dilakukan penulis adalah mengunakan metode deskriptif, penelitian deskriptif adalah penelitian yang fokusnya untuk menggambarkan kondisi responden berdasarkan variabel-variabel yang di teliti. Dimana dalam metode ini menggambarkan semua data yang kemudian dianalisis dan dibandingkan berdasarkan kenyataan yang sedang berlangsung dan selanjutnya mencoba untuk memberikan pemecahan masalahnya.

Metode *action* atau tindakan merupakan penelitian langsung, disertai dengan praktek di lapangan. Membuat suatu program yang akan dilaksanakan secara sistematis dan terencana, serta mempunyai nilai perbaikan yang signifikan. Penelitian tindakan ini lebih efektif, karena akan terlihat langsung hasilnya.Salah satu syarat dalam melakukan penelitian tindakan adalah adanya keinginan dari orang yang memilki masalah untuk mengidentifikasi masalah yang ada dan mempunyai keinginan untuk memecahkannya.

3.2.2 Jenis dan Metode Pengumpulan data

Metode pengumpulan data merupakan suatu teknik atau cara untuk memperoleh, mencari, mengumpulkan dan mencatat data yang di gunakan untuk menyusun karya ilmiah. Adapun pengumpulan data itu sendiri terbagi menjadi dua yaitu data primer dan data sekunder.

3.2.2.1. Sumber Data Primer

Data primer adalah data yang diperoleh langsung dari unit pengamatan (instansi) atau responden penelitian baik melalui pengamatan maupun pencatatan terhadap objek penelitian.

Dalam penelitian ini penulis mengumpulkan data primer dengan menggunakan teknik:

1. Pengamatan Langsung (Observasi)

Observasi merupakan suatu cara pengumpulan data melalui pengamatan dan pencatatan oleh pengumpul data terhadap gejala atau peristiwa yang diselidiki pada obyek penelitian secara langsung tetapi tidak ada interaksi dengan objek yang diteliti dan tanpa sepengetahuan objek yang sedang diteliti tersebut dengan cara melakukan tinjauan langsung ke lapangan.

Penulis melakukan tinjauan langsung ke lapangan untuk mencari data-data yang diperlukan terhadap objek yang diteliti di bagian tata usaha SMA Darul hikam Bandung.

2. Wawancara

Wawancara merupakan proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab, sambil bertatap muka antara si penanya atau pewawancara dengan si penjawab atau responden. Dalam penelitian ini penulis melakukan wawancara secara langsung dengan mengajukan berbagai pertanyaan

kepada pihak yang ikut terlibat langsung yaitu diantaranya pada bagian kurikulum bidang kesiswaan.

3.2.2.2. Sumber Data Sekunder

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 lain-lain.

Dokumentasi adalah teknik pengumpulan data dengan cara mengumpulkan dokumen-dokumen yang berhubungan dengan objek penelitian. Dalam hal ini, dokumen yang diperoleh akan dianalisis agar diperoleh data yang sesuai dengan penelitian. Adapun dokumen-dokumen yang dapat digunakan dalam tahap analisis sistem yaitu:

a. Data pendaftaran siswa baru.
b. Data siswa dan guru SMA Darul Hikam Bandung.
c. Data penilaian.
d. Data mata pelajaran.
e. Data pembagian kelas

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Dalam pembangunan suatu sistem berbasis teknologi informasi diperlukan suatu pendekatan dan pengembangan sistem yang akan menentukan proses penyelesaian rekayasa perangkat lunak, adapun pendekatan sistem yang dilakukan penulis adalah dengan menggunakan pendekatan berorientasi objek (*object oriented*) dan pengembangan sistem dengan menggunakan model *prototype*.

3.2.3.1. Metode Pendekatan

Menurut Bambang Heriyanto (2004:269) pendekatan berorientasi objek adalah cara memandang persoalan menggunakan model-model yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas.

Pendekatan sistem berorientasi objek berbeda dengan pendekatan konvensional yang memandang perangkat lunak sebagai fungsi dan data yang terisolasi. Pada pendekatan konvensional kebanyakan berfokus pada data terutama pada basis data dan pemodelan informasi. Sementara pada pendekatan berorientasi objek berpusat pada objek yang mengkombinasikan data dan fungsionalitas.

Metode pendekatan sistem yang di gunakan adalah pendekatan dengan *Object Oriented* yang di visualisasikan dengan UML dan di antara nya adalah sebagai berikut *: Use Case, , Activity Dagram, Sequence Diagram, Class*

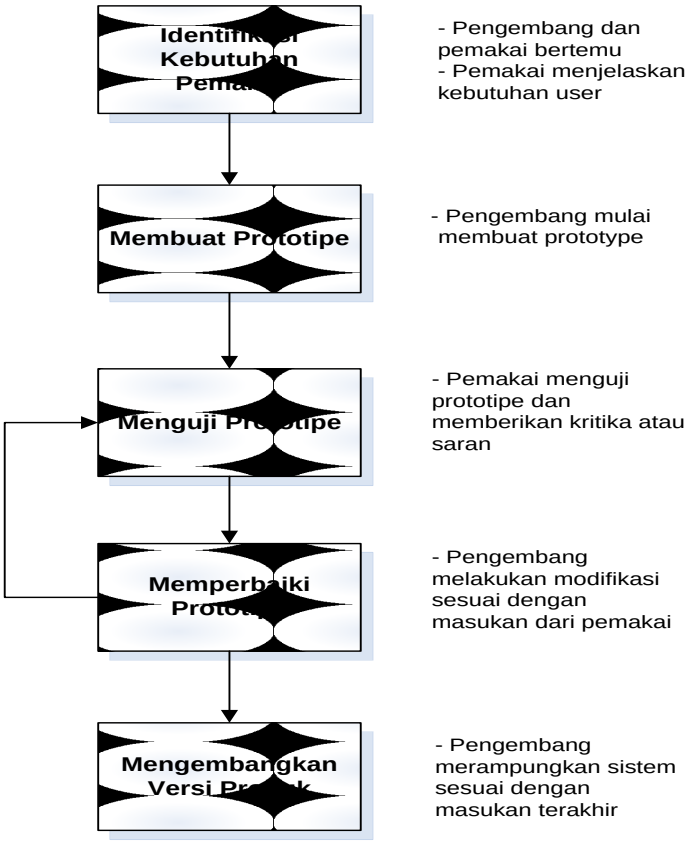
Diagram, Collaboration Diagram, Component Diagram dan *Deployment Diagram.*

3.2.3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang penulis gunakan yaitu menggunakan metode *Prototype* karena metode ini menawarkan bagi pengembang sistem apabila tidak memiliki kepastian terhadap efisiensi *algoritma*, kemampuan penyesuaian dari sebuah sistem operasi atau bentuk-bentuk yang harus dilakukan oleh interaksi manusia dengan mesin. Dilihat dari situasi tersebut metode *prototype* menawarkan pendekatan yang terbaik.

Dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu langsung untuk mendefinisikan keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang telah diketahui dan mencari bidang-bidang yang masih memerlukan pendefinisian kemudian dilakukan perancangan kilat terhadap kebutuhan yang telah teridentifikasi pada pertemuan. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan/pemakai. Perancangan ini menuntun pembangunan perangkat lunak yang akan diberikan kepada pemakai. Selanjutnya *prototype* itu dievaluasi oleh pemakai dan digunakan sebagai landasan untuk memperbaiki spesifikasi kebutuhan. Proses

ini akan berulang sampai *prototype* yang dikembangkan memenuhi seluruh kebutuhan pemakai.



Gambar 3.2 Mekanisme Pengembangan Sistem dengan *Prototype*

(**Sumber** : Abdul Kadir, Pengenalan Sistem Informasi 2003, Andi. Yogyakarta)

Tahapan dalam metode *Prototype* :

1. Identifikasi kebutuhan (Data)

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

2. Membangun *prototyping*

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

   a. Merancang sistem

      Dalam tahap ini prototipe dirancang secara terstuktur dari proses basis data hingga rancangan menu program.

   b. Mengkodekan sistem

      Dalam tahap ini prototyping yang sudah dirancang diterjemahkan ke dalam bahasa pemrograman yang sesuai.

3. Menguji sistem

   Setelah sistem sudah menjadi suatu perangkat lunak, harus diuji dahulu sebelum digunakan.

4. Evaluasi Sistem

   Pelanggan mengevaluasi apakah sistem yang sudah buat sudah sesuai dengan yang diharapkan, apabila belum sesuai maka tahapan 2 dan 3 diulang

kembali hingga sesuai dengan yang diharapkan oleh pelanggan dan lanjut ke tahap berikutnya.

5. Penerapan sistem

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

3.2.3.3. Alat Bantu Analisis dan Perancangan

1. UML (Unified Modelling Language)

Dalam suatu proses pengembangan software, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegosiasikan, dapat dikatakan bahwa pada tahap rancangan. merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool/model untuk merancang pengembangan software yang berbasis object-oriented adalah UML. Alasan mengapa UML digunakan adalah, pertama, scalability dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua, dynamic modeling, dapat dipakai untuk pemodelan sistem dinamis dan real time.

1) *Actor*

Menurut Bambang Heriyanto (2004:269) Dalam pemodelan sistem dengan UML, actor adalah seseorang atau sesuatu yang berinteraksi dengan sistem. Yang dimaksud dengan berinteraksi adalah actor mengirim

atau menerima pesan ke atau dari sistem atau mempertukarkan informai dengan sistem.

Secara prinsip dapat kita kenali 3 jenis actor untuk hampir semua sistem/perangkat lunak yang kita kembangkan, yaitu sebagai berikut :

1. Aktor yang pertama yaitu orang-orang yang hadir secara fisik, atau para pengguna. Mereka adalah aktor yang paling umum dan hadir di setiap sistem/perangkat lunak.
2. Aktor yang kedua yaitu sistem lain.
3. Aktor yang ke tiga yaitu waktu menjadi aktor ketika ia memicu event-event tertentu bagi sistem/perangkat lunak yang kita kembangkan.

2) *Use-case Diagrams*

Use-case Diagrams digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem.

Menurut Bambang Heriyanto (2004:267) Use Case adalah interaksi antara actor eksternal dan sistem, hasil yang dapat diamati oleh actor, berorientasi pada tujuan, dideskripsikan di diagram use-case dan teks.

Use-case Diagrams menyediakan cara medeskripsikan pandangan eksternal terhadap sistem dan interaksi-interaksinya dengan dunia luar. Dengan cara ini diagram *use-case* menggantikan diagram konteks pada pendekatan konvensional. Untuk penciptaan model *use-case* melibatkan

pendefinisian sistem, pencarian aktor-aktor dan *use-case*, mendeskripsikan *use-case* dan mendefinisikan hubungan antar *use-case* dan terakhir adalah melakukan validasi model.

Use-case bertindak sebagai mekanisme terstuktur untuk diagram-diagram interaksi. Umumnya, satu diagram interaksi digambarkan untk masing-masing *use-case* diagram. Salah satu bahaya *use-case* merupakan bagian vital dari pengembangan berorientasi objek. Kita seharusnya menggunakannya setiap ingin memahami kebutuhan-kebutuhan sistem.

3) *Class Diagrams*

Class menggambarkan keadaan (atribut / properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda / fungsi).

Class Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class Diagram* juga menunjukkan property dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunkan istilah fitur sebagai istilah umum yang meliputi property dan operasi sebuah *class*.

Class diagrams menggambarkan struktur dan deskripsi *class*, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, agregasi dan hubungan dinamis.

Class memiliki tiga area pokok, yaitu:

a) Nama
b) Atribut
c) Metoda

Class dapat merupakan implementasi dari sebuah interface, yaitu *class* abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian interface mendukung resolusi metoda pada saat *run-time*.

Hubungan antar *Class*:

a. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain.
b. Agregasi, yaitu hubungan yang menyatakan bagian.
c. Pewarisan, yaitu hubungan hierarkis antar *class*. Class dapat diturunkan dari class lain dan mewarisi semua stribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya.
d. Hubungan dinamis, yaitu rangkaian pesan yang di-*passing* dari satu *class* kelapada *class* lain.

4) *Component Diagrams*

Menurut Bambang Heriyanto (2004:293) Komponen adalah bagian fisik dan dapat diganti sistem yang memenuhi dan menyediakan realisasi sekumpulan antar muka.

Component Diagrams menunjukan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini memodelkan pandangan implementasi fisik dari sistem. *Component diagrams* berisi:

a. Komponen
b. Antarmuka
c. *Dependency*, generalisasi, asosiasi dan realisasi
d. Paket, untuk mengelompokkan elemen-elemen model menjadi potongan-potongan besar.

5) *Deployment Diagrams*

Deployment Diagrams digunakan untuk memodelkan aspek fisik dari sistem berorientasi objek, yaitu memodelkan konfigurasi node-node pengolahan waktu jalan dan komponen-komponen yang tinggal di node-node itu.

Deployment Diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana. Deployment diagram sangatlah sederhana.

Deploy Diagrams bisa juga berisi komponen-komponen, masing-masing komponen itu berada di suatu node. Juga dapat berisi paket atau subsistem yang digunakan mengelompokkan elemen-elemen di model menjadi potongan-potongan.

Penggunaan model *deployment* yaitu:

a. Memodelkan *embedded system*

b. Memodelkan sistem *client/server*

c. Memodelkan sistem tersebar penuh

6) *Sequence Diagrams*

Sequence diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sebuah *Sequence diagram,* secara khusus, menjabarkan behavior sebuah scenario tunggal. Diagram tersebut menunjukkan sebuah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *use case*.

*Sequence diagram*s biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu.

7) *Collaboration Diagrams*

Collaboration diagrams mendefinisikan peran-peran yang dimainkan letika satu tugas dilakukan. Peran-peran dimainkan oleh instan-instan yang berinteraksi.

Collaboration diagrams menyatakan komunikasi diantara objek-objek yang menunjukan pesan-pesan yang ada, urutan pesan dan hubungan antar objek-objek.

Collaboration diagrams digunakan untuk:

a. Pandangan dalam arti perilaku sistem, berfokus pada link-link di antara objek-objek
b. Ilustrasi dari suatu diagram *use-case*
c. Menyatakan objek-objek yang diperlukan untuk merealisasikan suatu layanan
d. Memeriksa jalur-jalur pengaksesan.

Collaboration diagrams menekankan pada organisasi objek yang berpartisipasi di interaksi. *Sequence diagrams* dan *collaboration diagrams* sama-sama menunjukkan interaksi (aspek dinamis). *Sequence diagrams* fokus pada waktu sedangkan *collaboration diagrams* fokus pada ruang. Sebagaimana *sequence diagrams*, *collaboration diagrams* juga dapat digunakan untuk mengilustrasikan eksekusi satu operasi, *use-case* atau skenario interaksi di sistem.

8) *Activity Diagrams*

Menurut Bambang Heriyanto (2004:325) *Activity diagrams* adalah diagram flowchart yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain. Kita menggunakan diagram ini untuk memodelkan aspek dinamis sistem. Aktivitas adalah eksekusi nonatomik yang berlangsung di *state machine*. Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas berupa operasi-operasi dan aktivitas-aktivitas di *use-case*.

Activity diagram dapat digunakan untuk:

a. Pandangan dalam yang dilakukan di operasi.
b. Pandangan dalam bagaimana objek-objek bekerja.
c. Pandangan dalam di aksi-aksi dan pengaruhnya pada objek-objek.
d. Pandangan dalam dari suatu *use-case.*
e. Logik dari proses bisnis.

3.2.4 Pengujian Software

1. *Black Box Testing*

Menurut Bambang Heriyanto (2004:569) Pengujian adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan

yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan yang terjadi.

Pengujian black box adalah pengujian aspek *fundamental* sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar.

Pada pengujian *black-box*, kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sedini mungkin di proses pengembangan perangkat lunak.

Pada pengujian *black box*, mencoba beragam masukan dan memeriksa keluaran yang dihasilkan. Teknik pengujian *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 analisis yang lain.

Adapun faktor-faktor pengujian *black-box* adalah :

1. *File integrity*

   ada data yang dimasukkan melalui aplikasi akan tidak bias diubah. Prosedur yang akan memastikan bahwa file yang digunakan benar dan data dalam file tersebut akan disimpan sekuensial dan benar.

2. *Service levels*

Menekankan bahwa hasil yang diinginkan didapat dalam waktu yang diinginkan oleh *user*. Untuk mencapai keinginan tersebut, harus dilakukan penyesuaian antara keinginan *user* dengan sumber daya yang ada.

3. *Ease of use*

Menekankan perluasan usaha yang diminta untuk belajar, mengoprasikan dan menyiapakan inputan, dan menginterpretasikan output dari sistem. Faktor ini tersangkut dengan *usability* system terhadap interaksi antara manusia dan system.

4. *Authorization*

Menjamin data diproses sesuai dengan ketentuan manajemen. *Authorization* menyangkut proses transaksi secara umum dan khusus.

Fokus Pengujian *Black box testing* yaitu sebagai berikut :

a) Menguji fungsi-fungsi khusus dari aplikasi.
b) *Test input* dan *output* untuk fungsi yang ada tanpa memperhatikan prosesnya.

Beberapa jenis kesalahan yang dapat di identifikasi :

1) Fungsi tidak benar atau hilang,
2) Kesalahan antar muka,
3) Kesalahan pada struktur data (pengaksesan basis data),

4) Kesalahan inisialisasi dan akhir program, dan

5) Kesalahan performasi.