BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Penelitian yang dilakukan di SMP Pandu Bandung bertujuan untuk membuat suatu sistem informasi. Sistem informasi yang akan dibuat adalah sistem informasi akademik berbasis web dengan maksud untuk mempermudah guru dan staf sekolah dalam mengelola data akademik.

3.1.1. Sejarah Singkat Perusahaan

Sekolah Menengah Pertama Pandu merupakan sekolah Katolik swasta yang telah berdiri sejak 1 Januari tahun 1967, yang berlokasi di Jalan Pandu No. 6 Bandung. Sekolah ini bernaung dibawah Yayasan Salib Suci. Yayasan Salib Suci berkedudukan di kota Bandung, yang beralamat di Jl. Van Deventer 18 Bandung. Saat ini dipimpin oleh seoran Pastor yang bernama Leo Van Bourden. Kegiatan Yayasan ini salah satunya adalah bagian pendidikan. Sekolah yang dimiliki Yayasan Salib Suci tersebar di beberapa daerah antara lain Bandung, Purwakarta, Karawang, Pamanukan, Cirebon, Subang, Tasik, Garut dan sebagainya. Yayasan ini memiliki 69 sekolah di wilayah Jawa Barat, di Bandung sendiri terdapat 8 Taman Kanak-Kanak, 8 Sekolah Dasar, 5 Sekolah Menengah Pertama, dan 4 Sekolah Menengah Atas. Saat ini SMP Pandu dipimpin oleh bapak Agustinus Tilarso Sudarmo, S.Pd.

3.1.2. Visi dan Misi Perusahaan

Visi dan misi yang dimiliki oleh SMP Pandu Bandung dalam menjalankan kegiatannya ialah:

a. Visi SMP Pandu

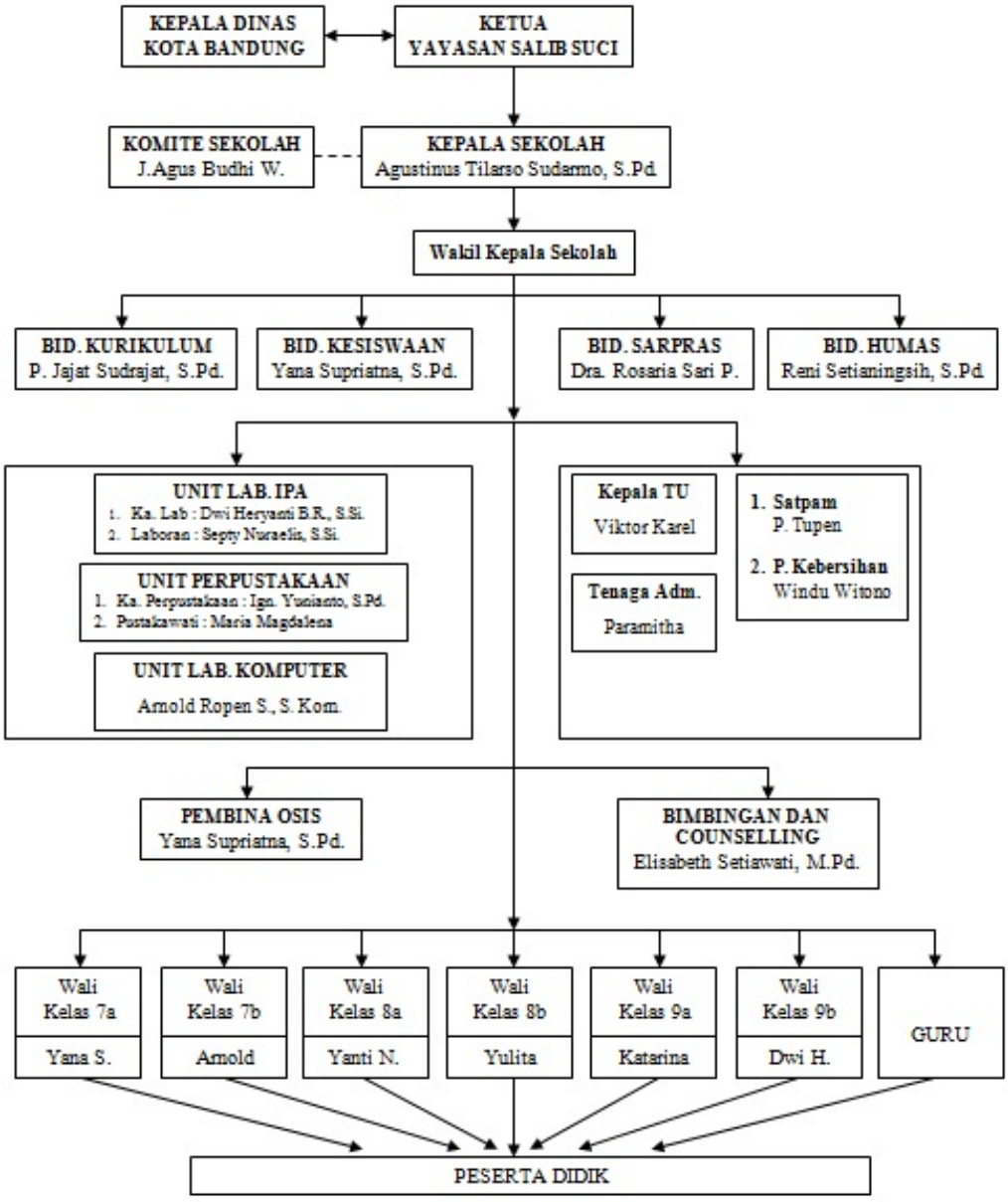
Menumbuhkan insan pembelajar yang "CEKATAN" (Cerdas, Empati, Kreatif, Aktif, Terampil, Antusias, dan memiliki hati Nurani) dan cinta lingkungan.

b. Misi SMP Pandu

1. Menciptakan suasana kondusif dalam kegiatan pembelajaran.
2. Memberikan pelayanan dengan hati yang tulus kepada siwa, orang tua tanpa membeda-bedakan.
3. Menumbuhkan rasa cinta dan kepedulian pada lingkungan.
4. Memberi kesempatan kepada siswa untuk dapat berkembang sesuai dengan potensinya.
5. Memfasilitasi siswa agar dapat mengembangkan seluruh aspek kepribadiannya sehingga siap untuk melanjutkan pendidikan ke jenjang berikutnya.

3.1.3. Struktur Organisasi Perusahaan

Adapun struktur organisasi yang ada pada SMP Pandu Bandung sebagai berikut :



Gambar 3.1. Struktur Organisasi SMP Pandu Tahun Pelajaran 2011-2012

(Sumber: Dokumen SMP Pandu Bandung)

3.1.4. Deskripsi Tugas

Berdasarkan struktur organisasi yang tertera, berikut ini akan diuraikan tugas, wewenang dan tanggung jawab dari masing-masing unit dalam struktur organisasi tersebut sebagai berikut:

1. Yayasan Salib Suci

   Bertugas mengontrol seluruh kegiatan di sekolah.

2. Kepala Sekolah

   Kepala Sekolah sebagai edukator, administrasi, manajer dan supervisi pendidikan, bertanggung jawab penuh atas penyelenggaraan pendidikan pada SMP Pandu Bandung.

3. Komite Sekolah

   Komite Sekolah sebagai pemberi pertimbangan dalam penentuan dan pelaksanaan kebijakan pendidikan, sebagai pendukung baik yang berwujud finansial, pemikiran, maupun tenaga dalam penyelenggaraan pendidikan, sebagai pengontrol dalam rangka transparansi dan akuntabilitas dalam penyelenggaraan pendidikan.

4. Wakil Kepala Sekolah Bidang Kurikulum

   Bertugas sebagai perencana dan mendesain semua program dan kegiatan pendidikan serta pembelajaran di sekolah.

5. Wakil Kepala Sekolah Bidang Kesiswaan

   a. Merencanakan dan melaksanakan penerimaan siswa baru setiap awal tahun pelajaran.

b. Melaksanakan pembinaan osis mencakup pemilihan pengurus osis, kegiatan osis, pembina pengurus osis.
c. Membina, melaksanakan dan mengkoordinasikan pelaksanaan kegiatan sekolah dengan bagian-bagian dalam lingkungan sekolah dan luar sekolah.
d. Menyusun tata tertib sekolah dan memonitor pelaksanaan tata tertib sekolah.

6. Wakil Kepala Sekolah Bidang Sarana Prasarana

   Memelihara dan mengamankan sarana atau prasarana termasuk mendistribusikan pengamanan, penghapusannya serta pengembangannya serta mengelola dan mengadakan koordinasi dalam hal pengadaan sarana atau prasarana dengan bagian atau bidang yang ada hubungannya dengan pendanaan atau keuangan.

7. Wakil Kepala Sekolah Bidang Humas
   a. Mengatur dan menyelengarakan hubungan sekolah dengan orang tua atau wali siswa.
   b. Membina hubungan antara sekolah dengan BP3.
   c. Membina pengembangan hubungan antara sekolah dengan masyarakat.
   d. Menyelenggarakan/mengkoordinasi pelaksanaan upacara hari Nasional baik disekolah maupun diluar sekolah dengan urusan kesiswaan.
   e. Aktif membantu wali kelas dan guru BK dalam menangani kasus-kasus masalah siswa diluar.

8. Unit Lab. IPA
   a. Bertanggung jawab terhadap pengelolaan laboratorium.

b. Menginventaris ulang terhadap alat dan perlengkapan di laboratorium.

c. Menyusun jadwal penggunaan alat oleh kelas/siswa.

d. Mengadakan perbaikan / pemeliharaan ruang laboratorium.

e. Menyusun perencanaan pengadaan / pembelian bahan.

9. Unit Lab. Komputer

a. Bertanggung jawab terhadap pengelolaan lab. komputer.

b. Menginventaris ulang terhadap perangkat di laboratorium.

c. Menyusun jadwal penggunaan lab. komputer oleh kelas/siswa.

d. Mengadakan perbaikan / pemeliharaan ruang laboratorium.

e. Menyusun perencanaan pengadaan / pembelian peralatan.

10. Unit Perpustakaan

a. Menyusun program pengelolaan perpustakaan.

b. Melaksanakan pelayanan perpustakaan.

c. Membuat rencana pengembangan perpustakaan.

d. Melaksanakan inventarisasi buku perpustakaan.

11. Kepala TU

a. Menyusun program kerja tata usaha.

b. Membantu kepala sekolah.

c. Pengurusan administrasi guru, pegawai, siswa.

12. Tenaga Admin

a. Membantu kepala TU.

b. Mencatat surat masuk dan surat keluar.

c. Melakukan kegiatan administrasi yang berhubungan dengan kegiatan belajar mengajar.

13. Bimbingan dan Konseling

a. Menyusun program dan pelaksanaan bimbingan penyuluhan.
b. Koordinasi dengan wali kelas dalam rangka mengatasi masalah yang dihadapi oleh siswa tentang kesulitan belajar.
c. Memberikan pelayanan bimbingan penyuluhan kepada siswa agar lebih berprestasi dalam kegiatan belajar.
d. Melaksakan koordinasi dalam urusan praktek dengan kepala sekolah, wali kelas dan guru dalam menilai siswa bila terjadi pelanggaran oleh siswa.

14. Wali Kelas

a. Menyusun organisasi kelas.
b. Membuat denah tempat duduk siswa.
c. Koordinasi dengan guru BP/BK dalam membuat peta siswa dikelasnya.
d. Membuat daftar inventaris kelas.
e. Mengisi buku kelas.
f. Mengisi raport siswa pada setiap akhir smester.
g. Mengontrol buku absen siswa.
h. Mengamati perkembangan kepribadian siswanya.
i. Membuat catatan khusus tentang siswa terutama bagi siswa yang mengalami kesulitan dan memerlukan bantuan/ penanganan.
j. Membagi laporan pendidikan (raport).

k. Melaporkan setiap permasalahan siswa dikelasnya kepada wakil kepala sekolah bidang kesiswaan untuk ditindak lanjuti.

15. Peserta Didik

Adapun siswa disini harus mengikuti segala kegiatan proses belajar mengajar, mematuhi tata tertib sekolah dan menjaga nama baik sekolah di luar sekolah.

3.2. Metode Penelitian

Pada subbab ini penulis akan menerangkan metode penelitian yang akan digunakan dalam penelitian ini. Metode penelitian ini meliputi desain penelitian, jenis dan metode pengumpulan data, serta metode pendekatan dan pengembangan sistem. Desain penelitian yang digunakan oleh penulis adalah Deskriptif dan *Action Research*. Sedangkan jenis pengumpulan data menggunakan data primer, dan data sekunder. Metode pendekatan sistem yang penulis pilih untuk perancangan aplikasi ini adalah pendekatan berorientasi objek dan metode pengembangan sistem yang digunakan adalah *Rational Unified Process (RUP)*.

3.2.1. Desain Penelitian

Penelitian ini menggunakan metode Deskriptif dan *Action Research*. Penelitian deskriptif dapat diartikan sebagai proses pemecahan masalah yang diselidiki dengan melukiskan keadaan subyek dan obyek pada saat sekarang berdasarkan fakta-fakta yang ada. Dimana dalam metode ini menggambarkan semua data yang kemudian dianalisis dan dibandingkan berdasarkan kenyataan

yang sedang berlangsung dan selanjutnya mencoba untuk memberikan pemecahan masalahnya.

Pelaksanaan metode penelitian deskriptif tidak terbatas sampai pada pengumpulan dan penyusunan data, tetapi meliputi analisis dan interpretasi tentang data tersebut, selain itu semua yang dikumpulkan memungkinkan menjadi kunci terhadap apa yang diteliti.

Penelitian ini menggunakan metode *Action Research* (penelitian tindakan). Metode *Action Research* 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 . 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 hanya dapat kita peroleh dari sumber asli (nara sumber) / 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 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 SMP Pandu 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.

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.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Dalam merancang suatu sistem dibutuhkan sebuah metode pendekatan dan pengembangan sistem untuk membantu dalam proses perancangan dan pembuatannya. Berikut adalah penjelasan metode pendekatan dan pengembangan sistem yang akan digunakan.

3.2.3.1. Metode Pendekatan Sistem

Pada dasarnya saat ini pengembangan sistem dapat dikategorikan dalam 2 pendekatan pengembangan, yaitu pengembangan secara terstruktur daan pengembangan secara *object oriented* (berorientasi objek). Metode pendekatan yang penulis ambil adalah pendekatan berorientasi objek yang divisualisasikan dengan UML.

Dalam jurnal yang ditulis oleh Hendrik, Fathul Wahid yang berjudul "Pengembangan Learning Management System" menerangkan bahwa :

"Metode berorientasi obyek (*object oriented method*) merupakan suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan abstraksi obyek-obyek yang ada di dunia nyata. Brooks (1987) – dalam Nugroho (2002) – menyatakan bahwa bagian tersulit dari pengembangan perangkat lunak dan / atau sistem informasi adalah tahap analisis, yang didalamnya abstraksi dilakukan. Dalam metode berorientasi obyek kejadian atau hubungan antar entitas dalam dunia nyata direpresentasikan dalam obyek-obyek. "

3.2.3.2. Metode Pengembangan Sistem

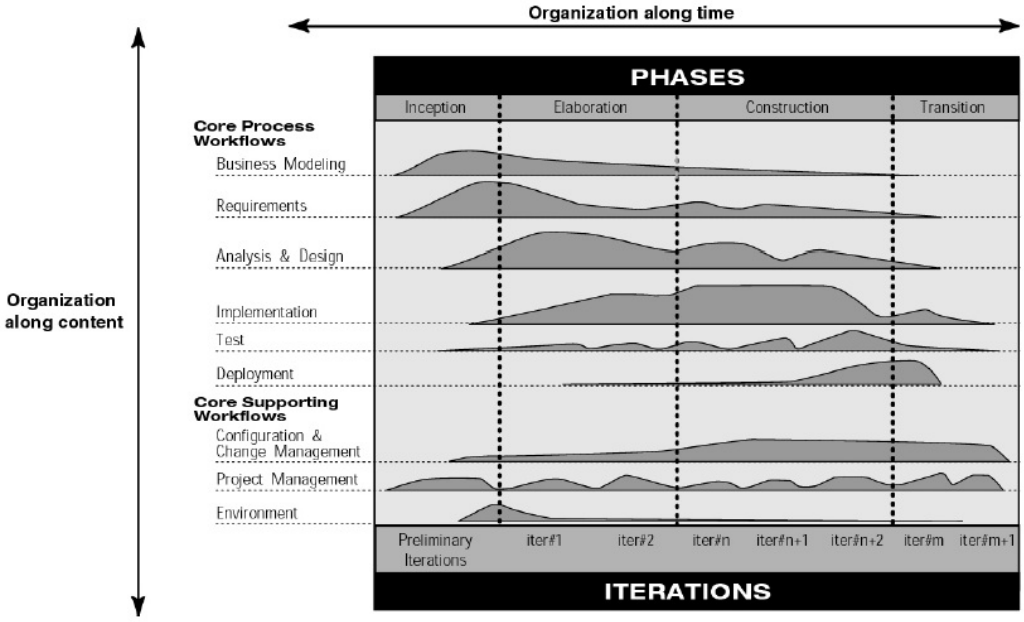
Metode pengembangan yang penulis pakai adalah metode *Rational Unified Process* (RUP). *Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

1. **Dimensi pertama** digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase

selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception, Elaboration, Construction,* dan *Transition.*

2. **Dimensi kedua** digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing, what, how* dan *when*. Dimensi ini terdiri atas *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration* dan *Change Manegement, Project Management, Environtment.*



Gambar 3.2. Arsitektur *Rational Unified Process* (RUP)

(Sumber : *Rational-The Software Development Company, Rational Unified Process: Best Practices for Software Development Teams*)

3.2.3.3. Alat Bantu Analisis dan Perancangan

Pada perancangan sistem informasi akademik ini, penulis menggunakan metode pendekatan berorientasi objek, terdapat banyak alat bantu yang dapat digunakan dalam analisis dan desain berorientasi obyek menyebabkan munculnya ketidakjelasan alat bantu apa yang paling unggul. Para pengguna harus memilih diantara bahasa pemodelan dan alat bantu pemodelan yang beragam. Untuk mengatasi masalah tersebut maka *Object Management Group* (OMG) kemudian mengeluarkan UML, dimana dengan adanya UML ini diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkugan industri perangkat lunak.

1. *Use Case Diagram*

Diagram *use case* menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar.

Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirements* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case* diagram menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use case* diagram.

Dalam diagram *use-case* terdapat beberapa hubungan antar *use-case* antara lain: *Extend* dan *Include*. Hubungan *include* menunjukkan bahwa sebuah *business use-case* selalu termasuk dalam alur kerja dari *business use-case* lainnya.

Hubungan *extend* menunjukkan bahwa sebuah *use-case* merupakan kelanjutan dari *business use-case* lainnya jika suatu syarat tertentu telah dipenuhi.

2. *Activity Diagram*

Activity diagram (diagram aktivitas) adalah diagram *flowchart* yang disempurnakan. Diagram aktivitas menggambarkan operasi pada suatu obyek atau proses pada sebuah organisasi. Kelebihan diagram aktivitas dibandingkan dengan diagram *flowchart* adalah adanya dukungan konkurensi (pelaksanaan aktivitas secara bersamaan), pengiriman pesan dan *swimlane* (pelaku/penanggung jawab aktivitas).

Diagram aktivitas sangat berguna ketika kita ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi. Diagram aktivitas diawali dengan lingkaran hitam, dan diakhiri dengan lingkaran hitam bertepi putih. Aktivitas digambarkan dengan kotak persegi panjang bersudut lengkung. Setiap aktivitas dihubungkan dengan panah dari awal hingga akhir diagram aktivitas.

Sama halnya dengan diagram *flowchart*, diagram aktivitas pun memiliki simbol yang sama untuk menggambarkan keputusan. Keputusan digambarkan dengan intan, namun deskripsi kondisi yang menyertai keputusan diletakkan diluar simbol intan. Diagram aktivitas dapat menggambarkan konkurensi, yaitu satu atau lebih aktivitas yang berjalan secara bersamaan. Konkurensi diawali dengan sebuah garis tebal horizontal yang menjadi tempat keluarnya garis aktivitas. Konkurensi juga diakhiri dengan garis tebal horizontal.

3. *Sequence Diagram*

Sequence diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sebuah *Sequence diagram*, secara khusus, menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sebuah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *use case*.

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

4. *Class Diagram*

Diagram *class* menggambarkan seluruh *class* yang terlibat dalam sebuah perangkat lunak. Diagram ini menggambarkan seluruh atribut dan operasi suatu *class* beserta hubungannya dengan *class* lainnya.

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.

5. *Object Diagram*

Object diagrams memodelkan pandangan statik terhadap sistem untuk memodelkan struktur objek.

Menurut Bambang Heriyanto (2004:291) *Object Diagram* adalah diagram instan (*instance diagram*) yang mendeskripsikan instan-instan kelas.

Pemodelan struktur objek melibatkan *snapshot* dari objek-objek sistem pada suatu waktu. *Object diagrams* merepresentasikan satu *frame* statik papan cerita dinamis dari diagram interaksi.

Kegunaan *object diagram* adalah mendeskripsikan bagaimana kumpulan objek tertentu saling berhubungan. *Object diagrams* adalah diagram instan yang mendeskripsikan instan-instan kelas. Diagram instan berguna untuk dokumentasi skenario, serta kasus pengujian dan mendiskusikan contoh-contoh instanisasi *class diagrams*.

6. *Component Diagram*

Diagram komponen menggambarkan alokasi semua kelas dan obyek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *soure code, binary code* dan komponen tereksekusi.

7. *Deployment Diagram*

Diagram *deployment* menggambarkan bagaimana konfigurasi perangkat keras dimana sistem akan diinstalasikan. Diagram ini memberi simbol untuk segala bentuk perangkat keras, seperti: komputer, printer, router, hub, jaringan, dan sebagainya.

Diagram *deployment* menunjukkan tata-letak perangkat keras secara fisik, dan komponen yang terdapat didalamnya. Diagram ini menunjukkan secara lengkap arsitektur dari sistem dan perangkat kerasnya. Diagram ini akan memperjelas pengguna bagaimana hasil akhir dari sistem yang dikembangkan, dan mempermudah teknisi dalam pemeliharaan sistem.

Deployment diagram 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.

3.3. Pengujian Software

Penulis akan menggunakan teknik *black box testing* dalam pengujian *software*. 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. Pengujian *black box* merupakan metode perancang data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dijalankan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

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.

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
5. Kesalahan performasi.