BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Objek penelitian ini dilakukan di lembaga pendidikan atau sekolah dimana meliputi sejarah sekolah, struktur organisasi, visi dan misi sekolah serta deskripsi kerja yang ada di lingkungan sekolah SMP Negeri 2 Bandung yang akan dijelaskan di bawah ini.

3.1.1. Sejarah Singkat SMP Negeri 2 Bandung

Sejarah SMP Negeri 2 Bandung dimulai sekitar tahun 1948 dengan nama *Indofuroese Vereneging Kubbk School* dan berlokasi di Jalan Kalipah Apo, kemudian berturut-turut pindah ke Jalan Kasatriaan, Jalan Papandayan, dan Jalan Babatan. Pemidahan terakhir SMP Negeri 2 Bandung adalah ke lokasi saat ini, yaitu di Jalan Sumatera, dan pada tahun 1950 berubah nama menjadi SMP Negeri 2 Bandung.

SMP Negeri 2 Bandung menempati bangunan bekas sekolah Belanda, yaitu *Urfese Larage School* yang dibangun pada tahun 1913 oleh Pemerintah Hindia Belanda dan diambil alih oleh pemerintah Indonesia hingga sekarang. Posisinya kira-kira 400 meter di sebelah selatan Kantor Wali Kota Bandung.

3.1.2. Visi dan Misi SMP Negeri 2 Bandung

1. Visi SMP Negeri 2 Bandung

"Sekolah dengan karakter dan prestasi di tingkat nasional / internasional"

2. Misi SMP Negeri 2 Bandung

1). Menjalani kehidupan berdasarkan persyaratan dalam agama dan hukum

2). Jujur, bertanggung jawab, disiplin, pekerja keras dan mandiri

3). Menjaga lingkungan, pemeliharaan hidup sehat, bersih dan nyaman

4). Menciptakan suasana harmoni, demokratis dan patriotik

5). Mengembangkan logika cara berpikir, kritis, kreatif dan inovatif

6). Meningkatkan profesionalisme pendidik dan tenaga kependidikan melalui seminar, lokakarya pelatihan, dan pembelajaran mandiri

7). Melakukan proses belajar yang aktif, kreatif, efektif dan menyenangkan dengan pengajaran yang berkualitas

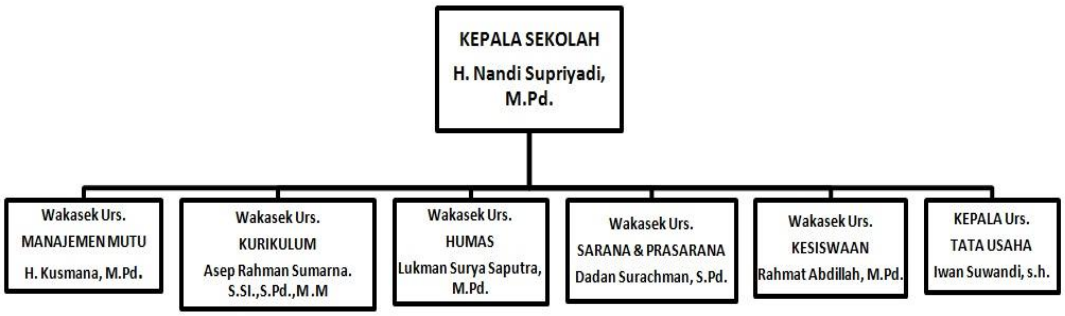
8). Memperkenalkan budaya lokal ke tingkat nasional dan internasional

9). Berpartisipasi dan berprestasi dalam kegiatan seni, budaya dan akademik di tingkat nasional / internasional

10). Membangun kerjasama antar sekolah di Asia, Eropa dan daerah lainnya

3.1.3. Struktur Organisasi SMP Negeri 2 Bandung

Struktur Organisasi SMP Negeri 2 Bandung



Gambar 3.1 Struktur Organisasi SMP Negeri 2 Bandung

Sumber : (*Website* SMP Negeri 2 Bandung)

3.1.4. Deskripsi Tugas

1. Kepala Sekolah

Sesuai dengan fungsi kepala sekolah sebagai administrator, edukator, manajer, dan *supervisor*, maka tugas kepala sekolah ialah:

a. Sebagai penanggung jawab pelaksanaan pendidikan disekolah termasuk dalam pelaksanaan administrasi sekolah.
b. Kepala sekolah mempunyai tugas merencanakan, mengorganisir, mengawasi dan mengevaluasi proses pendidikan di sekolah yang meliputi aspek edukatif dan administratif. Hubungan dengan pelaksanaan kurikulum sedangkan administratif meliputi pengaturan administrasi

belajar mengajar, siswa, keuangan, perlengkapan dan hubungan masyarakat.

2. Wakil Kepala Sekolah urusan Manajemen Mutu

Wakil kepala sekolah urusan manajemen mutu bertugas membantu kepala sekolah dalam :

a. Mengelola informasi hasil evaluasi diri, evaluasi kinerja RSBI, hasil supervisi pengawas dan akreditasi sebagai dasar perbaikan mutu berkelanjutan
b. Menginventarisasi target kinerja pada tiap bidang kegiatan sebagai dasar penentuan target kinerja penjaminan mutu
c. Menentukan indikator mutu yang menjadi sasaran evaluasi kinerja
d. Mengolah data hasil evaluasi dan penjaminan mutu
e. Menyusun rekomendasi perbaikan mutu

3. Wakil Kepala Sekolah urusan Kurikulum

Wakil kepala sekolah urusan kurikulum bertugas membantu kepala sekolah dalam :

a. Menyusun program pengajaran/jadwal mengajar
b. Pembagian tugas mengajar
c. Menyusun program evaluasi belajar, semester dan akhir tahun pelajaran
d. Mengorganisasi/mengkoordinasi pelaksanaan pembagian *raport*, STTB dan lain-lain
e. Mengkoordinasi penyusunan program satuan pelajaran

f. Menyusun laporan pelaksanaan pengajaran secara berkala

6. Wakil kepala sekolah urusan humas

Wakil kepala sekolah urusan humas bertugas membantu kepala sekolah dalam:

a. Mengatur dan menyelenggarakan hubungan sekolah dengan orang tua / wali siswa
b. Membina pengembangan hubungan antar sekolah dengan lembaga-lembaga pemerintah, dunia usaha, dunia industri, dan lembaga sosial lainnya
c. Menyusun laporan pelaksanaan hubungan masyarakat secara berkala

5. Wakil kepala sekolah urusan sarana dan prasarana

Wakil kepala sekolah urusan sarana dan prasarana bertugas membantu kepala sekolah dalam:

a. Melaksanakan urusan inventaris barang sekolah
b. Merencanakan pendayagunaan sarana/prasarana sekolah
c. Melaksanakan penyusunan laporan keadaan bangunan sekolah, meliputi kerusakan maupun pengembangan

6. Wakil kepala sekolah urusan kesiswaan

Wakil kepala sekolah urusan kesiswaan bertugas membantu kepala sekolah dalam:

a. Menyusun program pembinaan kesiswaan

b. Melaksanakan bimbingan, pengarahan, dan pengendalian kegiatan siswa dalam rangka peningkatan kedisiplinan dan tata tertib sekolah
c. Membina dan melaksanakan koordinasi keamanan, kebersihan, ketertiban, keindahan, kekeluargaan dan perindangan
d. Memberikan pengarahan dalam pemilihan pengurus OSIS
e. Memberikan kepengurusan OSIS

7. Tata Usaha

Adapun tugas daru tata usaha adalah:

a. Menyusun program kerja tata usaha
b. Membantu kepala sekolah
c. Pengurusan administrasi guru, pegawai, 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 objek 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.

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.

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.

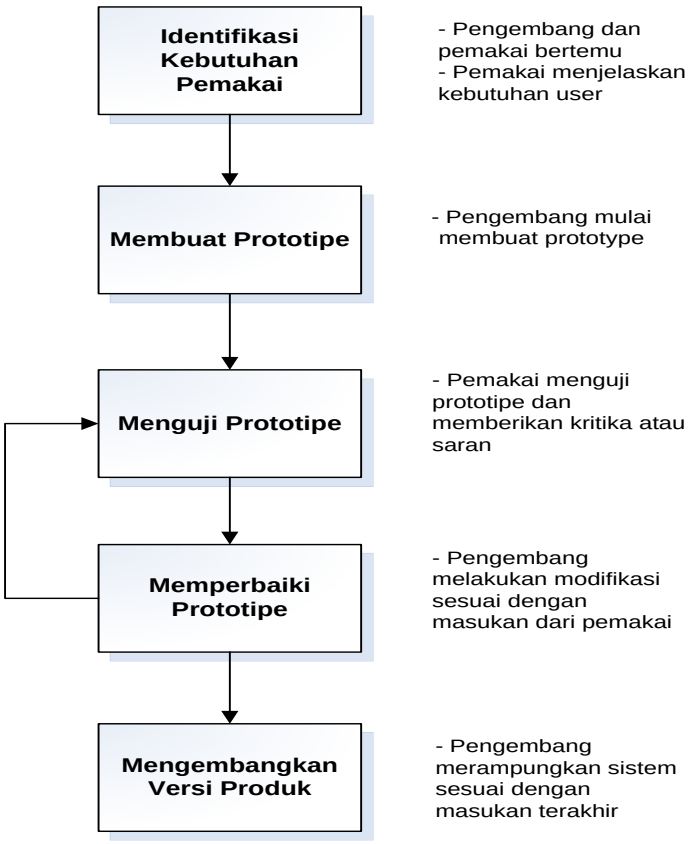
Metode pendekatan sistem yang di gunakan adalah pendekatan dengan *Object Oriented* dengan menggunakan alat bantu UML, adapun diagram yang digunakan adalah sebagai berikut *: Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, Component Diagram* dan *Deployment Diagram*.

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 *prototyping* 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) 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.

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 informasi 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 ketiga yaitu waktu menjadi aktor ketika ia memicu *event-event* tertentu bagi sistem/perangkat lunak yang kita kembangkan.

2) 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. Logika dari proses bisnis.

3) 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.

4) *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 atribut 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* kepada *class* lain.

5) Component Diagrams

Menurut Bambang Heriyanto (2004:293) Komponen adalah bagian fisik dan dapat diganti dengan 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.

6) 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

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 bisa 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.