BAB III
ANALISA SISTEM

3.1 Objek Penelitian

3.1.1 Sejarah Singkat PT. Krakatau Industrial Estate Cilegon (KIEC)

PT. Krakatau Industrial Estate Cilegon (KIEC) adalah anak perusahaan dari PT. Krakatau Steel yang berdiri sejak tanggal 16 Juni 1982. Pada awalnya PT. KIEC hanya bergerak di bidang kawasan industri berupa pengurusan jasa sewa dan jual lahan industri untuk industri hilir produk baja PT. Krakatau Steel saja, tetapi pada tahun 1995 PT. KIEC memperoleh tanggung jawab dari PT. Krakatau Steel untuk mengelola lapangan Golf, Sarana Olah Raga (SOR), hotel dan Perkantoran. Saat ini PT. KIEC telah bergerak di bidang kawasan industri, kawasan komersial serta kawasan hunian.

Bidang usaha atau bisnis PT. KIEC meliputi :

1) Properti Industri
2) Kawasan Industri, Standar Factory Building (SFB), dan Gudang
3) Hotel, Perkantoran, Golf, dan SOR
4) Properti Residen
5) Real Estate
6) Investasi dan Perdagangan
7) Depo Peti Kemas dan Konblok

Kegiatan PT. Krakatau Industrial Estate Cilegon (KIEC) selaku perusahaan kawasan industri adalah :

1). Membangun kawasan industri dengan mempersiapkan dan menyediakan lahan industri siap bangun yang telah di lengkapi dengan sarana dan prasarana penunjang.
2). Melaksanakan kegiatan perawatan atas prasarana, sarana serta fasilitas lainnya yang di sediakan.
3). Memberikan bantuan dan mengurus permintaan perizinan bagi perusahaan industri yang ada di dalam kawasan industri.
4). Memberikan pelayanan kepada para penanam modal dalam bentuk pengamanan kawasan industri.
5). Mengarahkan pelaksanaan pembangunan di dalam kawasan industri sesuai dengan ketentuan peraturan perundangan.
6). Membantu pemerintah daerah dalam memantau kegiatan industri yang ada di dalam kawasan industri.

PT. Krakatau Industrial Estate Cilegon (KIEC) bekerja sama dengan anak perusahaan lain seperti PT. CIP, PT. Tobu Steel, PT. Prok Indonesia dan yang lainnya.

3.1.2 Visi dan Misi PT. Krakatau Industrial Estate Cilegon

visi

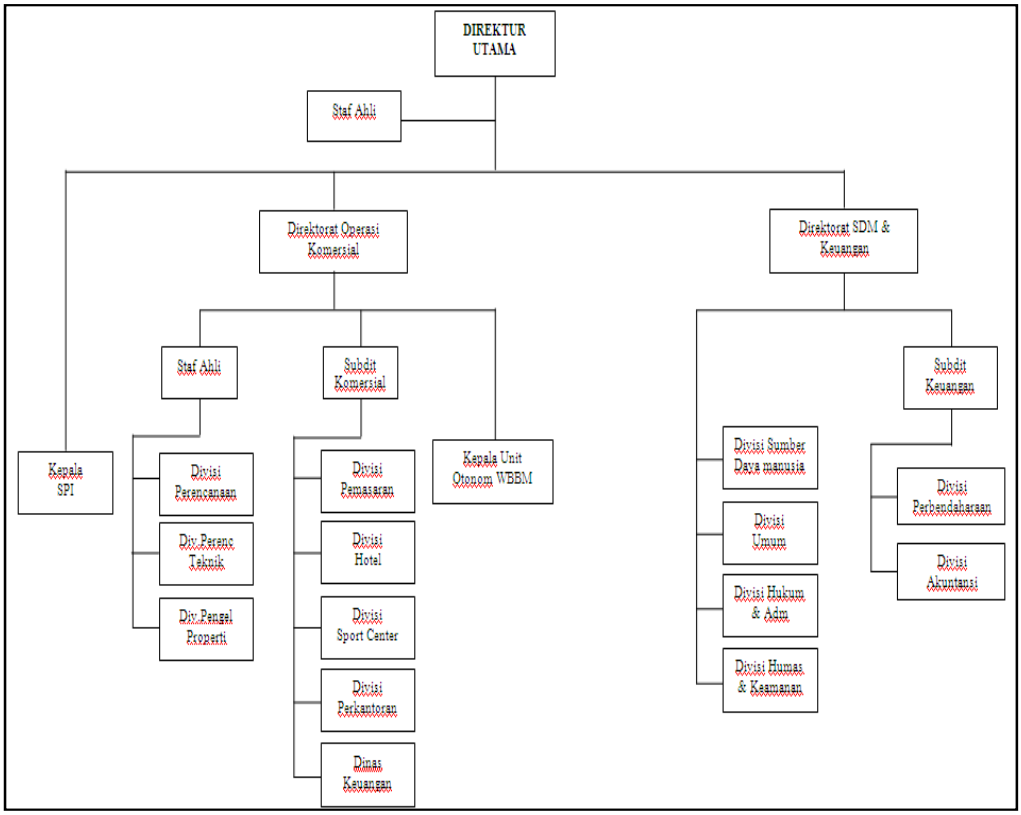
Pemain Properti Nasional yang Terkemuka

Dan dengan misi :

"Menyediakan Properti Industri, Komersial, Hunian
dan Infrastruktur terkait
yang memberikan Solusi bagi Investor, Pelanggan dan Pihak-Pihak terkait
lainnya "

3.1.3 Struktur Organisasi

Berdasarkan Rapat Dewan Direksi struktur orgaanisasi NO.SK.01/0804/2003 pada PT. Krakatau Industrial Estate Cilegon sebagai b



Gambar 3.1 Srtuktur Organisasi PT. KIEC

3.1.4.Tugas dan Wewenang

1) **Direktur Utama**

   Mempunyai tugas dan wewenang mengelola dan merumuskan kebijakan perusahaan secara menyeluruh dan terintegrasi sesuai dengan kebijakan umum yang di gariskan oleh Rapat umum pemegang. Serta menetapkan kebijakan dan sasaran rencana jangka panjang perusahaan, sasaran jangka menengah dan Rencana Kerja Anggaran Perusahaan (RKAP) untuk tercapainya misi dan sasaran perusahaan sehingga memperoleh laba dalam arti yang luas.

2) **Staf Ahli**

   Mempunyai tugas dan wewenang mengkoordinasikan, menganalisa dan merekomendasikan berbagai kebijakan mengenai administrasi keuangan, SDM dan Logistik agar di peroleh aktivitas operasional dan tujuan perusahaan yang terkendali dan berkembang.

3) **Direktorat Operasi dan Komersil**

   Mempunyai tugas dan wewenang merencanakan, merumuskan, mengendalikan dan mengembangkan kebijakan-kebijakan strategis perusahaan dalam bidang pengembangan bisnis, perencanaan strategis, pengelolaan properti industri, perencanaan, pengawasan dan perawatan pembangunan, pengelolaan hotel, Sport center, pengelolaan perkantoran dan pergudangan, pengelolaan unit otonom WBBM untuk mencapai visi dan misi perusahaan.

4) Direktorat Operasi dan Komersial

Mempunyai tugas dan wewenang merencanakan dan mengandalikan kegiatan penyusunan rencana jangka panjang dan jangka menengah perusahaan, pengembangan bisnis perusahaan, perencanaan tata ruang dan prasarana, pengelolaan properti industri, perawatan kawasan dan pengawasan pembangunan seluruh unit bisnis perusahaan dalam mendukung sasaran jangka panjang dan jangka menengah.

a) Subdit Operasi

Mempunyai tugas dan wewenang merencanakan dan mengendalikan kegiatan penyusunan rencana jangka panjang dan jangka menengah perusahaan, pengembangan bisnis perusahaan, perencanaan tata ruang dan prasarana, pengelolaan properti industri, perawatan kawasan dan pengawasan pembangunan seluruh unit bisnis perusahaan dalam mendukung sasaran jangka panjang dan jangka menengah perusahaan.

(1) Divisi Perencanaan & Pengembangan Bisnis

Mempunyai tugas dan wewenang mengorganisasikan dan mengkoordinasikan kegiatan perencanaan strategis perusahaan dengan mempersiapkan Rencana Jangka Panjang (RJP), Rencana Jangka Menengah (RJM) dan merumuskan protofolio bisnis, melakukan kaji ulang kelayakan investasi, kajian rencana pengembangan bisnis perusahaan dan rekomendasi-rekomendasi strategis.

(2) Divisi Pengelolaan Properti Industri

Mempunyai tugas dan wewenang mengorganisasikan dan mengkoordinasikan kegiatan pengendalian lingkungan kawasan industri, pelayanan kepada investor, perawatan properti industri dan pengawasan pembangunannya guna meningkatkan pelayanan terhadap pelanggan/investor.

b) Subdit Komersil

Mempunyai tugas dan wewenang merencanakan dan mengendalikan kegiatan pemasaran kawasan dan properti, pengelolaan hotel, pengelolaan sport center, pengelolaan perkantoran dan pergudangan serta keuangan hotel dan sport center sesuai dengan sasaran perusahaan.

(1) Divisi Pemasaran

Mempunyai tugas dan wewenang mengorganisasikan dan mengkoordinasikan kegiatan-kegiatan pemasaran dan penjualan properti industri (Lahan industri, SFB, Pergudangan), riset dan pengembangan pasar dan strategi untuk pemasaran industri.

(2) Divisi Hotel

Mempunyai tugas dan wewenang mengorganisasikan dan mengkoordinasikan kegiatan penjualan jasa hotel, operasional hotel meliputi room, food & beverage, dan perawatannya untuk mencapai target sales yang telah di tetapkan.

(3) Divisi Perkantoran & Pergudangan

Mempunyai tugas dan wewenang mengorganisasikan dan mengkoordinasikan kegiatan penjualan dan pengelolaan jasa perkantoran, gudang SFB dan depo secara efisien dan efektif.

c) Kepala Unit Otonom WBBM

Mempunyai tugas dan wewenang membantu kepala Direktorat Operasi & Komersial dalam mengelola dan mengembangkan unit usaha WBBM yang meliputi : pengelolaan personil, pengelolaan keuangan, pengoperasian serta pemeliharaan sarana dan prasarana gedung wisma baja sesuai dengan kebijakan perusahaan dengan memperhatikan efektifitas dan efisiensi biaya.

5) Direktorat SDM dan Keuangan

Mempunyai tugas dan wewenang merencanakan, merumuskan dan menggambarkan kebijakan keuangan baik fungsi perbendaharaan maupun akuntansi, SDM, hukum & administrasi, humas & keamanan, logistik agar tercipta kelancaran serta kondisi yang sehat sehingga mampu mendukung kegiatan operasional dan tujuan perusahaan.

a) Sub Direktorat Keuangan

Mempunyai tugas dan wewenang merencanakan, mengkoordinasikan, menyelenggarakan dan mengendalikan fungsi perbendaharaan dan akuntansi serta perpajakan untuk mendukung kegiatan operasional perusahaan.

b) Divisi SDM

Mempunyai tugas dan wewenang mengatur, mengorganisasikan, mengkoordinasikan dan mengendalikan perencanaan, penyusunan, pengembangan, pengelolaan karier, evaluasi kinerja, kompensasi dan hubungan sumber daya manusia sesuai dengan kebijakan perusahaan.

c) Divisi Perbendaharaan

Mempunyai tugas dan wewenang mengatur, mengorganisasikan, mengkoordinasikan, dan mengendalikan fungsi Dinas Pendanaan dan Penagihan serta Dinas Pajak Asuransi dan Faktur tagihan sedemikian rupa sehingga sumber dana dapat di kelola secara efektif dan efisien untuk dapat menjamin tersedianya dana pada saat diperlukan oleh perusahaan dan kewajiban perpajakan dapat di penuhi tepat pada waktunya.

d) Divisi Akuntansi

Mempunyai tugas dan wewenang mengorganisasikan dan mengkoordinasikan kegiatan akuntansi, budgeting, sistem informasi akuntansi perusahaan, sistem pengendalian anak perusahaan dan perusahaan patungan yang efisien dan efektif agar terselenggaranya sistem pengendalian intern dan mencapai kegiatan dan ketepatan penyajian informasi akuntansi/keuangan kepada semua pihak yang berkepentingan.

e) Divisi Keamanan

Mempunyai tugas dan wewenang menyelenggarakan dan mengatur kegiatan pengamanan terhadap harta kekayaan perusahaan maupun personil dengan mengkoordinir penjagaan, patroli, penyelidikan/perizinan agar terjamin keamanan dan ketertiban di seluruh kawasan dan K3 (Kesehatan dan Keselamatan Kerja) sehingga terjamin ketenangan berusaha dan menumbuhkan kepercayaan konsumen dalam menggunakan produk perusahaan.

3.2. Metode Penelitian

3.2.1. Desain Penelitian

Dalam penelitian ini, peneliti menggunakan rancangan penelitian berdasarkan metode deskriptip yaitu suatu metode dalam meneliti status kelompok manusia, suatu obyek, suatu set kondisi,suatu system pemikiran ataupun suatu kelas peristiwa pada masa sekarang.

3.2.2. Jenis dan Metode Pengumpulan Data

3.2.2.1. Jenis Data

Dalam penelitian ini, terdapat dua jenis data yang digunakan yaitu sebagai berikut :

1. Sumber Data Primer

   Data primer adalah data yang diperoleh langsung dari unit pengamatan atau responden penelitian.

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.

3.2.2.2. Metode Pengumpulan Data

1. Sumber Data Primer

   Dalam penelitian ini penulis mengumpulkan data primer dengan menggunakan teknik. Metode pengumpulan data yang di gunakan adalah :

   a. Penelitian Kepustakaan (Library Research)

      Yaitu penelitian dengan membaca literature yang berhubungan dengan masalah yang akan di bahas.

   b. Penelitian Lapangan (Field Research)

      Yaitu penelitian langsung ke perusahaan dan instansi yang bersangkutan, teknik yang di lakukan yaitu dengan cara :

      a). Wawancara

         Data diperoleh dengan melakukan wawancara langsung dengan cara tanya jawab dengan pihak-pihak yang di anggap atau di harapkan dapat mengetahui permasalahan yang terjadi di PT Krakatau industrial Estate Cilegon Khususnya Pada bagian SDM, yang menagani tunjangan kesehatan.

b). Observasi Lapangan

Dengan teknis ini, maka penulis mengharapkan dapat mengetahui keadaan sistem yang sedang berjalan saat ini, sehingga dapat ditarik kesimpulan mengenai permasalahan yang timbul atau kebutuhan-kebutuhan apa saja yang dapat dipenuhi dalam pengembangan sistem yang dilakukan.

c). Metode Pengamatan

Dengan cara melakukan pengamatan langsung terhadap sistem yang kerja secara langsung terhadap objek yang sedang diteliti.

2. Sumber Data Sekunder

Adapun data yang berasal dari sumber data sekunder diperoleh dengan teknik dokumentasi. Dokumentasi adalah teknik pengumpulan data dengan cara mengumpulkan dokumen-dokumen yang berhubungan dengan obyek penelitian. Dalam hal ini, dokumen-dokumen yang diperoleh dianalisis sehingga diperoleh data-data yang sesuai untuk kegiatan pengembangan sistem.

3.2.3. Metode Pendekatan dan Pengembangan

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 *metode Prototype.*

3.2.3.1. Metode Pendekatan

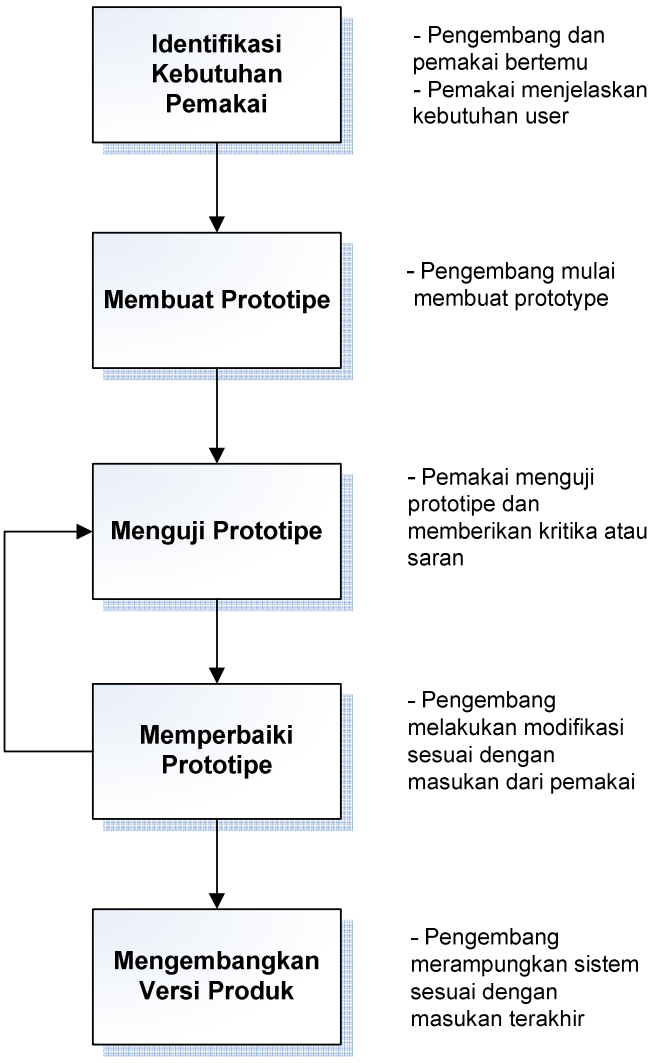
Metode pendekatan sistem yang di gunakan adalah pendekatan dengan *Object Oriented* yang di visualisasikan dengan UML dan di antara nya adalah sebagai berikut *: Use Case, , Sequence Diagram, Class Diagram, Collaboration Diagram, Component Diagram* dan *Deployment Diagram.*

3.2.3.2. Metode Pengembangan

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 Prototipe

(Sumber : Abdul Kadir, Pengenalan Sistem Informasi 2003, Andi: Yogjakarta)

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) *Actor*

Dalam pemodelan sistem dengan UML, actor adalah seseorang atau sesuatu yang berinteraksi dengan system yang sedang kita kembangkan. Aktor berada di luar lingkupan sistem/perangkat lunak yang sedang kita kembangkan dan bersifat eksternal.

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 actor yang paling umum dan hadir di setiap system/perangkat lunak.
2. Aktor yang kedua yaitu sistem lain.
3. Aktor yang ke tiga yaitu waktu mengadi actor ketika ia memicu event-event tertentu bagi system/perangkat lunak yang kita kembangkan.

2) *Use-case Diagrams*

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

Menurut Martin Fowler (2005 : 141), Use Case *adalah "teknik untuk merekam persyaratan fungsional sebuah sistem"*.

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* kepada *class* lain.

4) *Object Diagrams*

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

Menurut Martin Fowler (2005 : 125), Object Diagram merupakan "sebuah gambaran tentang objek-objek dalam sebuah sistem pada satu titik waktu. Karena lebih menonjolkan perintah-perintah dari pada class, object diagram lebih sering disebut sebagai sebuah diagram perintah".

Pemodelan struktur objek melibatkan *snapshot* dari objek-objek sistem pada suatu waktu. *Object diagrams* merepresentasikan satu frame statik papn 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*.

5) *Component Diagrams*

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

7) *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.

8) *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.

9) *Statechart Diagrams*

Statechart merupakan perluasan *State diagrams*. *Statechart* diperkenalkan oleh Harel, kemudian dipopularkan oleh Rumbaugh dan kawan-kawan di OMT. *Statechart* mempermudah pemodelan perilaku sistem secara tidak ambigu.

Statechart menyediakan notasi dan sekumpulan konvensi fasilitas dekomposisi hirarki mesin-mesin *finite state* dan mekanisme berkomunikasi di antara mesin-mesin *finite state* konkuren.

Satu *state diagrams* untuk satu kelas objek dimana perilaku dinamisnya penting dan menunjukkan pola aktivitas. Masing-masing mesin *state* berjalan secara konkuren dan dapat mengubah *state* secara independen. Diagram-diagram *state* beragam kelas membentuk satu

model dinamis tunggal lewat kejadian bersama (*shared events*). *Statechart* mendeskripsikan objek berupa state-state yang dimilikinya, kejadian-kejadian yang dapat berlangsung beserta transisi.

10) *Activity Diagrams*

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.

3.4 Pengujian *Software Black Box Testing*

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 ditujukan untuk menghasilkan perangkat lunak *(software)* yang bebas kesalahan, paling tidak secara teknik.

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*

   Menekankan pada 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.