BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Penelitian yang dilakukan di Purwacaraka Music Studio bertujuan untuk merancang suatu aplikasi yang dapat membantu dan mempermudah kegiatan belajar mengajar pada murid dengan membangun aplikasi mobile piano, gitar dan drum virtual berbasis *platform mobile* pada android.

3.1.1 Sejarah Purwacaraka Music Stuido

Purwacaraka Music Studio (PCMS) lahir dari tangan pemusik kenamaan yaitu Purwacaraka. Kesadaran akan pentingnya musik tersebut merupakan sesuatu yang mendorong Purwacaraka untuk terus mengajarkan musik kepada masyarakat Indonesia. Sejak pertama berdiri 22 tahun yang lalu, tepatnya tahun 1988 di Bandung, sebagai lembaga pendidikan musik terbesar di Indonesia yang memiliki standarisasi secara nasional dan manajemen terstruktur. Hingga saat ini purwacaraka Musik Studio telah tersebar di 31 kota.

Mengenai instrumen musik yang dipelajari oleh siswa pada Program Purwacaraka Music Studio adalah sebagai berikut : vocal, biola, drum, keyboard, piano klasik, piano pop, gitar klasik, gitar elektrik dan gitar bass.

Tujuan yang ingin dicapai dengan didirikannya PCMS ini adalah:

1. Memberikan pendidikan musik kepada khalayak umum karena musik tanpa disadari sudah menjadi bagian penting dalam hidup manusia dan akan tetap ada sepanjang masa.

2. Menyediakan pelayanan dan pengembangan bakat yang luas untuk pendidikan calon musisi, subspesialis dan entertainer yang berkualitas di bidang musik.
3. Menyelenggarakan kegiatan performance di depan umum sebagai ajang unjuk diri sekaligus sebagai result dari hasil belajar di PCMS serta melatih mental anak didik terhadap khalayak umum dimana tren musik sedang berkembang yang antara lain stasiun TV juga turut menampilkan acara lomba musik, yang bersifat kompetitif.

3.1.2 Visi dan Misi Purwacaraka Music Studios

Visi

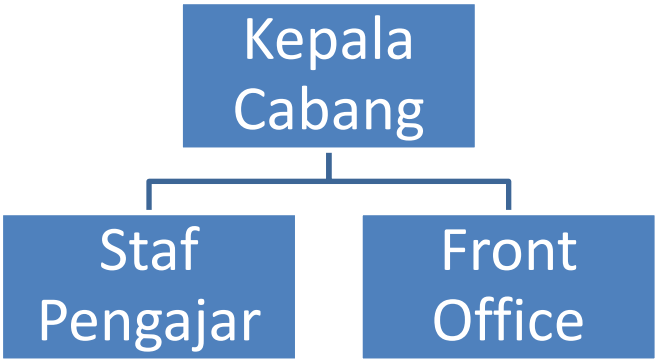
a) Melalui system franchise diharapkan dapat menjadi lembaga pendidikan musik (music studio) terbesar di Indonesia.
b) Memiliki metoda pengajaran musik yang baku yang dapat dijadikan acuan bagi pendidikan musik di Indonesia.
c) Kami akan selalu mewujudkan “We Teach Music Better”.

Misi

a) Menjadi mitra masyarakat dalam peningkatan kualitas pendidikan musikdi Indonesia.
b) Menjadikan musik sebagai bagian dalam kehidupan keluarga dalam rangka peningkatan kualitas hidup masyarakat Indonesia.

3.1.3 Stuktur Orgnanisasi

Berikut adalah struktur organisasi yang berada di Purwacaraka Muc Studio Jl. Karwitan 144 Bandung:



Gambar 3.1 Stuktur Organisasi Purwacaraka Music Studio

3.1.4 Deskripsi Tugas

1. Kepala Cabang

   Pengarah dan pengatur manajemen dari PCMS terkait dengan para karyawannya. Mengajukan program kerja pada Owner serta mengadakan event-event atau special event sebagai bentuk Apresiasi seni, unjuk kebolehan, uji hasil studi, sekaligus melakukan Marketing bersamaan dengan pelaksanaan event-event tertentu. Kepala Cabang berhubungan langsung dengan Ir. Purwa Tjaraka dalam hal penerimaan surat lamaran calon pengajar/guru yang akan diajukan kepada Ir. Purwa Tjaraka untuk di audisi atau seleksi.

2. Staf Pengajar Purwacaraka Music Studio

   Purwacaraka Music Studio memastikan kualitas pendidikan yang tinggi, masing-masing Staf pengajar telah lulus kualifikasi, termasuk melalui tahap audisi dan wawancara yang dilakukan secaa pribadi oleh Purwacaraka selaku kepala sekolah. Seminar tutorial yang diberikan kepada staf pengajar di

Purwacaraka Music Studio yang diberikan kepada staf pengajar secara berkala, menetapkan standar pengajaran dan pengetahuan yang tepat untuk melakukan kurikulum dai Purwacaraka Music Studio. Staf pengajar purwacaraka Music Studio mampu menyediakan komunikasi yang baik dengan siswa maupun orang tua.

3. Front Office

   Ditugaskan untuk bagian pendaftaran bagi siswa baru yang ingin bergabung di Purwacaraka Music Studio dan pembayaran biaya kursus bagi siswa yang telah bergabung. Adapun fungsi lain memberikan informasi kepada siswa yang sudah dan belum bergabung.

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. 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 model *prototipe*.

3.2.1. Desain Penelitian

Penelitian deskriptif dapat diartikan sebagai proses pemecahan masalah yang diselidiki dengan menggambarkan keadaan subyek atau obyek penelitian pada saat sekarang berdasarkan fakta-fakta yang terlihat.

Deskriptif hanya menggambarkan keadaan terhadap objek yang di teliti dan tanpa adanya pengujian hipotesis. Dengan penelitian deskriptif melalui pendekatan kasus, akan didapatkan suatu pemecahan masalah yang lebih terarah pada sasaran yang akan dicapai.

Penelitian dilakukan dengan tahapan-tahapan yaitu sebagai berikut:

1. Memahami kebutuhan *user*.

   Yaitu menganalisis dan menentukan setiap kebutuhan yang diperlukan dalam membangun dan merancang sistem.

2. Mendeskripsikan kebutuhan *user*.

   Mengidentifikasi setiap permasalahan yang muncul untuk dihasilkan sebuah analisis penyelesaian permasalahan. Mengidentifikasi yaitu dengan mengumpulkan setiap kebutuhan yang diperlukan dan menunjang terbentuknya penyelesaian permasalahan *user*.

3. Merancang prototipe.

   Membuat sebuah model penyelesaian sebagai bahan pertimbangan dari kebutuhan yang ingin dicapai oleh user. Model dapat dijadikan bentuk penyelesaian yang bisa dianalisis dan diperbaiki kembali untuk mencapai keinginan user.

4. Mengevaluasi perancangan.

   Model penyelesaian yang telah dirancang, dilakukan kemudian akan di evaluasi oleh *user* untuk menentukan kekurangan dan kelemahan dari sistem yang dirancang.

3.2.2. Jenis dan Metode Pengumpulan Data

Koleksi data merupakan tahapan dalam proses penelitian yang penting, karena hanya dengan mendapatkan data yang tepat maka proses penelitian akan berlangsung sampai peneliti mendapatkan jawaban dari perumusan masalah yang sudah ditetapkan. Jenis pengumpulan data yang digunakan oleh penulis ada 2, yaitu data primer dan data sekunder. Data primer adalah data yang hanya dapat kita peroleh dari sumber asli (nara sumber). Data sekunder merupakan data yang sudah tersedia sehingga kita hanya cukup mencari dan mengumpulkannya.

3.2.2.1. Data Primer

Data primer penulis dapatkan dari hasil observasi dan wawancara. Berikut adalah penjelasannya :

1. Observasi yaitu penulis dapat mengamati semua proses pengajaran alat musik di kelas piano, gitar dan drum yang kemudian diambil kesimpulan dari hasil pengamatan tersebut.
2. Wawancara yaitu penulis melakukan wawancara secara langsung dengan Bpk. Putut Nuryawan untuk memperoleh informasi tentang proses pembelajaran alat musik yang sedang berjalan. Wawancara dilakukan pada bagian Kepala Cabang.

3.2.2.2. Data Sekunder

Data sekunder yang digunakan yaitu sumber tertulis yang antara lain terdiri dari:

1. Sumber buku, jurnal ilmiah dan internet

Buku-buku, jurnal ilmiah, dan data yang berasal dari internet sangat berguna untuk menunjang penelitian, yakni sebagai tinjauan pustaka untuk menganalisis penelitian yang dilakukan. Selain itu, juga terdapat standar dan peraturan yang terkait dengan penelitian untuk mengevaluasi keadaan yang ada.

2. Dokumentasi yaitu dengan mengumpulkan dokumen-dokumen perusahaan yang mendukung penelitian baik itu jurnal perusahaan, profil perusahaan, harga bus, jadwal keberangkatan dan sebagainya.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Dalam merancang suatu sistem informasi 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.

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. ”

Kelebihan yang dimiliki oleh metode pendekatan berorientasi objek adalah sebagai berikut:

1. Dibandingkan dengan metode SSAD, OOAD lebih mudah digunakan dalam pembangunan sistem
2. Dibandingkan dengan SSAD, waktu pengembangan, level organisasi, ketangguhan,dan penggunaan kembali (reuse) kode program lebih tinggi dibandingkan dengan metode SSAD (Sommerville, 2000).
3. Tidak ada pemisahan antara fase desain dan analisis, sehingga meningkatkan komunikasi antara user dan developer dari awal hingga akhir pembangunan sistem.
4. Analis dan programmer tidak dibatasi dengan batasan implementasi sistem, jadi desain dapat diformalisikan yang dapat dikonfirmasi dengan berbagai lingkungan eksekusi.
5. Relasi obyek dengan entitas (thing) umumnya dapat di mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam sistem. Hal ini memudahkan dalam mehami desain (Sommerville, 2000).

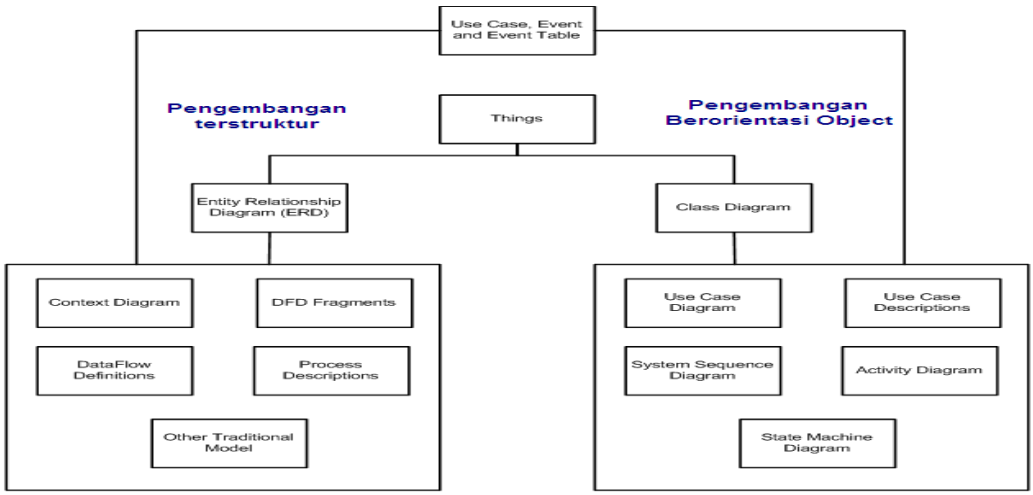
6. Memungkinkan adanya perubahan dan kepercayaan diri yang tinggi terhadap kebernaran *software* yang membantu untuk mengurangi resiko pada pembangunan sistem yang kompleks (Booch, 2007).
7. *Encapsuliation data* dan *method*, memungkinkan penggunaan kembali pada proyek lain, hal ini akan memperingan proses desain, pemrograman dan reduksi harga.
8. OOAD memungkinkan adanya standarisasi obyek yang akan memudahkan memahami desain dan mengurangi resiko pelaksanaan proyek.
9. Dekomposisi obyek, memungkinkan seorang analis untuk memecah masalah menjadi pecahan-pecahan masalah dan bagian-bagian yang di-*manage* secara terpisah. Kode program dapat dikerjakan bersama-sama. Metode ini memungkinkan pembangunan *software* dengan cepat, sehingga dapat segera masuk ke pasaran dan kompetitif. Sistem yang dihasilkan sangat fleksibel dan mudah dalam memelihara.

Kekurangan yang dimiliki oleh metode pendekatan berorientasi objek adalah sebagai berikut:

1. Pada awal desain OOAD, sistem mungkin akan sangat *simple.*
2. Pada OOAD lebih fokus pada coding dibandingkan dengan SSAD.
3. Pada OOAD tidak menekankan pada kinerja team seperti pada SSAD.
4. Pada OOAD tidak mudah untuk mendefinisikan *class* dan obyek yang dibutuhkan sistem.

5. Sering kali pemrogramam berorientasi obyek digunakan untuk melakukan anlisisis terhadap fungsional sistem, sementara metode OOAD tidak berbasis pada fungsional sistem.
6. OOAD merupakan jenis manajemen proyek yang tergolong baru, yang berbeda dengan metode analisis dengan metode terstruktur. Konsekuensinya adalah, team developer butuh waktu yang lebih lama untuk berpindah ke OOAD, karena mereka sudah menggunakan SSAD dalam waktu yang lama (Hantos, 2005).
7. Metodologi pengembangan sistem dengan OOAD menggunakan konsep reuse. Reuse merupakan salah satu keuntungan utama yang menjadi alasan digunakannya OOAD. Namun demikian, tanpa prosedur yang emplisit terhadap reuse, akan sangat sulit untuk menerapkan konsep ini pada skala besar (Hantos, 2005).

Adapun secara logika alur pengembangan sistem di gambarkan sebagai berikut :

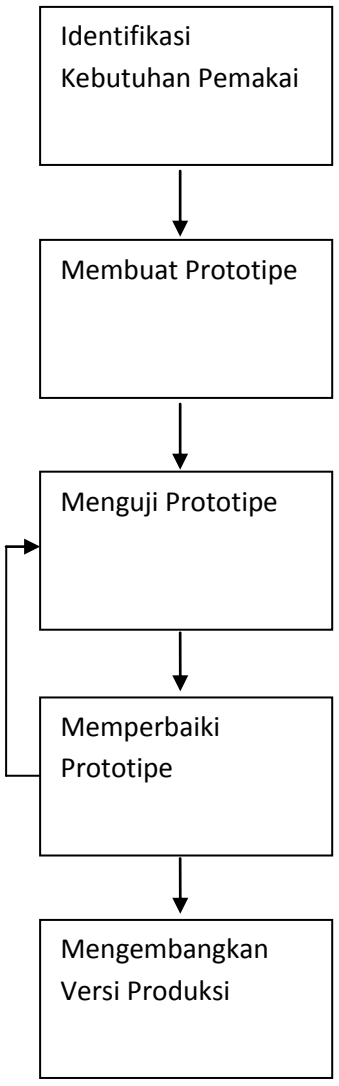


Gambar 3.2. Model Pendekatan Pengembangan Sistem

3.2.3.2. Metode Pengembangan Sistem

Metode yang penulis gunakan dalam pengembangan sistem adalah Metode Prototipe. Metode Prototipe merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh pemakai (*user*). Dari pengertian metode prototipe diatas penulis akan memberikan beberapa alas an mengapa penulis menggunakan metode pengembangan sistem dengan prototipe yaitu dikarenakan penulis akan lebih mudah dalam merancang sistem yang di inginkan perusahaan dan dapat diterima oleh *user* sebagai pemakai, penulis menginginkan perancangan sistem yang telah dihasilkan kemudian dipresentasikan kepada *user* dan *user* diberikan kesempatan untuk memberikan masukan-masukan sehingga sistem informasi yang dihasilkan betul-betul sesuai dengan keinginan dan kebutuhan *user*, terutama bagi perusahaan sendiri. Perubahan dan presentasi prototipe ini dapat dilakukan berkali-kali sampai dicapai kesepakatan bentuk sistem informasi yang akan diterapkan.

Metode prototipe dirancang agar dapat menerima perubahan-perubahan dalam rangka menyempurnakan prototipe yang sudah ada sehingga pada akhirnya dapat menghasilkan sistem informasi yang dapat diterima dan perubahan-perubahan yang terjadi dianggap dapat merupakan sebagian dari proses pengembangan itu sendiri.



1. Pengembang dan pemakai
2. Pemakai menjelaskan kebutuhan system
3. Pengembang mulai membuat prototype
4. Pemakai menguji prototype dan memberikan kritikan atau saran
5. Pengembang melakukan modifikasi sesuai dengan pemasukan pemakai
6. Pengembang merampungkan system sesuai dengan masukan terakhir dari pemakai.

Gambar 3.3 Mekanisme Pengembangan Sistem dengan Prototipe.
(Sumber: Abdul Kadir, 2003)

Berikut adalah langkah-langkah penulis dalam merancang sebuah sistem yang menggunakan mekanisme pengembangan sistem dengan prototipe yang sesuai dengan gambar diatas, langkah- langkah tersebut antara lain:

1. Penulis akan mengidentifikasi kebutuhan *user*, supaya penulis bisa merancang sistem yang akan dibangun sesuai dengan yang diharapkan *user*. Sebelum pada tahap perancangan, penulis akan memulai pada tahap awal terlebih

dahulu yaitu penulis akan menganalisis sistem dengan cara melakukanmengumpulkan data yaitu dengan *field reserch* (metode penelitian)/observasi, dan *interview* (wawancara) dan dengan cara literatur yaitu dengan dokumentasi terhadap kebutuhan yang diinginkan pemakai, baik dalam model *interface*, teknik, prosedural maupun dalam teknologi yang akan digunakan.

2. Pada tahap kedua yaitu membuat prototipe, penulis akan membuat prototype sistem tersebut untuk memperlihatkan kepada pemakai model sistem yang akan dirancang.
3. Pada tahap ketiga yaitu pengujian prototipe, penulis akan melakukan uji coba sistem yang telah dirancang untuk memastikan bahwa sistem tersebut dapat digunakan dengan baik dan benar, sesuai kebutuhan pemakai.
4. Pada tahap keempat yaitu memperbaiki prototipe, penulis akan menentukan apakah sistem tersebut dapat diterima oleh pemakai, atau harus dilakukan beberapa perbaikan atau bahkan dibongkar semuanya dan mulai dari awal lagi, dan setelah perbaikan sistem itu selesai dikerjakan, penulis akan kembali lagi pada tahap yang ketiga yaitu dengan melakukan pengujian prototipe kembali.
5. Pada tahap kelima, tahap terakhir yaitu mengembangkan versi produksi, penulis akan merampungkan sesuai dengan masukan terakhir dari pemakai dan memberikan gambaran bagaimana penggunaan sistem tersebut kepada pemakai setelah sistem tersebut disetujui.

3.2.3.3. Alat Bantu Analisis dan Perancangan

Pada perancangan sistem informasi 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 (Suhendar, 2002).

UML merupakan gabungan dari Booch, OMT, OOSE dengan membuang elemen-elemen yang tidak praktis, ditambah dengan elemen-elemen lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu, sehingga UML lebih ekspresif dan seragam daripada metode lainnya (Suhendar,2002).

a) Diagram *Use case*

   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.

b) Diagram *Class*

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.

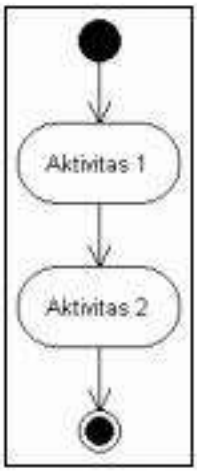
c) Diagram Aktivitas

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. Diagram aktivitas adalah sebagaimana pada Gambar 3.4.

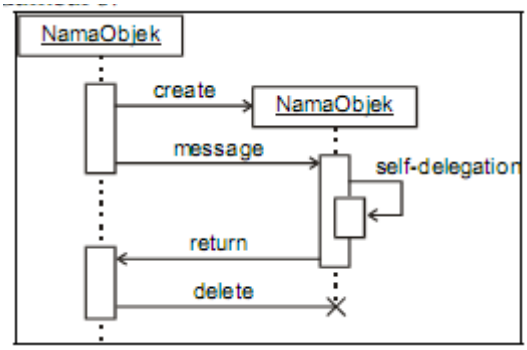


Gambar 3.4. Diagram Aktivitas

d) Diagram Interaksi

Diagram interaksi menggambarkan interaksi antar Obyek dalam sebuah urutan waktu. Nama asli diagram interaksi dalam UML adalah *sequence* diagram, namun diterjemahkan sebagai diagram interaksi agar lebih tepat.

Diagram interaksi dapat digunakan untuk memperjelas setiap *use case*. Sebuah *use case* dapat melibatkan lebih dari satu Obyek, dan interaksi antar Obyek akan mudah diamati dengan diagram interaksi. Contoh diagram interaksi ditunjukkan pada Gambar 3.5.



Gambar 3.5. Diagram Interaksi (*Sequence*)

e) Diagram Kolaborasi

Diagram kolaborasi merupakan sebuah cara alternatif dalam menggambarkan interaksi antar obyek. Tipe diagram ini menekankan pada hubungan (relationship) antar obyek, sedangkan *sequence* diagram lebih menekankan pada urutan kejadian.

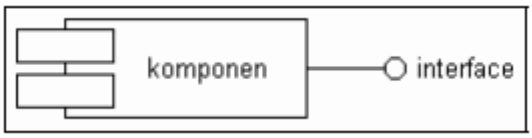
Dalam satu diagram kolaborasi terdapat beberapa obyek, *link*, dan *message*. Diagram kolaborasi digunakan sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem.

f) Diagram Keadaan

Diagram ini memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah obyek, kejadian yang menyebabkan transisi dari satu *state* atau aktivitas kepada yang lainnya, dan aksi yang menyebabkan perubahan satu *state* atau aktivitas.

g) Diagram Komponen

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. Gambar 3.6. menunjukkan diagram komponen.



Gambar 3.6. Diagram Komponen

h) *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.

3.2.4. 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.

Pengujian *black box* bukan merupakan alternatif dari pengujian *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*.

Pengujian *black box* digunakan untuk menemukan kesalahan dalam beberapa kategori, diantaranya adalah :

1) Fungsi-fungsi yang salah atau hilang
2) Kesalahan *interface*
3) Kesalahan dalam struktur data atau akses *database* eksternal
4) Kesalahan performa
5) Kesalahan inisialisasi dan terminasi