BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Objek penelitian merupakan suatu sasaran penelitian dengan tujuan dan kegunaan tertentu untuk mendapatkan sesuatu yang bermanfaat dalam suatu penelitian.

Penelitian ini dilakukan di SD Negeri Cihaurgeulis 1 yang beralamat beralamat Jl.Surapati No 155 bandung 40123.Penelitian ini dilaksanakan dikelas 4 sampai kelas 6. Pada penelitian ini, penulis menjelaskan tentang sejarah, visi dan misi serta lambang sekolah yaitu pemerintah kota bandung.

3.1.1. Sejarah Singkat Perusahaan

SD Negeri Cihaurgeulis berdiri pada tahun 1974 dengan nama pertama SD Impres,pecahan dari SD Melati.Kepala Sekolah yang pertama yaitu Drs.Rusde ,yang merupakan kepala sekolah SD Negeri Cihaurgeulis 1.Pertama berdiri SD Negeri Cihaurgeulis dibangun hanya 1 bangunan,yang terdiri dari 4 ruangan,diantaranya :

1. 3 Ruang kelas
2. 1 Ruang kepala sekolah dan guru

Jumlah guru yang mengajar yaitu 6 orang ditambah 1 orang guru honor dan 1 orang penjaga sekolah.Kemudian tahun 1975 berkembang menjadi 3 SD,diantaranya SD Negeri Cihaurgeulis 1 dan 2 SD Lainnya yaitu SD Negeri

Cihaurgeulis 3 dan SD Negeri Cihaurgeulis 5 dan Dipimpin oleh kepala sekolah Epon sayati dan Ibu tuti.

Pada Tahun 1998 SD Negeri Cihaurgeulis 5 digabungkan ke SD Negeri Cihaurgeulis 1 dan 3 karena jumlah murid yang kurang dari 150 siswa,selain itu sebagian gurunya di mutasi ke SD lain,kemudian tahun 2010 penggabungan SD Negeri Cihaurgeulis 3 menjadi SD Negeri Cihaurgeulis 1 dengan 1 kepala sekolah dan 20 orang guru dengan jumlah murid 315 orang.

Pada tahun ajaran 2011 dan 2012 kelas bertambah menjadi 14 kelas dengan jumlah guru 22 guru PNS(Pegawai Negeri Sipil) dan 11 guru honorer serta 1 penjaga sekolah,dengan Jumlah siswa 412 orang sampai sekarang.



Gambar 3.1 Logo Pemerintah Kota Bandung

Arti Lambang

Lambang kota Bandung ditetapkan dengan Peraturan Daerah Kota besar Bandung tahun 1953, tertanggal 8 Juni 1953, yang diijinkan dengan Keputusan Presiden tertanggal 28 april 1953 No. 104 dan diundangkan dalam Berita Propinsi Jawa Barat tertanggal 28 Agustus 1954 No. 4 lampiran No. 6 Lambang tersebut bertokoh PERISAI yang berbentuk JANTUNG. Perisai tersebut terbagi dalam dua bagian oleh sebuah BALOK- LINTANG mendatar bertajuk empat buah, yang

berwarna HITAM dengan pelisir berwarna PUTIH(PERAK) pada pinggir sebelah atasnya:

1. bagian atas latar KUNING (EMAS) dengan lukisan sebuah GUNUNG berwaarna HIJAU yang bertumpu pada blok-lintang daaan

2. bagian bawah latar PUTIH(PERAK) dengan lukisan empat bidaang jalur mendatar berombak yang berwarna BIRU.

Di bawah perisai itu terlukis sehelai PITA berwarna KUNING (EMAS) yang melambai pada kedua ujungnya, Pada pita itu tertulis dengan huruf-huruf besar latin berwarna HITAM amsal dalam bahasa KAWI, yang berbunyi GEMAH RIPAH WIBAWA MUKTI.

Sebagai tokoh lambang itu diambil bentuk perisai atau tameng, yang dikenal kebudayaan dan peradaban sebagai senjata dalam perjuangan untuk mencapai sesuatu tujuandengan melindungi diri. Perkakas perjuangan yang demikian itu dijadikan lambang yang mempunyai arti menahan segala mara bahaya dan kesukaran.

1. KUNING (EMAS), berarti : kesejahteraan, keluhungan.
2. HITAM (SABEL), berarti : kokoh, tegak, kuat.
3. HIJAU (SINOPEL), berarti : kemakmuran sejuk
4. PUTIH (PERAK), berarti : kesucia
5. BIRU (AZUUR), berarti : kesetiaan

Arti Motto Kota Bandung

Gemah ripah wibawa mukti, berarti : tanah subur rakyat makmur.

Arti Lambang Bendera :

Bendera yang digunakan oleh Kotamadya Bandung adalah berdasarkan Surat Keputusan Dewan Perwakilan Rakyat Daerah Sementara Kota Besar Bandung tanggal 8 Juni 1953 No. 9938/53.

Bentuk bendera tersebut adalah seperti yang tercantum pada diktum Keputusan tersebut diatas sebagai berikut :

1. Bendera yang dipergunakan oleh Kota Besar Bandung dan tiga bidang jalur mendatar, masing- masing berturut-turut dari atas kebawah berwarna HIJAU, KUNING dan BIRU

2. Perbandingan-perbandingan antara lebarnya dan jalur-jalur tersebut dibawah huruf a urutan dari atas kebawah adalah 2:1:1;

3. Perbandingan antara panjang dan lebarnya berbeda itu 7:5.

3.1.2. Visi dan Misi

Visi

1. Mewujudkan tercapainya peningkatan mutu pendidikan.
2. Mengembangkan suasana kondusif.
3. Meningkatkan budaya pelayaran.
4. Mengembangkan sikap *profesionalisme* guru.

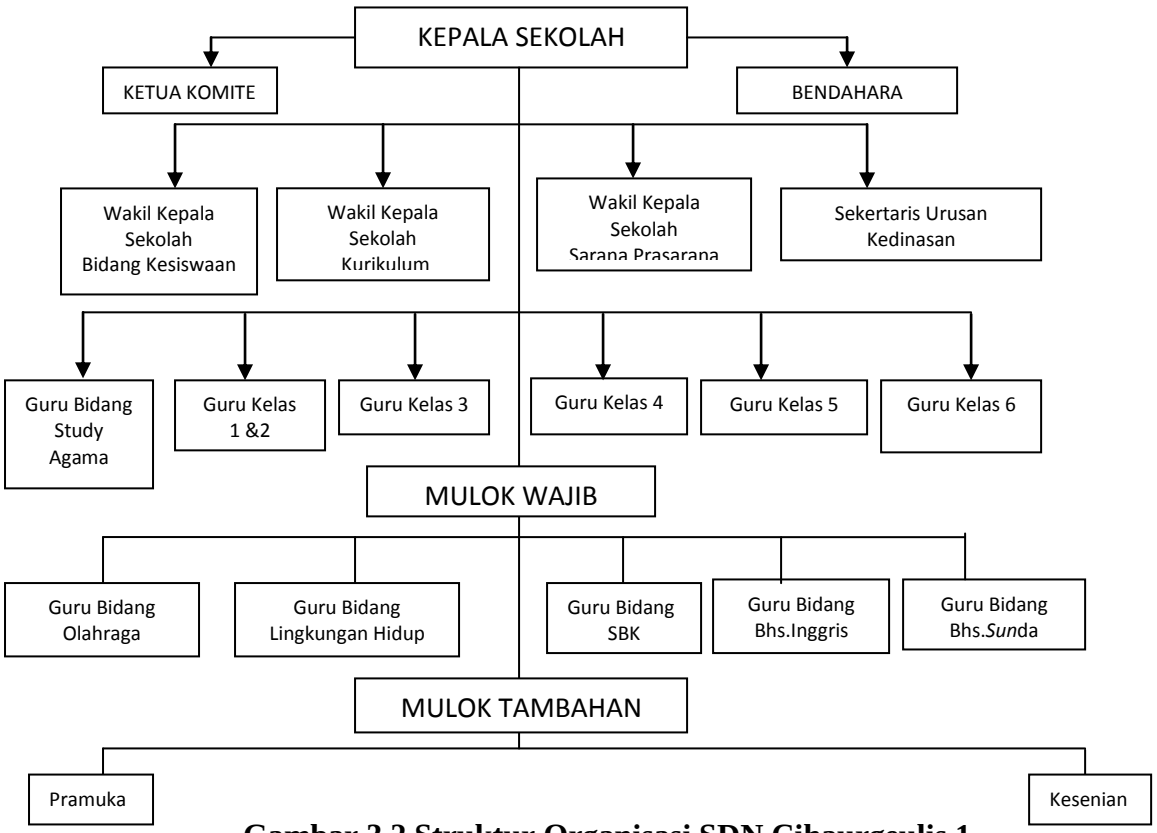
Misi

1. Berorentasi pada kualitas.
2. Terwujudnya manusia yang berpendidikan dan terampil serta takwa kepada tuhan Yang Maha Esa.
3. Mengembangkan kebudayaan dan lingkungan hidup.

Strategi

1. Peralatan kelembagaan.
2. Meningkatkan proses kegiatan belajar mengajar.
3. Meningkatkan kualitas bekerja kependidikan.
4. Meningkatkan profesionalisme kinerja kependidikan.

3.1.3. Struktur Organisasi



Gambar 3.2 Struktur Organisasi SDN Cihaurgeulis 1

3.1.4 Deskripsi Tugas

Menurut Jogiyanto (2005: 28) *Deskripsi tugas merupakan suatu rincian yang menunjukan posisi, tanggung jawab, wewenang, fungsi dan tugas-tugas yang harus dikerjakan oleh seorang personil di dalam suatu organisasi.*

Kepala sekolah :

1. Dibantu oleh para wakasek dan kaur TU merencanakan program kerja sekolah meliputi bidang umum, kurikulum, kesiswaan, ketenagaan, sarana prasarana, keuangan, ketatausahaan, dan hubungan masyarakat.
2. Mengorganisasikan semua komponen yang ada disekolah dalam pembagian tugas sesuai dengan bidang keahliaanya.
3. Melaksanakan program kerja sekolah dibantu oleh semua komponen yang ada di sekolah.
4. Melaksanakan *supervise* meliputi : pemeriksaan administrasi PBM, Kunjungan kelas, pemeriksaan sarana prasarana, pemeriksaan administrasi tata usaha, pemeriksaan BP/BK, pemeriksaan keuangan rutin dan komite (partisipai masyarakat).

Wakil Kepala Sekolah Bidang Kesiswaan :

1. Menyusun program kesiswan/OSIS
2. Melaksanaan bimbingan, pengarahan dan pengendalian kegiatan siswa/OSIS dalam rangka menegakkan disiplin dan tata tertib sekolah
3. Membina dan melaksanakan koordinasi K6
4. Memberikan pengarahan dalam pemilihan pengurus OSIS
5. Melaksanakan pembinaan pengurus OSIS dalam berorganisasi

6. Menyu*Sun* program dan jadwal pembinaan siswa secara berkala dan insidentil.
7. Melaksanakan pemilihan calon siswa teladan.
8. Mengadakan pemilihan siswa untuk mewakili sekolah dalam kegiatan di luar sekolah.
9. Menyusun laporan pelaksanaan kegiatan kesiswaan secara berkala.
10. Mengatur mutasi siswa.
11. Pembinaan terhadap ekstrakulikuler, Pramuka, PMR, Kesenian, Olah raga, Kerohanian.

Wakil Kepala Sekolah Bidang Kurikulum :

1. Menyusun program pengajaran.
2. Menyusun pembagian tugas guru dan jadwal pelajaran.
3. Menyusun program ulangan blok, unas dan pelaksanaan program.
4. Menyusun jadwal evaluasi belajar.
5. Menyu*sun* pelaksanaan UAS/UNAS.
6. Menetapkan kriteria persyaratan naik kelas/tidak naik kelas.
7. Menetapkan jadwal penerimaan Buku Laporan Hasil Belajar Siswa dan Ijazah.
8. Mengkoordinasikan dan mengarahkan penyusunan satuan pelajaran.
9. Menyediakan buku kemajuan kelas.
10. Menyusun laporan pelaksanaan pelajaran.

Wakil Kepala Sekolah Sarana Prasarana :

1. Menyusun rencana kebutuhan sarana dan prasarana sekolah.
2. Mengadministrasikan pendayagunaan sarana dan prasarana.
3. Pengelolaan pembiayaan alat-alat pengajaran.
4. Menyusun laporan urusan sarana dan prasarana secara berkala.

Sekertaris Urusan Kedinasan :

Guru :

1. Hadir di sekolah 15 menit sebelum pelajaran dimulai.
2. Menandatangani daftar hadir.
3. Hadir dan meninggalkan kelas tepat waktu.
4. Melaksanakan semua tugasnya secara tertib dan teratur.
5. Menguasai kurikulum dan materi pelajaran.
6. Membuat program tahunan pada setiap awal tahun pelajaran.
7. Membuat program semester pada awal semester.
8. Membuat persiapan mengajar
9. Melaksanakan praktik untuk mata pelajaran yang memerlukan praktik
10. Melaksanakan ulangan harian

3.2. Metode Penelitian

Metode penelitian merupakan cara penelitian yang digunakan untuk mendapatkan data untuk mencapai tujuan tertentu. **Menurut Sugiyono (2009:3)** "*Metode penelitian diartikan sebagai cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu*".

Metode yang digunakan oleh penulis adalah metode deskriptif, yaitu metode yang menggambarkan atau menguraikan keadaan situasi pada tempat observasi, melakukan penelitian dan kemudian melakukan analisis sehingga dapat diperoleh suatu kesimpulan. Jenis penelitian yang dilakukan dalam penelitian ini adalah sumber data primer.

3.2.1. Desain Penelitian

Dalam melakukan suatu penelitian untuk perancangan *game edukasi* ini sangat perlu dilakukan perencanaan dan perancangan penelitian, agar penelitian yang dilakukan dapat berjalan dengan baik dan sistematis. Desain penelitian **menurut Moh. Nazir (2003:84)** memaparkan bahwa *"Desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksanaan penelitian".*

Desain penelitian untuk merancang english education game yaitu semua proses penelitian yang dilakukan oleh penulis dalam melaksanakan penelitian mulai dari perancangan game sampai dengan pelaksanaan penelitian yang dilakukan pada waktu tertentu.

3.2.2. Jenis dan Metode Pengumpulan Data

Dalam melakukan penelitian ini penulis menggunakan metode deskriptif yaitu suatu metode dalam meneliti status sekelompok manusia suatu objek ,suatu set kondisi,suatu sistem penukiran ataupun mengenai fakta-fakta dan informasi dalam situasi atau kejadian dimasa sekarang secara sistematis faktual dan akurat.

3.2.2.1. Sumber Data Primer

Sumber data primer yaitu berupa teknik pengumpulan data yang diigunakan adalah sebagai berikut :

1. Observasi

Dalam hal ini penyusun melakukan pengamatan pada SD Negeri Cihaurgeulis 1 untuk mendapatkan data secara umum dengan melihat langsung dan mengamati.

2. Wawancara

Dalam hal ini penyusun melakukan wawancara untuk melengkapi bahan yang sudah ada selama observasi. Penyusun melakukan tanya jawab langsung kepada guru bahasa Inggris SD Negeri Cihaurgeulis 1.

3.2.2.2. Sumber Data Sekunder

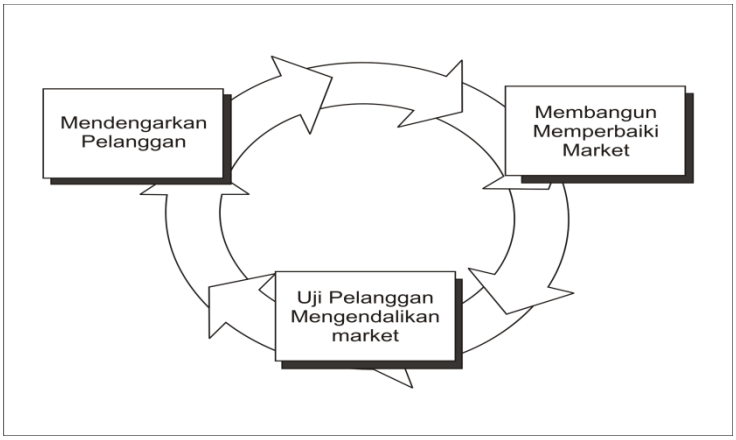
Sedangkan data sekunder diperoleh dengan cara melakukan kajian kepustakaan,yaitu mengkaji buku-buku literature,makalah,informasi melalui internet yang *relevan* dengan *variable* yang diteliti selain itu sumber data juga diperoleh dari instansi terkait yang ada hubungannya dengan *variable* yang digunakan dalam penelitian ini.Yang menjadi sumber dokumen sekunder penulis yaitu buku J2ME.

3.2.3. Metode Pendekatan Sistem

Metode Pendekatan sistem yang akan digunakan metode analisis dan pemrograman berorientasi objek. Alat-alat yang digunakan dalam pendekatan analisis dan pemrograman berorientasi objek yaitu dengan notasi UML dengan membuat enam diagram yaitu, *Use case diagram, Activity diagram, Class diagram,Sequence diagram,Component diagram,Deployment diagram*.Sedangkan perancangan program dimulai dari *Design,* Perancangan, *Cooding, Testing*.

3.2.4. Metode Pengembangan Sistem

Dalam perancangan *english education game* ini, penulis menggunakan metode *prototyping* yang merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Desain penelitian ini dimodelkan dengan menggunakan model proses *prototype*. Model proses *prototype* merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh pemakai *(user).* Berikut gambar dari metode pengembangan yang digunakan penulis :



Gambar 3.3 Metode Pendekatan *Prototype* Paradigma

(Sumber : Pressman, Roger S., 2002, *Rekayasa Perangkat Lunak*: Pendekatan Praktisi jilid Dua, Penerbit: Andi Offset, Yogyakarta).

Dari pengertian metode *prototype* diatas, penulis akan memberikan beberapa alasan mengapa penulis menggunakan metode pengembangan sistem dengan

prototype, yaitu dikarenakan penulis akan lebih mudah dalam merancang sistem yang diinginkan dan dapat diterima oleh *user* sebagai pemakai.

Metode *protoype* dirancang agar dapat menerima perubahan-perubahan dalam rangka menyempurnakan *prototype* yang sudah ada sehingga pada akhirnya dapat menghasilkan sistem informasi yang dapat diterima dan memberikan gambaran bagaimana penggunaan sistem tersebut kepada pemakai setelah sistem tersebut disetujui.

Tahapan-tahapan *Prototyping*

Tahapan-tahapan dalam *Prototyping* adalah sebagai berikut :

1. Pengumpulan kebutuhan

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

2. Membangun *prototyping*

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

3. Evaluasi *prototyping*

   Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginann pelanggan. Jika sudah sesuai maka langkah 4 akan diambil.Jika tidak *prototyping* direvisi dengan mengulangi langkah 1, 2 , dan 3.

4. Mengkodekan sistem

Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

6. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan , langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

7. Menggunakan sistem

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

Keunggulan dan Kelemahan *Prototyping*

Keunggulan *prototyping* adalah:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
3. Pelanggan berperan aktif dalam pengembangan sistem
4. Lebih menghemat waktu dalam pengembangan sistem
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Kelemahan *prototyping* adalah :

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.

3.2.5 Alat Bantu Analisis dan Perancangan

3.2.5.1. Alat Bantu Analisis

a. *Unifed Modeling Language* (UML)

Menurut Bambang Hariyanto (2010 : 259) *UML adalah bahasa grafis untuk mendokumentasi,menspesifikasikan , dan membangun sistem perangkat lunak.*UML berorientasi objek,menerapkan banyak level abstraksi, tidak bergantung proses pengembangan,tidak bergantung bahasa dari teknologi,pemaduan beberapa notasi diberagam metodologi,usaha bersama dari banyak pihak,didukung oleh kakas-kakas yang diintegrasikan lewat XML.Standar UML dikelola oleh OMG(Object Management Group).

b. Diagram – Diagram UML

Diagram mengemukakan banyak hal,penggunaan notasi yang terdefinisi baik dan ekspresif adalah penting pada proses perangkat lunak :

1. Notasi standar memungkinkan pengembangan mendeskripsikan scenario atas rumusan arsitektur dan kemudian mengkomunikasikan secara tidak ambigu.
2. Notasi yang bagus membebaskan otak untuk berkonsentrasi pada masalah-masalah yang lebih lanjut.
3. Notasi yang baik memungkinkan mengeliminasi keperluan pemeriksaan konsistensi dan kebenaran keputusan-keputusan dengan menggunakan tool terotomatisasi.

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk

melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara

umum.Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar *UML* menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

3. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama dan *stereotype*
2. Atribut
3. *Method*

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private,* tidak dapat dipanggil dari luar *class* yang bersangkutan.
b. *Protected,* hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
c. *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface,* yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat lang*Sun*g diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis 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. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar *UML* mendefinisikan *icon* khusus untuk objek *boundary, controller* dan *persistent entity*.

5. *Component* Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) diantaranya.Komponen piranti lunak adalah modul berisi code,baik berisi source code maupun binary code,baik *library* maupun *executabl*e,baik yang muncul pada *compile time* ,*link time* , maupun *run time*.Umumnya komponen terbentuk dari beberapa class dam/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa interface yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

6. *Deployment* Diagram

Diagram *deployment* digunakan untuk memodelkan aspek fisik dari sistem beriorentasi objek,yaitu memodelkan konfigurasi *node* - *node* pengolahan waktu jalan dan komponen-komponen yang tinggal di *node-node* itu.

3.2.5.2. Perancangan

1. NeatBeans IDE*(Integrated Development Environment)*

NetBeans IDE *(Integrated Development Environment)* adalah *IDE (Integrated Development Environment)* open source yang ditulis sepenuhnya dengan bahasa pemrograman J2ME menggunakan platform NetBeans. NetBeans IDE*(Integrated Development Environment)* mendukung pengembangan semua tipe aplikasi J2ME (J2SE, web, EJB, dan aplikasi mobile).Fitur lainnya adalah sistem proyek berbasis Ant,control versi,dan *refactriong*.

Versi terbaru saat ini adalah NetBeans IDE*(Integrated Development Environment)* 5.5.1 yang dirilis Mei 2007 mengembangkan fitur-fitur J2ME EE yang sudah ada (termasuk J2ME Persistence support, EJB-3 dan JAX-WS). Sementara paket tambahannya, NetBeans Enterprise Pack mendukung pengembangan aplikasi perusahaan J2ME EE 5, meliputi alat desain visual SOA, skema XML, web service dan pemodelan UML. NetBeans C/C++ Pack

mendukung proyek C/C++.

Modularitas: Semua fungsi IDE(*Integrated Development Environment)* disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang

didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman J2ME, editing, atau dukungan bagi CVS.

NetBeans memuat semua modul yang diperlukan dalam pengembangan J2ME dalam sekali download, memungkinkan pengguna untuk mulai bekerja sesegera mungkin. Modul-modul juga mengijinkan NetBeans untuk bisa dikembangkan. Fitur-fitur baru, seperti dukungan untuk bahasa pemrograman lain, dapat ditambahkan dengan menginstal modul tambahan. Sebagai contoh, Sun Studio, Sun J2ME Studio Enterprise, dan Sun J2ME Studio Creator dari Sun Microsystem semuanya berbasis NetBeans *IDE(Integrated Development Environment).*

3.2.6 Pengujian *Software* Menggunakan *Black box*

Dalam pengujian software, penulis menggunakan pengujian ***black box*** karena lebih dipahami dan lebih mudah diketahui letak kesalahannya baik dari proses Programnya maupun tampilan atau output yang dihasilkan.

Pengujian *black box* :

1. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.
2. Disebut juga pengujian *behavioral* atau pengujian partisi.
3. Pengujian *blackbox* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.
4. Pengujian *blackbox* berusaha menemukan :
5. Fungsifungsi yang tidak benar atau hilang
    a. Kesalahan *interface.*

b. Kesalahan dalam struktur data atau akses *database* eksternal.

c. Kesalahan kinerja.

d. Inisialisasi dan kesalahan terminasi.

6. Dengan mengaplikasikan teknik blackbox maka kita menarik serangkaian *test case* yang memenuhi kriteria beirkut :

   a. *Test case* yang mengurangi, dengan harga lebih dari satu, jumlah *test case* tambahan yang harus di desain untuk mencapai pengujian yang dapat dipertanggungjawabkan.

   b. *Test case* yang member tahu kita sesuatu mengenai kehadiran atau ketidakhadiran kelas kesalahan, daripada member tahu kesalahan yang berhubungan hanya dengan pengujian spesifik.