BAB 2 LANDASAN TEORI

2.1 Profil SMP AL- Falah Dago

Profil tempat penelitian tugas akhir ini menjelaskan sejarah , logo, visi dan misi serta struktur organisasi di Sekolah Menengah Pertama Al-Falah Dago.

2.1.1 Sejarah instansi

SMP Al Falah Dago adalah sekolah yag berada di bawah naungan Yayasan Pesantren Islam Al Falah (YPIA). Beralamat di Jalan Cisitu Baru No.52 Kelurahan Dago kecamatan Coblong Kota Bandung yang berada dipusat kota Bandung. Berawal dari Pesantren Al Falah yang dibangun secara bertahap sejak tahun lima puluh oleh pendirinya, yaitu almarhum Bapak KH. Saefudin Ahmad atau yang lebih dikenal denga panggilan ustadz Idi, Beliau adalah seorang pendidik yang juga aktif di pergerakkan sejak zaman penjajah Belanda. Jepang bahkan sampai zaman kemerdekaan, Beliau beberapa kali keluar masuk penjara karena perjuangannya yang konsisten untuk menegakkan Syari'at Islam.

Keyakinan akan pentingnya pembangunan sumber daya insani ummat sesuai dengan teladan dari Rasulullah SAW pada tahap awal da'wahnya, bangunan yang pertamakali didirikan adalah ruang belajar (madrasah) dengan mengkhususkan pada pendidikan agama. Kemudian setelah cukup banyak orang yang mau melakukan sholat baru dibangun mesjid. Menurut pendapatnya akan sia-sia dibangun mesjid jika belum ada orang yang mau sholat dan meramaikan mesjid tersebut.

Pada tahun 1958, madrasah diniyyah Al Falah dirubah statusnya menjadi Madrasah Ibtidaiyyah Al Falah, dan Bapak Ustadz H. Saefuddin Ahmad menjadi guru agama yang diperbantukan dari Departemen Agama dan sekaligus merangkap sebagai kepala sekolah. Pada 13 Januari 1961, atas usulan beberapa teman beliau, Pesantren Al Falah dibuat Badan Hukum berupa yayasan yang diberi nama Yayasan Pesantren Islam Al Falah, dengan ketua Bapak KH Ahmad Sobandi, dan Pak Ustadz menjadi wakil ketuanya.

Sekolah lanjutan mulai dirintis pada tahun 1964, yaitu dengan didirikannya sekolah Pendidikan Guru Agama (PGA 4 tahun), tetapi kemudian PGA ini dirubah menjadi Tsanawiyyah pada tahun 1974 sesuai dengan ketentuan Departemen Agama, dimana pada waktu itu sekolah PGA swasta jumlahnya dibatasi dengan alasan sudah kelebihan tenaga pengajar agama islam untuk sekolah dasar.

Karena pertimbangan teknis dimana lulusan madrasah tsanawiyyah pada saat itu tidak dapat melanjutkan ke SMA, serta beberapa pertimbangan teknis dan administratif yang kurang kondusif, maka pada tahun 1976 Madrasah Tsanawiyyah (MTs) Al Falah ini dirubah menjadi Sekolah Menengah Pertama (SMP) Al Falah dengan menerapkan kurikulum Departemen Pendidikan dan Kebudayaan, Pada tahun 1976 ini juga Madrasah Ibtidaiyyah dirubah statusnya menjadi Sekolah Dasar (SD), Dengan demikian sejak tahun ini sekolah-sekolah Al Falah kurikulumnya berorientasi pada Departemen Pendidikan dan Kebudayaan sedangkan pola pengajaran salafiyyah tetap dijalankan secara baik dan konsisten.

2.1.2 Logo SMP AL –Falah Dago



Gambar 2. 1 **Logo Smp Al-Falah Dago**

2.1.3 Visi dan Misi SMP AL-Falah Dago

Visi Sekolah

Menyiapkan Generasi Muslim Yang Bertaqwa, Berbudaya Islami, Berprestasi Dan Berkarya.

Misi Sekolah

a) Menciptakan lingkungan yang religius

b) Menerapkan pendidikan Islam yang terpadu

c) Menciptakan dan mengembangkan jaringan kerja sama

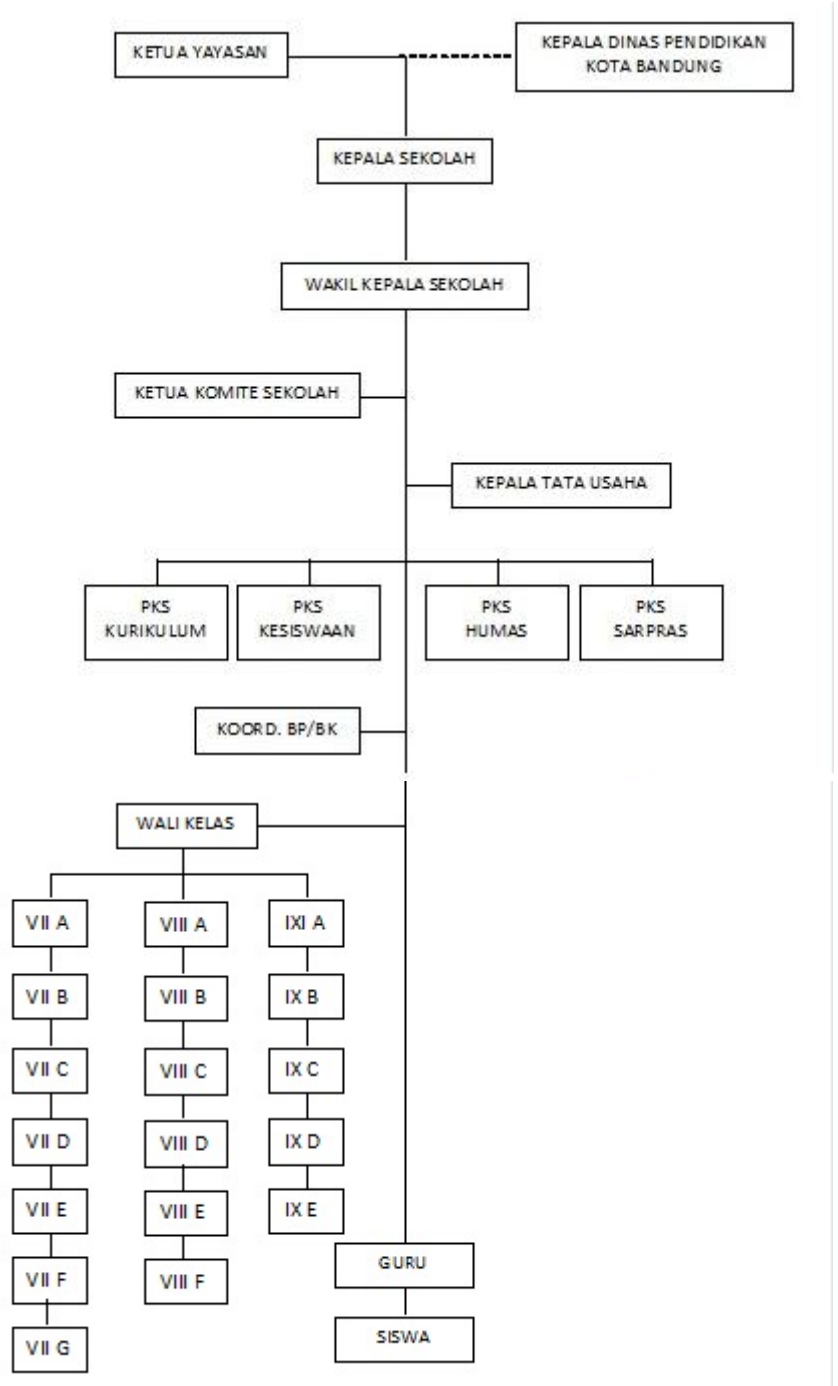
d) Meningkatkan hubungan kekeluargaan di lingkungan sekolah

e) Mendorong kreatifitas dan kemandirian siswa

f) Mengembangkan potensi yang dapat meningkatkan kualitas siswa

g) Meningkatkan profesionalisme kerja guru dan karyawan melalui MGMP, Penataran, dan Pelatihan

2.1.4 Struktur organisasi



Gambar 2. 2 Struktur Organisasi Smp Al-Falah Dago

2.1.5 Deskripsi Pekerjaan

Menganai penjabaran pekerjaan tiap-tiap bagian pada gambar 2.2:

2.1.5.1 WAKASEK KURIKULUM

1. Bekerja sesuai dengan jam belajar mengajar min 12 Jam Pelajaran untuk jabatan dari 12.30 – 17.30
2. Membuat program kegiatan kurikulum selama satu tahun
3. Membuat rumusan Kalender Pendidikan
4. Menyusun jadwal pelajaran untuk satu tahun ajaran
5. Mengkoordinasikan kegiatan ujian – ujian yang akan dilaksanakan SMP Al Falah
6. Mengkoordinasikan nilai – nilai tengah semester dan akhir semester
7. Menentukan pendidik yang akan mengajar di SMP Al Falah
8. Menyiapkan program *supervise*
9. Membuat laporan berkenaan dengan kegiatan – kegiatan kurikulum yang telah dilaksanakan dan bertanggung kepada Kepala Sekolah

TUGAS DAN TANGGUNG JAWAB BAGIAN KURIKULUM SESUAI DENGAN PERMENDIKNAS NO. 19 TAHUN 2007

1. Menyusun dan mengembangkan Kurikulum Tingkat Satuan Pendidikan (KTSP)
2. Menyusun dan Mengembangkan Kelender Pendidikan
3. Mengembangkan dan menkoordinasikan program pembelajaran selama satu tahun sesuai dengan kalender pendidikan yang disepakati dan menjalankan jadwal pelajaran
4. Membuat program kegiatan evaluasi belajaran peserta didik dengan kegiatan diataranya :
   a. Evaluasi belajar mata pelajaran
   b. Evaluasi belajar Ujian Tengah Semester
   c. Evaluasi belajar Ujian Akhir Semeser dan Ujian Kenaikan Kelas
5. Membuat program hasil evaluasi pembelajaran dan mengkoordinasikannya dengan kegiatan :
   a. Program remedial

b. Program pengayaan
c. Absen kegiatan remedial
d. Daftar pengembalian hasil nilai ulangan harian

6. Membuat dan memusyawarahkan peraturan akademik dan mensosialisasikannya kepada seluruh pendidik dan tenaga kependidikan

2.1.5.2 WAKASEK KESISWAAN

1. Bekerja sesuai dengan jam belajar mengajar min 12 Jam Pelajaran untuk jabatan dari 12.30 – 17.30
2. Membuat program kegiatan kesiswaan selama satu tahun
3. Membuat program kegiatan PSB dan mengkoordinasikan dengan pihak sekolah dan Yayasan
4. Mendata peserta didik pada saat masuk PSB dan peserta didik kelas VIII dan IX
5. Memprogram dan mengkoordinasikan kegiatan Ekstrakulikuler
6. Mengkoordinasikan mutasi peserta didik yang masuk atau keluar bekerja sama dengan kepala TU
7. Menyiapkan program supervisi peserta didik
8. Membuat program ketertiban peserta didik dan menertibkan peserta didik selama kegiatan pembelajaran
9. Membuat laporan berkenaan dengan kegiatan – kegiatan kesiswaan yang telah dilaksanakan dan bertanggung kepada Kepala Sekolah

TUGAS DAN TANGGUNG JAWAB BAGIAN KESISWAAAN SESUAI DENGAN PERMENDIKNAS NO. 19 TAHUN 2007

1 Menyusun dan menetapkan petunjuk pelaksanaan operasional mengenai proses penerimaan peserta didik yang meliputi:

a. SMP/MTs berasal dari lulusan SD, MI, Paket A atau satuan pendidikan bentuk lainnya yang sederajat.
b. secara obyektif, transparan, dan akuntabel sebagaimana tertuang dalam aturan sekolah/madrasah.

c. tanpa diskriminasi atas dasar pertimbangan gender, agama, etnis, status sosial, kemampuan ekonomi bagi SD/MI, SMP/MTs penerima subsidi dari Pemerintah dan/atau Pemerintah Daerah.
d. berdasar kriteria hasil ujian nasional.
e. sesuai dengan daya tampung sekolah

2 Memberikan layanan konseling kepada peserta didik.
3 Melaksanakan kegiatan ekstra dan kokurikuler untuk para peserta didik.
4 Melakukan pembinaan prestasi unggulan.
5 Melakukan pelacakan terhadap alumni.

2.1.5.3 WAKASEK SARANA PRASARANA

1. Menyediakan ATK bagi perluan Ketatausahaan.
2. Menyediakan alat – alat administrasi dan alat peraga bagi pendidik dan tenaga kependidikan
3. Menyediakan kebutuhan seragam peserta didik
4. Menyediakan alat pratikum dan sumber belajar bagi pendidik dan berkerja sama dengan bagian laboratorium dan perpustakaan
5. Menggandakan dokumen sekolah dan mengkoordinasikan dengan bagian Tata Usaha

TUGAS DAN TANGGUNG JAWAB BAGIAN SARANA PRASARANA SESUAI DENGAN PERMENDIKNAS NO. 19 TAHUN 2007

1. Menetapkan kebijakan program secara tertulis mengenai pengelolaan sarana dan prasarana, meliputi :
    a. merencanakan, memenuhi dan mendayagunakan sarana dan prasarana pendidikan;
    b. mengevaluasi dan melakukan pemeliharaan sarana dan prasarana agar tetap berfungsi mendukung proses pendidikan;
    c. melengkapi fasilitas pembelajaran pada setiap tingkat kelas di sekolah/madrasah;

d. menyusun skala prioritas pengembangan fasilitas pendidikan sesuai dengan tujuan pendidikan dan kurikulum masing-masing tingkat;
e. pemeliharaan semua fasilitas fisik dan peralatan dengan memperhatikan kesehatan dan keamanan lingkungan.

2. Pengelolaan fasilitas fisik seperti ruangan SMP Al Falah.
3. Pengelolaan fasilitas fisik untuk kegiatan ekstra-kurikuler disesuaikan dengan perkembangan kegiatan ekstra-kurikuler peserta didik dan mengacu pada Standar Sarana dan Prasarana.

2.1.5.4 WAKASEK HUBUNGAN MASYARAKAT

1. Bertanggung jawab kepada Kepala Sekolah atas terwujudnya kerjasaman dengan Dunia Usaha / Dunia Instansi (DU/DI) dan instansi terkait wewenang
2. Menyusun program dan melakukan kegiatan promosi, komunikasi dan kerjasaman dengan Dunia Usaha / Dunia Instansi (DU/DI) dan instansi terkait
3. Melakukan home visit bagi peserta didik yang bermasalah dan melakukan pengontrolan berkenaan keadaan peserta didik dalam rumah bekerjasama dengan wali kelas
4. Memberikan informasi kepada setiap tamu ataupun dari Intansi terkait yang datang ke SMP al Falah
5. Melaporkan setiap kegiatan berkenaan kegiatan kehumasan kepada Kepala Sekolah

2.1.5.5 BAGIAN KONSELING

1. Membuat program kegiatan berkenaan dengan Bimbingan Konseling
2. Pengembangan kehidupan pribadi, yaitu bidang pelayanan yang membantu peserta didik dalam memahami, menilai bakat dan minat.
3. Pengembangan kehidupan sosial, yaitu bidang pelayanan yang membantu peserta didik dalam memahami dan menilai serta mengembangkan kemampuan hubungan sosial dan industrial yang harmonis, dinamis, berkeadilan dan bermartabat.

4. Pengembangan kemampuan belajar, yaitu bidang pelayanan yang membantu peserta didik mengembangkan kemampuan belajar untuk mengikuti pendidikan sekolah/madrasah secara mandiri.
5. Mencatat perkembangan perserta didik selama proses pembelajaran selama tiga tahun
6. Mencatat dan Memberikan konseling kepada peserta didik yang mengalami masalah atau kasus
7. Memberikan surat peringatan kepada peserta didik yang telah melakukan kesalahan dan tidak mentaati peraturan
8. Melakukan test potensi akademik untuk mengetahui IQ dan minat serta bakat peserta didik dan untuk kepentingan lainnya.
9. Memberikan masukan kepada kepala sekolah untuk mengeluarkan peserta didik dari sekolah dengan koordinasi dengan bagian kesiswaan jika melakukan tindakan yang tidak sesuai dengan tata tertib sekolah
10. Melakukan pendataan peserta didik yang melanjutkan atau tidak ke jenjang sekolah yang lebih tinggi
11. Memberikan laporan hasil kegiatan BK selama satu tahun.

2.1.5.6 KEPALA TATA USAHA

1. Menyusun program kerja tata usaha sekolah.
2. Penyusunan administrasi perlengkapan sekolah.
3. Penyusunan dan penyajian data/statistik sekolah.
4. Mengkoordinasikan dan melaksanakan 7 K.
5. Penyusunan laporan pelaksanaan kegiatan, pengurusan ketatausahaan secara berkala.

(Sesuai dengan Keputusan Dirjen Dikdasmen No. 260 dan 261 Tahun 1996 Tugas pokok Kepala Tata Usaha)

1. Bekerja sesuai dengan ketentuan jam pekerjaan Di SMP Al Falah 12.30 – 17.30.
2. Membuat Daftar Nomor Induk Siswa.
3. Mengisi Buku Klaper Siswa.
4. Mengisi Buku Induk Siswa.

5. Mengisi Buku Mutasi Siswa.
6. Membuat Daftar Keadaan Siswa dan Membukukan Daftar Keadaan Siswa.
7. Membukukan Daftar Siswa perkelas.
8. Mencatat Pendaftaran Siswa Baru dari bagian kesiswaan.
9. Menyimpan Daftar kumpulan nilai (*legger*).
10. Membantu bagian TAS yang lain dalam melaksanakan tugasnya.
11. Mendata raport peserta didik dan mendistribusikannya kepada wali kelas.

2.1.5.7 TUGAS DAN TANGGUNG JAWAB WALI KELAS

1. Bekerja min 8 Jam Pelajaran untuk jabatan
2. Mengisi Monografi
3. Membimbing peserta didik sesuai dengan kelas yang Diwali kelaskannya
4. Mendata peserta didik dengan selengkapnya melalui monografi dan dikoordinasikan dengan pihak kesiswaan dan tata usaha :
   a) Mengetahui jumlah anak didik
   b) Mengetahui jumlah anak didik putra (Pa)
   c) Mengetahui jumlah anak didik putrid (Pi)
   d) Mengetahui nama-nama anak didik
   e) Mengetahui identitas lain dari anak didik
   f) Mengetahui kehadiran anak didik setiap hari
   g) Mengetahui masalah-masalah yang dihadapi anak didik (tentang pelajaran, status sosial/ ekonomi, dan lain-lain).
5. Mendata peserta didik yang bermasalah dan dilaporkan kepada kesiswaan dan BK
6. Mengkoordinasikan dan membimbing peserta didik dalam kegiatan – kegiatan kelas
7. Mengisi dan membagikan hasil belajar tengah semester dan akhir semester
8. Mengkoordinasikan mutasi peserta didik yang masuk atau keluar bekerja sama dengan kepala TU
9. Menjaga kebersihan dan ketertiban kelas
10. Mengkoordinasikan dan mengontrol masalah nilai peserta didik

11. Membuat laporan berkenaan dengan kegiatan – kegiatan wali kelas yang telah dilaksanakan dan bertanggung kepada Kepala Sekolah
12. Melakukan home visit (kunjungan ke rumah) untuk mengetahui masalah peserta didik ataupun keadaan kondisi keluarga peserta didik.

2.2 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dari teori-teori yang berkaitan dengan media pembelajaran, pengembangan multimedia menggunakan metode *computer assisted instruction,* serta membahas tentang analisis terstruktur dan tools yang digunakan dalam pengembangan media pembelajaran.

2.2.5 Pembelajaran

Pembelajaran adalah suatu kombinasi yang tersusun meliputi unsur-unsur manusiawi, material fasilitas, perlengkapan dan prosedur yang saling mempengaruhi tercapainya tujuan pembelajaran. Ada 3 (tiga) rumusan yang dianggap lebih maju, yaitu:

1 Pembelajaran adalah upaya mengorganisasikan lingkungan untuk menciptakan kondisi belajar bagi siswa.

2 Pembelajaran adalah upaya mempersiapkan siswa untuk menjadi warga masyarakat yang baik.

3 Pembelajaran adalah suatu proses membantu siswa menghadapi kehidupan masyarakat sehari-hari.

Dalam istilah “pembelajaran” lebih dipengaruhi oleh perkembangan hasil-hasil teknologi yang dapat dimanfaatkan untuk kebutuhan belajar, siswa diposisikan sebagai subjek belajar yang memegang peranan utama sehingga dalam *setting* proses mengajar siswa dituntut beraktifitas secara penuh, bahkan secara individual mempelajari bahan pelajaran. Dengan demikian, jika dalam istilah “mengajar” (pengajaran) atau “*teaching*” menempatkan guru sebagai “pemeran utama” memberikan informasi, maka dalam “*instruction*” guru lebih banyak berperan sebagai fasilitator, memanage berbagai sumber dan fasilitas untuk dipelajari siswa [3].

2.2.5.7 Pengertian Media pembelajaran

Kata media pembelajaran berasal dari bahasa latin "*medius*" yang secara harfiah berarti "*tengah*", perantara atau pengantar. Dalam bahasa Arab, media perantara atau pengantar pesan dari pengirim kepada penerima pesan. Media dalam perspektif pendidikan merupakan instrumen yang sangat strategis dalam ikut menentukan keberhasilan proses belajar mengajar. Sebab keberadaanya secara langsung dapat memberikan dinamika tersendiri terhadap peserta didik.

Association for Education and Communication Technology (AET) mendefinisikan media yaitu segala bentuk yang dipergunakan untuk suatu proses penyaluran informasi. Sedangkan *Education Association (NEA)* mendefinisikan sebagai benda yand dapat dimanipulasi, dilihat, didengar, dibaca atau dibicarakan beserta instrument yang dipergunakan dengan baik dalam kegiatan belajar mengajar, dapat mempengaruhi efektifitas program instruksional.

Media pembelajaran adalah alat, metode dan teknik yang digunakan dalam rangka lebih mengefektifkan komunikasi dan interaksi antara guru dan siswa dalam proses pendidikan dan pengajaran di sekolah [3]. Pada penelitian kali in peneliti lebih cenderung menggunakan definisi ini, karena cakupannya lebih luas, tidak hanya dibatasi sebagai alat tetapi juga teknik dan metode.

2.2.5.8 Pembelajaran Berbantuan Komputer

Media pembelajaran berbasis komputer atau biasa disebut pembelajaran berbantuan komputer (*Computer Assisted Instruction / CAI*) adalah salah satu media pembelajaran yang sangat menarik dan mampu meningkatkan motivasi belajar peserta didik. Penggunaan komputer sebagai media pembelajaran interaktif dapat diwujudkan dalam berbagai bentuk, diantaranya *Computer Assisted Learning* (CAL), konferensi komputer, surat elektronik atau elektronik mail (*e-mail*) dan komputer multimedia yang kemudian disebut multimedia pembelajaran interaktif. Pembelajaran melalui *CAI* ini bersifat *off-line* sehingga dalam penggunaan tidak tergantung pada adanya akses internet [4].

Teknologi yang berkembang pesat saat ini telah memungkinkan komputer memuat dan menayangkan beragam bentuk media di dalamnya. Penggunaan multimedia interaktif berbasis komputer dapat meningkatkan ekspresi diri, menciptakan suasana yang aktif sehingga pelajar dapat terlibat langsung dan multimedia dapat menjembatani komunikasi pelajar dengan pengajarnya. Multimedia berbasis komputer dapat pula dimanfaatkan sebagai sarana dalam melakukan simulasi untuk melatih keterampilan dan kompetensi tertentu.

Multimedia didefinisikan oleh Haffost Feldmans yang dikutip oleh Rusman sebagai suatu sistem komputer yang terdiri dari hardware dan software yang memberikan kemudahan untuk menggabungkan gambar, video, fotografi, grafik dan animasi dengan suara, teks, dan data yang dikendalikan dengan program computer[4]. Menurut Luther seperti dikutip oleh Sutopo pengembangan pembelajaran interaktif berbantuan komputer berbasis multimedia meliputi beberapa tahap sebagai berikut [4] :

1. Konsep (*Consept*)

   Pada tahap ini dilakukan identifikasi perkiraan kebutuhan yang dihasilkan dari tahap pengamatan pada penelitian awal.

2. Rancangan (*Design*)

   Pada tahap ini dibuat desain visual tampilan screen, interface, script atau cerita, story board dan struktur navigasi.

3. Pengumpulan Materi (*Collecting Content Material*)

   Pada tahap ini dilakukan pengumpulan bahan seperti gambar, animasi, audio dan video berikut pembuatan gambar, grafik dan hal-hal lain yang diperlukan untuk tahapan berikutnya.

4. Perakitan (*Assembly*)

   Pada tahap ini dilakukan pembuatan ilustrasi, audio dan video, serta pembuatan aplikasi berdasarkan story board dan struktur navigasi yang berasal dari tahap design. Dalam tahap ini juga dilakukan pembuatan program.

5. Pengujian (*Testing*)

Pada tahap pengembangan multimedia perlu dilakukan testing (uji coba) setelah produksi.

6. Distribusi (*Distribution*)
   Pada tahap ini dilakukan pembuatan pedoman penggunaan model pembelajaran, kemasan dan dokumentasi

2.2.5.9 Manfaat media pembelajaran

Manfaat positif dari penggunaan media sebagai bagian integral pengajaran di kelas adalah sebagai berikut: a). Penyampaian pelajaran menjadi lebih baku. Setiap pelajar yang melihat atau mendengar penyajian melalui media menerima pesan yang sama; b).Proses pembelajaran menjadi lebih menarik. Media dapat diasosiasikan sebagai penarik perhatian dan membuat siswa tetap terjaga dan memperhatikan; c). Pembelajaran menjadi lebih interaktif dengan diterapkannya teori belajar dan prinsip-prinsip psikologis yang diterima dalam hal partisipasi siswa, umpan balik, dan penguatan; d). Lama waktu pengajaran yang diperlukan dapat dipersingkat untuk mengantarkan pesan-pesan dan isi pelajaran dalam jumlah yang cukup banyak dan kemungkinannya dapat diserap oleh siswa. e). Kualitas hasil belajar dapat ditingkatkan f). Pengajaran dapat diberikan kapanpun dan dimanapun; g). Sikap positif siswa terhadap apa yang mereka pelajari dan terhadap proses belajar dapat ditingkatkan. h). Peran guru dapat berubah kearah yang lebih positif,dalam proses belajar mengajar.

2.2.5.10 Computer Aided Instruction

Aplikasi teknologi berbasis computer dalam pembelajaran dikenal sebagai *Computer Assisted Instruction* (CAI). Computer Assisted Instruction merupakan pengembangan daripada teknologi informasi terpadu yaitu komunikasi (interaktif), audio, video, penampilan citra (image) yang dikemas dengan sebutan teknologi multimedia. *Computer Assisted Instruction* mencakup penggunaan computer yang berhubungan dengan siswa maupun pendidik [6]. Dalam hal ini CAI dapat digunakan untuk mengajar dan melatih dalam mempelajari disiplin ilmu. Adapun model yang terdapat dalam Computer Assisted Instruction sebagai berikut :

1. *Model Drill and Practice*

Metode *drills* dalam CAI pada dasarnya merupakan salah satu strategi pembelajaran yang bertujuan memberikan pengalaman belajar yang lebih konkret melalui penciptaan tiruan-tiruan bentuk pengalaman yang mendekati suasana yang sebenarnya. Hal ini sesuai dengan yang dikemukakan metode *DRILLS* dalam CAI, yang pada dasarnya merupakan salah satu strategi pembelajaran yang bertujuan memberikan pengalaman belajar yang lebih konkret melalui penyediaan latihan-latihan soal yang bertujuan untuk menguji penampilan siswa melalui kecepatan meyelesaikan soal-soal latihan yang diberikan program.

2. *Model Tutorial*

Program CAI model tutorial merupakan program pembelajaran individual yang dikemas dalam bentuk pembelajaran berprogram bentuk branching. Program pembelajaran ini dikembangkan dengan menggunakan perangkat lunak *macromedia flash, swish max, director, articulate* dan sejenisnya. Sistem computer dapat menyampaikan pembelajaran secara langsung kepada para siswa melalui cara berinteraksi dengan mata pelajaran yang diprogramkan ke dalam sistem[5]. Pengajaran dengan bantuan computer khususnya model tutorial dapat mengatasi berbagai masalah sebagai berikut :

a. Lemahnya siswa dalam melakukan investigasi dalam memperkaya sumber bacaan secara *online*
b. Terlalu banyak *sequence* materi kadang membosankan siswa untuk membacanya secara tuntas
c. Pembelajaran yang disampaikan guru kadang monoton
d. Prosedur variasi pembelajaran yang dikembangkan secara manual masih memberikan kesan kurang memotovasi siswa

3. Simulasi

Model simulasi pada dasarnya merupakan salah satu strategi pembelajran yang bertujuan memberikan pengalaman belajar yang lebih kongkrit melalui penciptaan tiruan-tiruan bentuk pengalaman yang mendekati suasana sebenarnya dan berlangsung dalam suasana yang tanpa risiko. Model simulasi ini terbagi ke

dalam empat kategori model, yaitu fisik, situasi, prosedur dan proses dimana masing-masing kategori tersebutdigunakan sesuai dengan kepentingan tertentu. Model ini dapat merefleksikan perilaku belajar siswa khususnya dalam melakukan tahapan *information processing*, mulai dari menerima, mengolah, mentransformasikan dan memproduksi pesan-pesan baru yang dikeluarkan dalam bentuk perilaku atau verbal.

Secara umum, tahapan materi program CAI model simulasi adalah sebagai berikut :

a. Pengenalan
b. Penyajian informasi
   - Simulasi 1
   - Simulasi 2.
c. Penutup

Simulasi berbasis computer berorientasi pada upaya dalam memberikan pengalaman nyata kepada siswa melalui peniruan susasana. Secara sederhana pola-pola pengoperasiannya sebagai berikut :

a. Komputer menyajikan materi simulasi
b. Siswa menyimak proses simulasi materi pelajaran
c. Melanjutkan atau mengulangi tahapan simulasi sebelumnya

4. *Instructional Games*

Program CAI model *Games* atau permainan merupakan program pembelajaran yang lebih menekankan pada penyajian bentuk-bentuk permainan dengan muatan bahan pelajaran di dalamnya. Model games membuat peserta didik terlibat dalam situasi menang atau kalah yang meminta mempraktikan kemampuan untuk mengetahui atau dalam proses perkembangan. Penekanan games ini teletak pada upaya memaksimalkan aktivitas belajar mengajar secara berkesinambungan dalam bentuk interaksi antara peserta didik dan materi pelajaran dalam bentuk permainan.

Permainan yang didesain biasanya menunjukan kategori belajar melatih kecerdasan visual, kinestetik dan atau auditif. Efek CAI model games ini memapu memberikan kondisi yang lebih rileks yang dirasakan siswa ketika belajar. Dengan kondisi ini, siswa tidak akan mengalamai kelelahan belajar karena materi yang disajikan dalam model games ini dalah betul-betul ubentuk permainan seperti halnya games yang banyak ditemui di rental-rental sekarang ini. Model ini dikenalkan kepada para pendidik guna mengatasi berbagai masalah sebagai berikut :

a. Keterbatasan jam pelajaran, sehingga siswa sulit memperdalam penjelasan guru di sekolahnya secara menarik dalam bentuk pengemasan materi tertentu
b. Suasana untuk siswa berkonsultasi dengan guru yang cendereung kaku dan sungkan mengenai materi pelajaran selama kegiatan belajar di kelas
c. Rasio guru dan siswa berkonsultasi dengan guru yang tidak seimbang sehingga menyulitkan siswa dan guru berinteraksi.

Metode *games* termasuk kedalam kelompok pembelajaran berbasis computer dengan pola pengajaran tipe branching, dimana informasi/mata pelajaran disajikandalam bentuk-bentuk permainan, lalu disusul dengan pertanyaan dan atau respons atas keberhasilan atau kegagalan menuntaskan permainan yang dimaksud. Respon siswa dianalisis oleh computer lalu dibandingkan dengan jawaban yang kemudian diberi respon. Program pembelajaran ini juga menuntut siswa untuk mengaplikasikan ide dan pengetahuna yang dimilikinya secara langsung dalam kegiatan permainan yang berhubungan dengan materi-materi pelajaran tertentu. Adapun tahapan pembelajaran dengan bantuan komputer model games adalah sebagai berikut :

a. *Presentation of information* (Penyajian informasi)
b. *Playing Instructional Games* (Mulai permainan pembelajaran)
c. *Judging of Responses* (Penilaian respon)
d. *Providing Feedback About Responses* (Pemberian balikan respons)
e. *Remediation* (Pengulangan)
f. *Check Score* (Melihat nilai)

g. *Exit* (Keluar)

2.2.6 Konsep Dasar Basis Data

Basis data (*database*) dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis data menunjukan suatu kumpulan data yang dipakai dalam suatu lingkungan perusahaan atau intansi-instansi. Perancangan basis data dalam sistem informasi disebut sistem basis data (*database system*).

2.2.6.7 Definisi Basis Data

Istilah basis data banyak menimbulkan interpretasi yang berbeda. Anthoni J. Fabbri dan A. Roberth Schwab mendefinisikan basis data sebagai berikut:

"Basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data."

Menurut George Tsu-der Chou, basis data dapat didefinisikan sebagai berikut: "Basis data sebagai kumpulan informasi bermanfaat yang diorganisasikan ke dalam tata cara yang khusus."

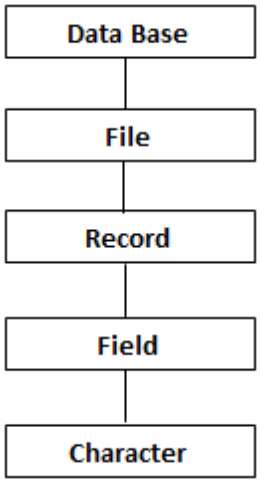
Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. Sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan computer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi / perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan. Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien.

Mengapa diperlukan database?

1. Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
2. Menentukan kualitas informasi akurat, tepat pada waktunya, dan relevan.
3. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
4. Mengurangi duplikasi data (*data redudancy*).

5. Hubungan data dapat ditingkatkan.
6. Mengurangi pemborosan tempat penyimpanan luar.

Sampai dengan membentuk suatu *database*, data mempunyai jenjang mulai dari karakter-karakter, *item data*, *record*, *file* dan kemudian *database*. Jenjang data dapat digambarkan pada gambar 2.3:



Gambar 2. 3 Jenjang Data

1. *Character*: merupakan bagian data yang terkecil, dapat berupa karakter *numeric*, huruf ataupun karakter-karakter khusus (*special character*) membentuk suatu *field*.
2. *Field*: Merepresentasikan suatu atribut dari *record* yang menunjukkan suatu *item* dari *data*, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari *field* membentuk *record*.
3. *File*: *File* terdiri dari *record-record* yang menggambarkan suatu kesatuan data yang sejenis.
4. *Database*: Kumpulan dari *file* yang membentuk suatu *database*.

2.2.6.8 Database Management System (DBMS)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak atau sistem yang khusu atau spesifik. Sistem ini yang akan menentukan bagaimana data diorganisasi, disimpan, diubah, dan diambil kembali. Di samping itu sistem ini juga

menerapkan mekanisme pengaman data, dan pemakaian data secara bersama, pemaksaan keakuratan atau konsistensi data dan sebagainya.

Perangkat lunak yang termasuk DBMS misalnya MY-SQL-server, dBase II+, dBase IV, FoxBase, RBase, MS-Access dan Borland-Paradox atau Borland Interbase, MS-SQL-Server, CA-Open Ingres, Oracle, Informix dan Sybase.

2.2.6.9 Tujuan Basis Data

Basis data pada prinsipnya ditujukan untuk pengaturan data agar terdapat kemudahan dalam pengambilan kembali data tersebut. Berikut ini terdapat beberapa tujuan dari basis data diantaranya yaitu:

1. Kecepatan dan kemudahan (*Speed*).
2. Efisiensi ruang penyimpanan (*Space*).
3. Keakuratan (*Accuracy*).
4. Ketersediaan (*Avaibility*).
5. Kelengkapan (*Completeness*).
6. Keamanan (*Security*).
7. Kebersamaan (*Sharability*).

2.2.6.10 Tahap Perancangan Basis Data

Perancangan basis data merupakan langkah untuk menentukan basis data, yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Perancangan basis data terdiri atas perancangan basis data secara konseptual, perancangan basis data secara fisik.

Perancangan basis data dibutuhkan agar diperoleh suatu sistem yang sesuai dengan apa yang diinginkan, di mana dapat melalui tahapan berikut:

a. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau biasa dikenal dengan diagram ER, secara grafis menggambarkan isi sebuah *database*. Diagram ini memiliki dua komponen utama yaitu entitas dan relasi. Untuk melambangkan fungsi di atas maka digunakan simbol-simbol yang bisa dilihat pada daftar simbol.

1. Entitas

Pada ER diagram, entitas digambarkan dengan sebuah bentuk pesegi panjang. Entitas adalah suatu apa saja yang ada di dalam sistem nyata maupun abstrak di mana data tersimpan. Entitas diberi nama dengan kata benda dan dapat dikelompokan dalam 4 jenis nama, yaitu: orang, benda, lokasi kejadian (terdapat unsur waktu di dalamnya).

2. Relasi

   Pada ER diagram, relasi dapat digambarkan dengan sebuah bentuk belah ketupat. Relasi adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (relationship) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau dengan kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan 2 bentuk empat persegi panjang.

3. Derajat Relasi

   Derajat Relasi atau *Relationship Degree* adalah jumlah entitas yang berpartisipasi dalam suatu relasi.

4. Atribut

   Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap relasi. Maksudnya atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun relasi, sehingga sering dikatakan bahwa atribut adalah elemen dari tiap entitas dan relasi.

5. Kardinalitas

   Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat 3 macam kardinalitas relasi, yaitu:

   a. One to One

      Tingkat hubungan satu ke satu dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

b. One to Many / Many to One

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. Many to Many

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisientitas yang petama maupun dari sisi yang kedua.

2.3 Metode Analisis Sistem

Model analisis yang digunakan dalam pengembangan aplikasi ini adalah analisis berorientasi object. Adapun *tools* pemodelan dalam analisis terstruktur akan dijelaskan sebagai berikut :

2.3.5 OOP(*Object Oriented Programming*)

Object Oriented Programming atau yang sering kita dengar dengan OOP, merupakan salah satu bahasa pemrogaman yang menggunakan konsep Object. OOP lebih digemari oleh para programmer karena lebih mudah dan fleksibel disbanding dengan bahasa pemrogaman imperative seperti pascal, cobol, dll. Didalam jenis bahasa ini, fungsi dan data menjadi satu kesatuan yang disebuat dengan Objek. Object Oriented Programming memiliki sebuah cara pandang dimana sebuah program bukanlah urut-urutan instruksi tapi diselesaikan oleh objek-objek yang saling bekerja sama untuk menyelesaikan suatu masalah. OOP, memang memiliki banyak kelebihan dan keuntungan. Misalnya saja OOP dapat mengurangi terjadinya Bug, kemudian OOP juga dapat meningkatkan kemampuan kita untuk membuat program yang lebih besar dan lebih ruwet. Hal itu dapat terjadi dikarenakan OOP memiliki konsep Encapsulation, Independensi, Modular, Object, Reusable,

Information Hiding dan lain sebagainya. Tak hanya itu, OOP juga memiliki kelebihan lain, yaitu :

a. Alami (*Natural*)

b. Dapat diandalkan (*Reliable*)

c. Dapat digunakan kembali (*Re-usable*)

d. Mudah untuk di maintain (*Maintainable*)

e. Dapat diperluas (*Extendable*)

f. Efisiensi waktu

Selain itu, OOP memiliki tujuan, tujuan tersebut adalah untuk memudahkan user dalam melakukan pengkodean dan pencarian error, misalnya saja dalam pembuatan program untuk membuat robot yang dilakukan oleh satu tim yang berisi enam orang. Kemudian robot itu sendiri dibagi menjadi enam bagian, dimana tiap bagiannya dikerjakan oleh satu orang. Ketika enam orang tersebut telah selesai maka bagian-bagian tersebut hanya tinggal digabungkan menjadi satu kesatuan, dan apabila terjadi error maka akan gampang terdekteksi, dimanakah letak objek yang rusah atau error.

a. Objek

Untuk mempermudah pemahaman, maka disini akan dijelaskan melalui analogi. Pada dasarnya semua benda yang adadi dunia nyata dapat dianggap sebagai objek. Misalnya rumah, mobil, sepeda, motor, gelas, komputer, meja, sepatu, dll. Setiap objek memiliki atribut sebagai status (state) dan tingkah laku sebagai behavior.

Contoh objek : Motor. Maka atribute (state) nya adalah pedal, roda, jeruji, speedometer, warna, jumlah roda. Sedangkan tingkah laku (*behavior*) nya adalah kecepatan menaik, kecepatan menurun, dan perpindahan gigi motor.

Analogi pemrograman berorientasi objek sama dengan penggambara pada dunia nyata seperti contoh di atas. Dalam OOP, state disimpan pada variabel dan tingkah laku disimpan pada method.

Dalam bahasa teoretis OOP, Objek berfungsi untuk membungkus data dan fungsi bersama menjadi satu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

b. Class

Definisi class yaitu template untuk membuat objek. Class merupakan prototipe atau blue prints yang mendefinisikan variabel-variabel dan methodmethod secara umum. Objek merupakan hasil instansiasi dari suatu class. Proses pembentukan objek dari suatu kelas disebut sebagai instantiation. Objek disebut juga sebagai instances.

Dalam bahasa teoretis OOP, class merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh ‘class of dog’ adalah suatu unit yang terdiri atas definisi-definisi data dan fungsifungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object.

Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang nonprogrammer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

c. Attributes

Atribut adalah data yang membedakan antara objek satu dengan yang lainnya.

Contoh Objek : VolcanoRobot (a volcanic exploration vehicle), mempunyai atribut sebagai berikut:

a) Status ~> exploring, moving, returning home

b) Speed ~> in miles per hour

c) Temperature ~> in Fahrenheit degrees

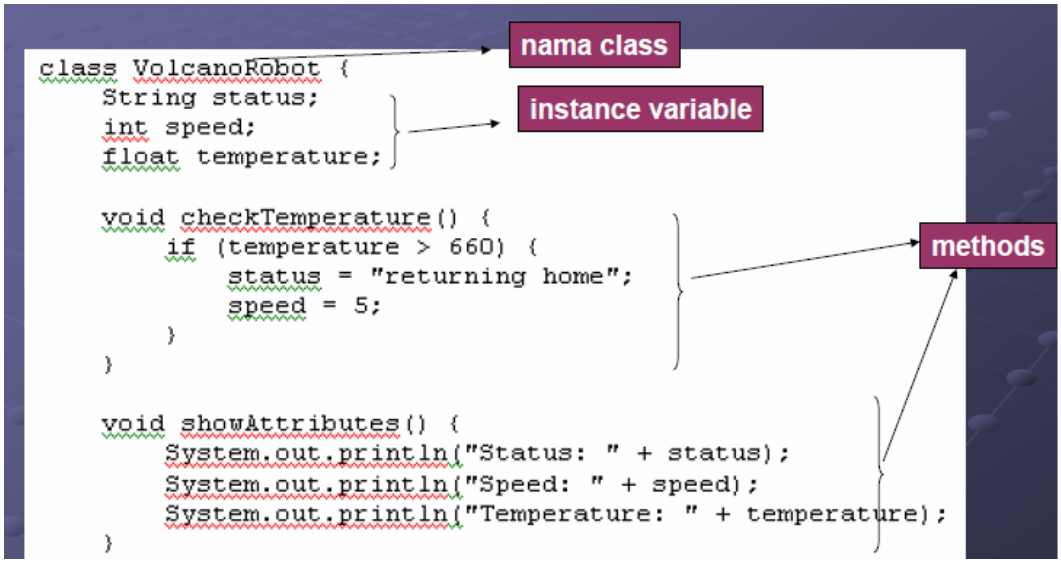
Dalam class, atribut sering disebut sebagai variabel. Atribut dibedakan menjadi dua jenis yaitu Instance Variable dan Class Variable. Instance variable adalah atribut untuk tiap objek dari kelas yang sama. Tiap objek mempunyai dan menyimpan nilai atributnya sendiri. Jadi, tiap objek dari class yang sama boleh mempunyai nilai yang sama atau berbeda.

Class Variable adalah atribut untuk semua objek yang dibuat dari class yang sama. Semua objek mempunyai nilai atribut yang sama. Jadi semua objek dari class yang sama mempunyai hanya satu nilai yang value nya sama.

d. Behavior

Behavior/tingkah laku adalah hal-hal yang bisa dilakukan oleh objek dari suatu class. Behavior dapat digunakan untuk mengubah nilai atribut suatu objek, menerima informasi dari objek lain, dan mengirim informasi ke objek lain untuk melakukan suatu tugas (*task*).

Contoh behavior untuk objek VolcanoRobot:



Gambar 2. 4 Contoh Behavior

Dalam class, behavior disebut juga sebagai methods. Methods sendiri adalah serangkaian statements dalam suatu class yang menghandle suatu task tertentu. Cara objek berkomunikasi dengan objek yang lain adalah dengan menggunakan method.

Contoh : class, object, attributes, dan behavior.

e. Abstraksi

Abstraksi adalah kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

2.3.5.7 Konsep – Konssep OOP

Berikut adalah bebrapa konsep dari bagian object oriented programming antara lain:

a. Enkapsulasi (*Encapsulation*)

Definisi enkapsulasi: Pembungkusan variabel dan method dalam sebuah obyek yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut. Variabel dan method yang dimiliki oleh suatu objek, bisa ditentukan hak aksesnya. Dalam OOP, konsep enkapsulasi sebenarnya merupakan perluasan dari struktur dalam bahasa C.

Contoh: jam tangan. Dalam hal ini, penting sekali untuk mengetahui waktu, sedangkan cara jam mencatat waktu dengan baik antara jam bertenaga baterai atau bertenaga gerak tidaklah penting kita ketahui.

Dengan kata lain enkapsulasi berfungsi untuk memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam/dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

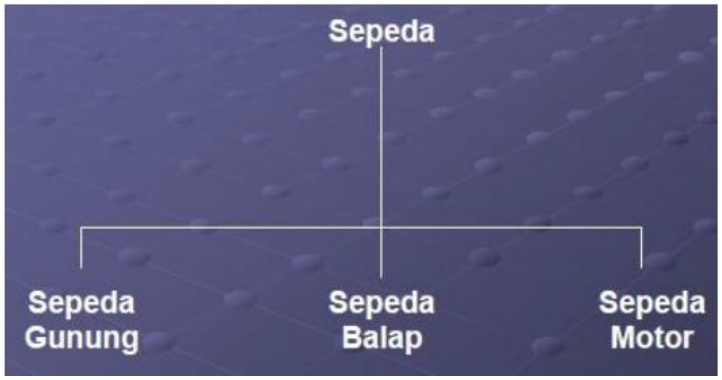
b. Pewarisan (*Inheritance*)

Pewarisan merupakan pewarisan atribut dan method dari sebuah class ke class lainnya. Class yang mewarisi disebut superclass dan Class yang diwarisi disebut subclass. Subclass bisa berlaku sebagai superclass bagi class lainya, disebut sebagai multilevel inheritance.

Contoh : terdapat class sepeda dan sepeda gunung. Sepeda termasuk superclass. Sepeda gunung termasuk subclass. Hal ini dikarenakan sepeda gunung memiliki variabel dan method yang dimiliki oleh sepeda.

Prinsip dasar inheritance yaitu persamaan-persamaan yang dimiliki oleh beberapa kelas dapat digabungkan dalam sebuah class induk sehingga setiap kelas yang diturunkannya memuat hal-hal yang spesifik untuk kelas yang bersangkutan.

Contoh pewarisan :



Gambar 2. 5 Contoh Pewarisan

c. Polimorfisme

Polimorfisme adalah kemampuan suatu obyek untuk mempunyai lebih dari satu bentuk. Polimorfisme tidak bergantung kepada pemanggilan subrutin. Metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim.

Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

d. Abstract Class dan Interface

Konsep akan abstract class, masih berkaitan dengan inheritance atau pewarisan karena disini, abstract class masih menggunakan super dan subclass. Jadi, Abstract Class itu sendiri merupakan kelas-kelas yang didefinisikan dengan

atribut dan method yang bersifat umum yang berlaku baik pada superclass maupun subclass.

Abstract class dideklarasikan dengan menggunakan keyword abstract dan class tersebut tidak dapat diinstantiasi seperti layaknya kelas-kelas lain. Selain itu abstract class biasanya digunakan untuk pengimplementasian frameworks. Dibawah ini ada sebuah contoh yang saya ambil dari sebuah web, yang mungkin dapat membantu kita dalam pembuatan dan pemahaman program dengan menggunakan abstract class.

Selanjutnya tentang interface, interface atau sering kita ketahui dengan antar muka bukanlah GUI (Graphics User Interface) yang memudahkan kita dalam menjalankan operasi, akan tetapi hal itu mungkin saja apabila kita mengimplementasikannya. Interface merupakan suatu class yang memiliki method atu bahkan sekumpulan method, akan tetapi method-method tersebut eblum memiliki isi atau perintah.

Berbeda dengan abstract class yang menggunakan keyword abstract untuk penggunaannya, interface menggunakan keyword interface diantara access modifier dan nama interface nya. Modifier di dalam interface juga tidak boleh menggunakan tipe static.

Berikut adalah perbedaan-perbedaan antara abstract class dan interface :

1. Interface dideklarasikan dengan keyword interface sedangkan abstract class dengan keyword abstract.
2. Interface tidak memiliki implementasi sedangkan abctract class dapat diimplementasikan
3. Interface hanya memiliki deklarasi method (method kosong) sedangkan abstract class tidak
4. Interface dapat mewariskan kepada lebih dari satu interface sedangkan abstract class dapat diimplementasikan kebanyak interface, namun hanya dapat satu yang di warisi.

5. Interface hanya bias digunakan bila implementasinya berubah sedangkan abstract method harus meng-override semua abstract method dan mungkin meng-override method-method virtual.
6. Interface membuat implementasi dapat dirubah-rubah sedangkan abstract class bias digunakan untuk menyediakan beberapa method default di class utama.
7. Interface meningkatkan keamanan dengan menyembunyikan implementasinya sedangkan abstract class dapat digunakan bila mengimplementasi framework.

2.3.6 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak,dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak.

Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch , metodologi coad , metodologi OOSE , metodologi OMT , metodologi shlaer-mellor ,metodologi wirfs-brock , dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain
yang menggunakan metodologi yang berlainan [4].

2.3.7 Konsep Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar 2.3.

Major Area	*View*	*Diagrams*	*Main Concepts*
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration role, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Gambar 2. 6 Konsep Dasar UML

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams*. *Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagaram tersebut.[5]

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

a. use case diagram
b. class diagram
c. statechart diagram
d. activity diagram
e. sequence diagram
f. collaboration diagram
g. component diagram
h. deployment diagram

2.3.7.7 Use Case Diagram

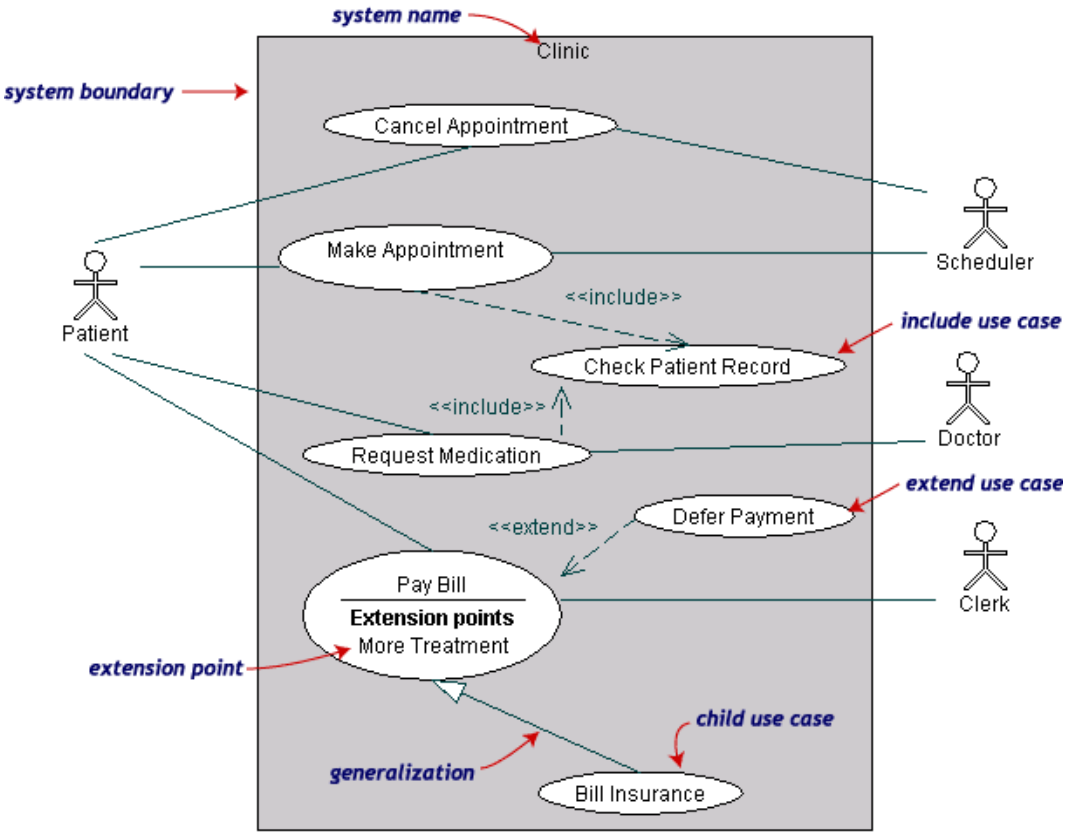
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login 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.

Contoh *use case diagram* :



Gambar 2. 7 Use Case Diagram

2.3.7.8 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

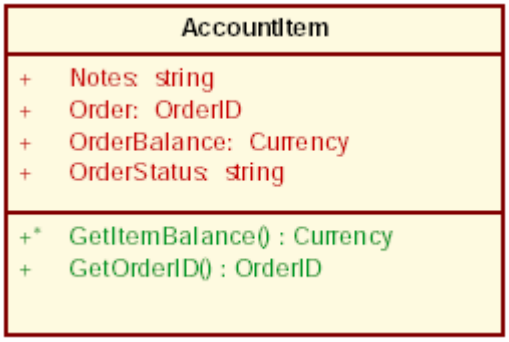
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a) Private, tidak dapat dipanggil dari luar class yang bersangkutan

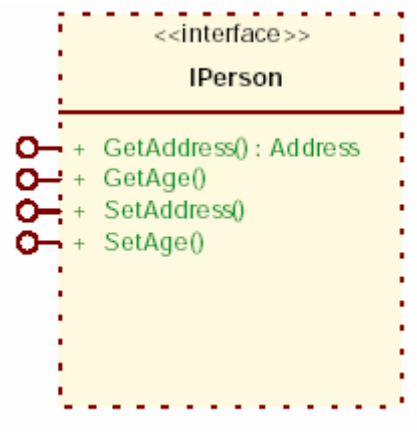
b) Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya

c) Public, dapat dipanggil oleh siapa saja



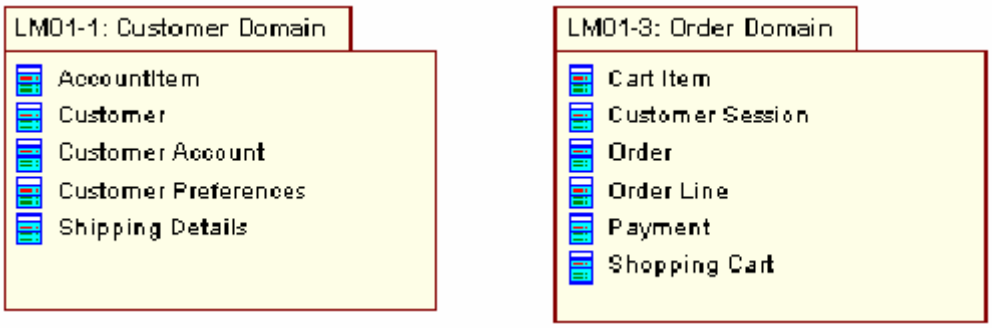
Gambar 2. 8 Atribut Class Diagram

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*.



Gambar 2. 9 Atribut Class Diagram

Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.



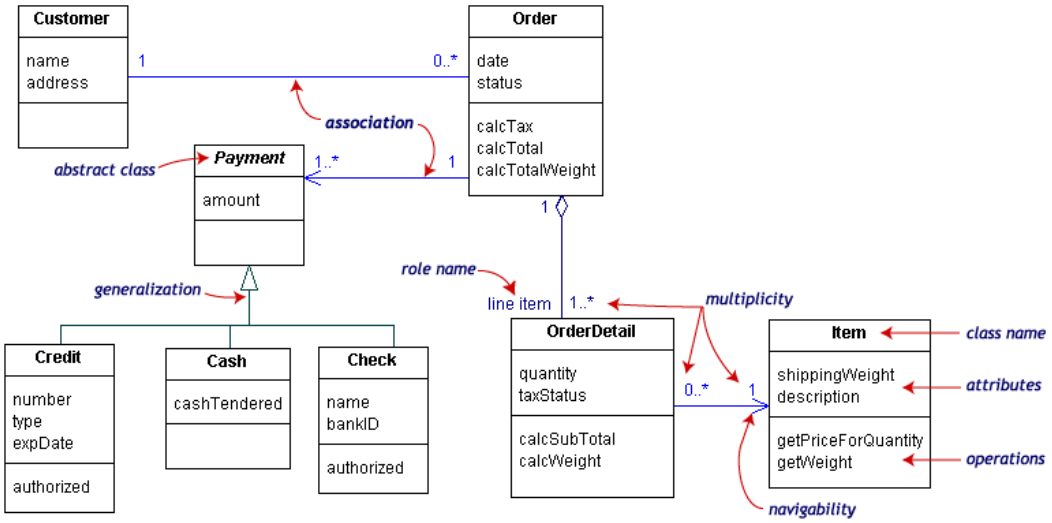
Gambar 2. 10 Atribut Class Diagram

Hubungan Antar Class :

A. 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.
B. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
C. 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.

D. 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.

Contoh class diagram :



Gambar 2. 11 Contoh Class Diagram

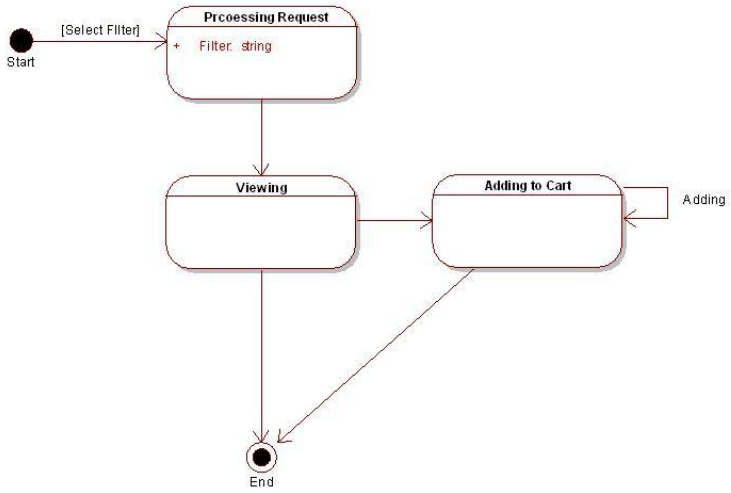
2.3.7.9 *Statechart* Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar state umumnya memiliki kondisi guard yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. Action yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

Contoh statechart diagram :



Gambar 2. 12 Contoh Statechart Diagram

2.3.7.10 Activty Diagram

Activity diagrams 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.

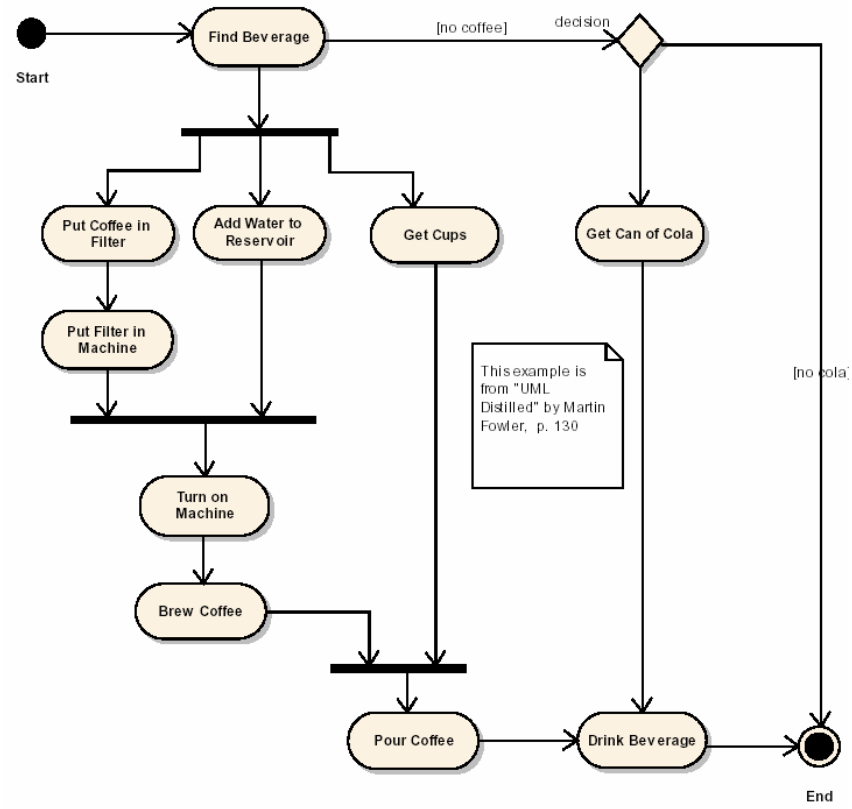
Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

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.

Contoh *activity diagram* tanpa *swimlane*:



Gambar 2. 13 Activity Diagram

2.3.7.11 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).

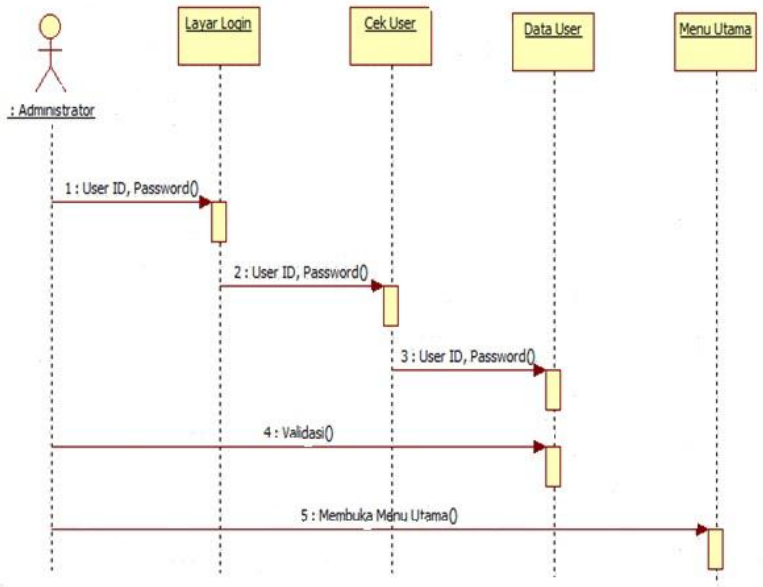
Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas

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*.

Contoh *sequence diagram* :

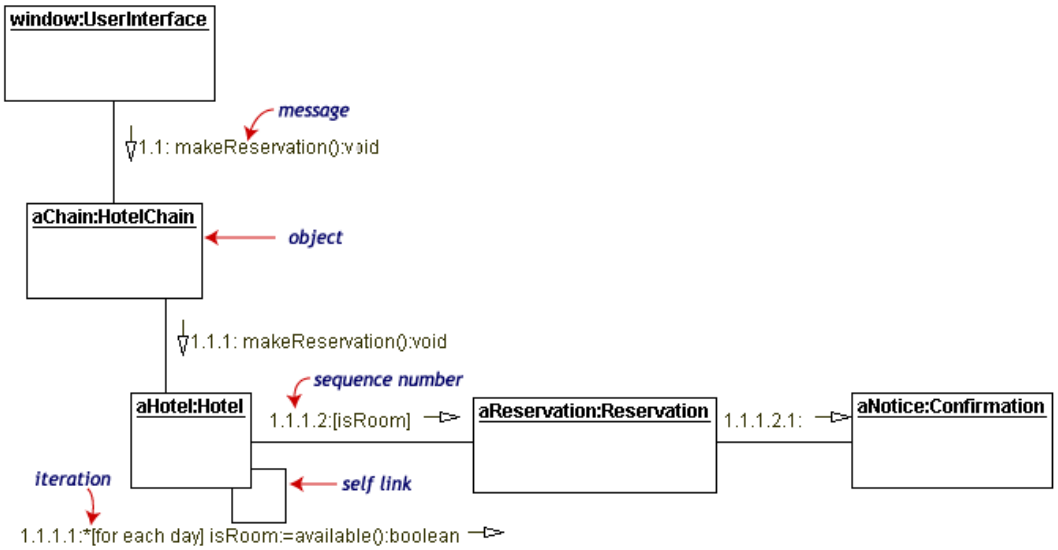


Gambar 2. 14 Contoh Sequance Diagram

2.3.7.12 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1.

Messages dari level yang sama memiliki prefiks yang sama.



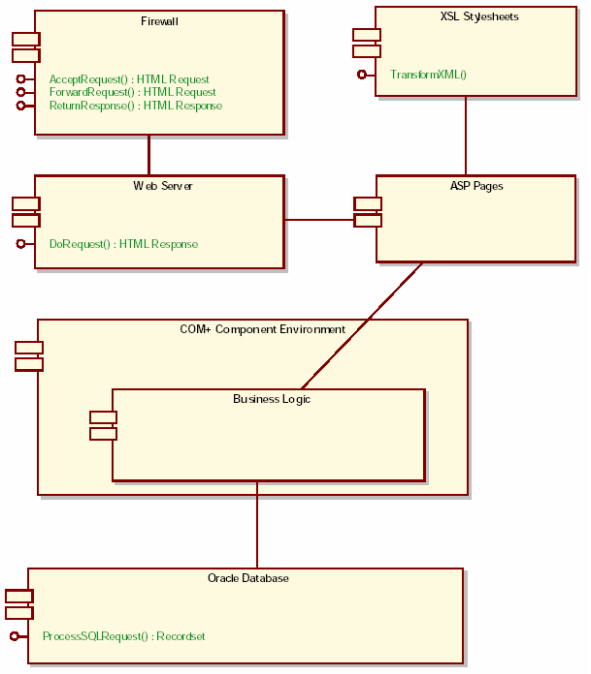
Gambar 2. 15 Contoh Collaboration Diagram

2.3.7.13 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya.

Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga darikomponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Contoh component diagram:

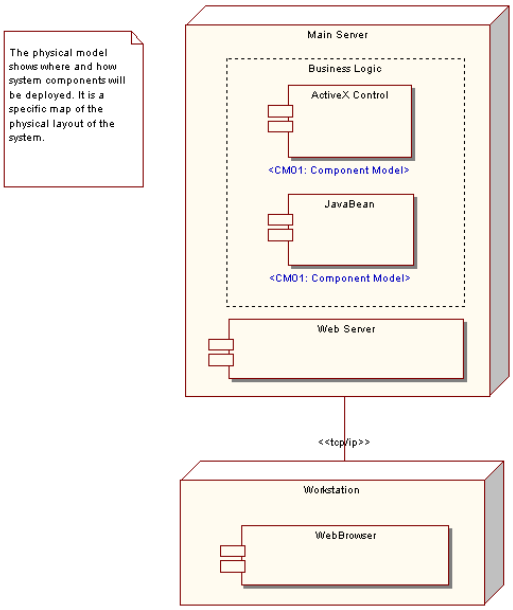


Gambar 2. 16 Contoh Component Diagram

2.3.7.14 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

Contoh *deployment diagram* :



Gambar 2. 17 Contoh Deployment Diagram

2.4 Software Pendukung Dalam Pengembangan Perangkat Lunak

Software yang digunakan dalam pembangunan aplikasi pembelajaran biologi dapat dipaparkan dibawah :

2.4.5 Pengertian Unity

Unity 3d adalah salah satu software yang bagus untuk mengembangkan game 3D dan selain itu juga merupakan software atau aplikasi yang interaktif dan atau dapat juga digunakan untuk membuat animasi 3 dimensi. Unity lebih tepat dijelaskan sebagai salah satu software untuk mengembangkan video game atau disebut juga game engine , yang sebanding dengan game engine yang lain contohnya saja: Director dan Torque game engine. Unity sebanding dengan mereka (Director dan Torque) dikarenakan mereka semua sama – sama menggunakan grafis yang digunakan untuk pengembangan aplikasi 3D.

Dalam beberapa tahun perkembangannya, sebelum dirilis, Unity telah diluncurkan pertama kali sebagai versi pra – rilis dengan GooBall sebuah video game yang didesain khusus untuk Apple Macintosh.

GooBall, dengan unity pra – rilis , telah diluncurkan atau diumumkan pada bulan Maret tahun 2005, sementara itu Unity diluncurkan secara resmi sebagai aplikasi yang bersifat komersial pada dua bulan setelahnya yaitu bulan Juni tahun 2005.Satu tahun kemudian yaitu tahun 2006, aplikasi pengembang game ini telah menjadi nominasi untuk Apple design awards dalam kategori "Best OS X Graphics".

Unity juga disebut sebagai aplikasi pengembang multiplatform, yang mana artinya unity mendukung untuk mengembangkan aplikasi game dan aplikasi yang lain untuk beberapa platforms seperti game console, Mobile Phone platforms, Windows dan OS X.

Sejak unity secara resmi dirilis sebagai Unity versi 1.0.1, banyak pembaharuan (update), upgrades dan fitur yang telah ditambahkan selama tahun perilisannya tersebut dan Unity terus berkembang secara terus – menerus. Sekarang ini Unity sudah ada pada versi 3.0 yang telah diumumkan pada bulan Maret 2010, bagaimanapun versi ini masih didalam pre order, yang mana artinya pengguna dapat memesan tetapi sebenarnya software masih belum rilis untuk pengiriman.

Unity ada atau datang dengan beberapa pilihan perijinan (license), berkisar dari yang gratis untuk Unity paket utama (dasar) untuk Unity pro, untuk satu license dijual dengan harga $1200. Baik Unity yang versi gratis dan Unity yang versi pro menawarka banyak fitur yang dapat digunakan, masih untuk yang versi gratis memperlihatkan sebuah halaman spash pada game yang tetap (regular) , dan desain untuk game online menggunakan Unity watermark.

Lebih dari beberapa tahun banyak game yang dikembangkan dan dibuat berjalan pada Unity, beberapa lebih ketetapan berada dalam satu bungkus atau masukkan: Butuh Kecepatan: Dunia, yang mana sekarang ini dalam perkembangan dan waktunya rilis berikutnya pada tahun ini, WolfQuest, yang mana rilis pada tahun 2007, Tiger Woods PGA Tour Online , yang mana telah dibuat pada April pada tahun 2007 dan Atmosphir , yang mana banyak game baru

yang dapat berjalan di Unity.Harus di catat akan dapat mengembangkan game untuk Iphone/Ipad platform, degan melakukan upgrade plug-in yang dibutuhkan oleh Uniy, untuk licensenya dapat diperoleh pada website Unity.

Fitur – fitur pada Unity antara lain:

1. *Rendering*

Graphics engine yang digunakan adalah Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS), dan proprietary APIs (Wii). Ada pula kemampuan untuk bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects.[4]

Unity dapat mengambil format desain dari 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Allegorithmic Substance. Asset tersebut dapat ditambahkan ke game project dan diatur melalui graphical user interface Unity.

ShaderLab adalah bahasa yang digunakan untuk shaders, dimana mampu memberikan deklaratif “programming” dari fixed-function pipeline dan program shader ditulis dalam GLSL atau Cg. Sebuah shader dapat menyertakan banyak varian dan sebuah spesifikasi fallback declarative, dimana membuat Unity dapat mendeteksi berbagai macam video card terbaik saat ini, dan jika tidak ada yang kompatibel, maka akan dilempar menggunakan shader alternatif yang mungkin dapat menurunkan fitur dan performa.

Pada 3 Agustus 2013, seiring dengan diluncurkannya versi 4.2, Unity mengijinkan developer indie mengunakan Realtime shadows hanya untuk Directional lights, dan juga menambahkan kemampuan dari DirectX11 yang memberikan shadows dengan resolusi pixel yang lebih sempurna, textur untuk membuat objek 3d dari grayscale dengan lebih grafik facial, animasi yang lebih halus dan mempercepat FPS.

2. *Scripting*

Script game engine dibuat dengan Mono 2.6, sebuah implementasi open-source dari .NET Framework. Programmer dapat menggunakan UnityScript (bahasa terkustomisasi yang terinspirasi dari sintax ECMAScript, dalam bentuk JavaScript), C#, atau Boo (terinspirasi dari sintax bahasa pemrograman phyton). Dimulai dengan dirilisnya versi 3.0, Unity menyertakan versi MonoDevelop yang terkustomisasi untuk debug script.

3. *Asset Tracking*

Unity juga menyertakan Server Unity Asset – sebuah solusi terkontrol untuk defeloper game asset dan script. Server tersebut menggunakan PostgreSQL sebagai backend, sistem audio dibuat menggunakan FMOD library (dengan kemampuan untuk memutar Ogg Vorbis compressed audio), video playback menggunakan Theora codec, engine daratan dan vegetasi (dimana mensuport tree billboarding, Occlusion Culling dengan Umbra), built-in lightmapping dan global illumination dengan Beast, multiplayer networking menggunakan RakNet, dan navigasi mesh pencari jalur built-in.

4. *Platforms*

Unity support pengembangan ke berbagai plaform. Didalam project, developer memiliki kontrol untuk mengirim keperangkat mobile, web browser, desktop, and console. Unity juga mengijinkan spesifikasi kompresi textur dan pengaturan resolusi di setiap platform yang didukung.

Saat ini platform yang didukung adalah BlackBerry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Web Player, Adobe Flash, PlayStation 3, Xbox 360, Wii U and Wii. Meskipun tidak semua terkonfirmasi secara resmi, Unity juga mendukung PlayStation Vita yang dapat dilihat pada game Escape Plan dan Oddworld: New ‘n’ Tasty.

Rencana platform berikutnya adalah PlayStation 4 dan Xbox One. Dan juga rumor untuk kedepanya mengatakan HTML akan menjadi platformnya, dan

plug-in Adobe baru dimana akan disubtitusikan ke Flash Player, juga akan menjadi platform berikutnya.

5. *Asset Store*

Diluncurkan November 2010, Unity Asset Store adalah sebuah resource yang hadir di Unity editor. Asset store terdiri dari koleksi lebih dari 4,400 asset packages, beserta 3D models, textures dan materials, sistem particle, musik dan efek suara, tutorial dan project, scripting package, editor extensions dan servis online.

6. *Physics*

Unity juga memiliki suport built-in untuk PhysX physics engine (sejak Unity 3.0) dari Nvidia (sebelumnya Ageia) dengan penambahan kemampuan untuk simulasi real-time cloth pada arbitrary dan skinned meshes, thick ray cast, dan collision layers.

2.4.6 Pengertian Blender

Pada tahun 1988 Ton Roosendaal mendanai perusahaan yang bergerak dibidang animasi yang dinamakan NeoGeo. NeoGeo adalah berkembang pesat sehingga menjadi perusahaan animasi terbesar di Belanda dan salah satu perusahaan animasi terdepan di Eropa. Ton Roosendaal selain bertanggung jawab sebagai art director juga bertanggung jawab atas pengembangan software internal.

Pada tahun 1995 muncullah sebuah software yang pada akhirnya dinamakan Blender. Setelah diamati ternyata Blender memiliki potensi untuk digunakan oleh artis –artis diluar NeoGeo. Lalu pada tahun 1998 Ton mendirikan perusahaan yang bernama Not a Number (NaN) Untuk mengembangkan dan memasarkan Blender lebih jauh. Cita – cita NaN adalah untuk menciptakan sebuah software animasi 3D yang padat, cross platform yang gratis dan dapat digunakan oleh masyarakat computer yang umum.

Sayangnya ambisi NaN tidak sesuai dengan kenyataan pasar saat itu. Tahun 2001 NaN dibentuk ulang menjadi perusahaan yang lebih kecil NaN lalu

meluncurkan software komersial pertamanya, Blender Publisher. Sasaran pasar software ini adalah untuk web 3D interaktif. Angka penjualan yang rendah dan iklim ekonomi yang tidak menguntungkan saat itu mengakibatkan NaN ditutup. Punutupan ini termasuk penghentian terhadap pengembangan Blender.

Karena tidak ingin Blender hilang ditelan waktu begitu saja, Ton Roosendaal mendirikan organisasi non profit yang bernama Blender Foundation. Tujuan utama Blender Foundation adalah tersu mempromosikan dan mengembangkan Blender sebagai proyek open source. Pada tahun 2002 Blender dirilis ulang dibawah syarat – syarat GNU General Public License.

Fitur – Fitur Blender ,antara lain:

Blender memiliki fitur sama kuat mengatur dalam lingkup dan kedalaman ke ujung lain tinggi 3D software seperti Softimage | XSI, Cinema 4D, 3ds Max dan Maya.

Perangkat lunak ini berisi fitur yang merupakan ciri khas dari model perangkat lunak high-end. Ini adalah Open Source yang paling populer grafis 3D aplikasi yang tersedia, dan merupakan salah satu yang paling didownload dengan lebih dari 200.000 download dari rilis masing-masing.

Fitur termasuk:

Model: Obyek 3D tipe, termasuk jerat poligon, permukaan NURBS, Bezier dan kurva B-spline; multiresolusi patung kemampuan; Modifier stack deformers; model Mesh; Python Scripting

Rigging: Skeleton kode ciptaan; Skinning; lapisan Bone; B-splines interpolated tulang

Animasi: animasi editor non-linear; Vertex framing kunci untuk morphing, animasi Karakter berpose editor; deformers animasi, pemutaran Audio; sistem kendala animasi

Rendering: raytracer inbuilt; oversampling, blor gerak, efek pasca produksi, ladang, non-square pixel, lapisan Render dan melewati; Render baking ke peta UV, Efek termasuk halo, suar lensa, kabut, vektor motion-blur proses pasca- , dan proses pasca-defocus; Ekspor naskah untuk penyaji eksternal

UV unwrapping: Laurent dan metode Berdasarkan Sudut unwrapping; unwreapping berdasarkan jahitan; falloff proporsional mengedit peta UV

Shading: membaur dan shader specular; Node editor; hamburan Bawah, shading Tangent; peta Refleksi

Fisika dan Partikel: sistem Partikel dapat dilampirkan ke mesh objek; simulator Fluida; solver Realtime tubuh lembut

Imaging dan Komposisi: multilayer OpenEXR dukungan; filter node komposit, konverter, warna dan operator vektor; 8 mendukung prosesor; sequencer realtime dekat; Bentuk gelombang dan U / V menyebar plits

Realtime 3D/Game Penciptaan: editor grafis logika; Bullet Fisika dukungan Perpustakaan; jenis Shape: polyhedron Convex, kotak, bola, kerucut, silinder, kapsul, majemuk, dan mesh segitiga statis dengan mode auto penonaktifan; tabrakan Diskrit; Dukungan untuk kendaraan dinamika; Mendukung semua modus pencahayaan OpenGL; Python scripting; Audio. Lintas Platfrom dengan GUI OpenGL seragam pada semua Platfrom ,siap untuk digunakan untuk semua versi windows (98, NT, 2000, XP), Linux,OS X , FreeBSD, Irix, SUN dan berbagai Sistem Oprasi lainnya. Kualitas tinggi arsitektur 3D yeng memungkinkan penciptaan cepat dan efisiens . Lebih dari 200.000 download (pengguna) dari seluruh dunia , Diekseskusi berukuran kecil, dan distribusi rendah.

2.4.7 Adobe Flash CS6

Adobe Flash Professional CS6 merupakan salah satu program animasi 2D *vektor* yang sangat digemari oleh para *Animator*. Dalam perkembangannya, program Adobe Flash Pro melakukan banyak penyempurnaan pada setiap versinya. Banyak fasilitas yang terdapat pada Adobe Flash Pro versi CS6 untuk membuat animasi 2D berbasis *vektor*.

Dengan Program Adobe Flash Professional CS6 kita dapat membuat berbagai aplikasi animasi 2D mulai dari animasi kartun, animasi interaktif, *game*, *company profile*, presentasi, *video clip*, *movie*, *web* animasi dan aplikasi animasi lainnya sesuai kebutuhan kita. Dan kini Flash didukung dengan kemampuannya dalam membuat animasi transformasi 3D, serta panel yang khusus untuk pembuatan animasi dengan *Actions Script* yang lebih mudah pemakaiannya.

Adobe Flash CS6 menyediakan berbagai macam fitur yang akan sangat membantu para animator untuk membuat animasi menjadi semakin mudah dan menarik. Adobe Flash CS6 telah mampu membuat dan mengolah teks maupun objek dengan efek tiga dimensi, sehingga hasilnya tampak lebih menarik [12].

2.5 Pengujian Aplikasi

Pengujian bertujuan untuk mencari kesalahan. Pengujian yang baik adalah pengujian yang memiliki kemungkinan besar dalam menentukan kesalahan [9]. Berikut adalah penjelasan mengenai pengujian :

2.5.5 *Black Box Testing* (Pengujian Kotak Hitam)

Pengujian *Black Box* digunakan untuk menguji fingsi-fungsi khusus dari perangkat lunak yang dirancang. Pada teknik ini, kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya. Beberapa jenis kesalahan yang dapat diidentifikasi yaitu :

a. Fungsi tidak benar
b. Kesalahan antarmuka
c. Kesalahan pada struktur data (pengaksesan basis data)
d. Kesalahan inisialisasi dan akhir program
e. Kesalahan performansi

Walaupun sulit untuk menulusuri kesalahan yang mungkin didapat, teknik pengujian *black box* lebih sering dipilih untuk menguji perangkat lunak karena kemudahan dalam pelaksanaannya.

2.5.6 Pengujian Beta

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuesioner mengenai tanggapan siswa terhadap aplikasi yang telah dibangun. Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari siswa. Pengujian *beta* terdiri dari *posttest, pretest,* dan kuesioner.

2.5.6.7 Skala Likert

Skala data yang digunakan untuk pengukuran variabel independen adalah skala *likert*. skala *likert* adalah skala yang digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Data yang telah terkumpul melalui angket, kemudian penulis olah kedalam bentuk kuantitatif, yaitu dengan cara menetapkan skor jawaban dari pertanyaan yang telah dijawab oleh responden, dimana pemberian skor tersebut didasarkan pada ketentuan Sugiyono menyatakan bahwa variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya [13]

Kuisioner ini digunakan untuk mengetahui jumlah jawaban dari para responden (siswa) melalui presentase, yaitu digunakan rumus sebagai berikut:

$$P = \frac{f}{n} \text{ x } 100\ \%$$

Keterangan:

P : Presentase

f : Frekuensi dari setiap jawaban angket

n : Jumlah responden

100 : Bilangan tetap