BAB 2
LANDASAN TEORI

2.1 Pengertian dan Batasan Batik

Indonesia merupakan negara yang memiliki berbagai macam kesenian dan budaya.Seni budaya batik khususnya sudah ada sejak zaman dahulu, dan hingga saat ini batik telah berkembang dan merupakan karya budaya nasional. Batik adalah salah satu seni budaya yang bersifat khusus, yaitu perpaduan antara seni dan tehnologi, dan batik pada umumnya merupakan karya seni yang memadukan antara seni motif atau ragam hias dan seni warna yang diproses melalui pencelupan dan penglorotan.

Sesuai dengan perkembangan ilmu pengetahuan serta pendidikan dibidang seni, sejalan pula dengan perkembangan seni batik baik itu dalam segi warna, gaya, tehnik maupun motif-motif yang diterapkan.

Melihat prospek dan harapan yang bagus dari perkembangan seni batik, maka perlu diperhatikan hal-hal mendasar seperti kwalitas, bahan dasar (kain, obat-obat pewarna, dan lainnya), sistem pengerjaan serta pengembangan ide didalam penyajiannya sehingga memberi nilai tambah nilai jual dari segi ekonomi dan juga memberi dampak positif pada prospek jangka panjang seni batik itu sendiri [1].

2.1.1 Batik Menurut Pengertian Umum

Kain batik bagi bangsa Indonesia sudah dapat dikatakan sangat mendarah daging, karena batik khususnya di Jawa dipakai sebagai busana untuk keacara pesta kundangan, perkawinan atau upacara adat lainnya bahkan juga digunakan sehari-hari. Bahkan ada pepatah jika belum menggunakan busana batik maka belum dapat dikatakan seseorang itu berbusana yang baik, jadi pengertian dulu batik adalah kain yang dipakai untuk berbusana baik.

Pengertian umum dahulu secara sederhana, kain batik adalah kain-kain bermotif yang dipakai untuk ikat kepala, selendang, sarung, dan kemben. Sedangkan

pengertian umum sekarang batik adalah kain bermotif yang dipergunakan untuk kemeja, rok wanita, taplak meja, gorden, sprei dan sarung bantal.

Secara terperinci, batik Indonesia mempunyai ciri-ciri sebagai berikut : Cara pembuatannya menurut tehnik pencelupan rintang , Zat perintang adalah lilin batik dengan ramuan khusus, Motif batik mempunyai ciri khas Indonesia yang mana tersusun dari ornamen- ornamen yang memiliki pengertian, keindahan, arti simbolis yang sesuai dengan kepribadian bangsa indonesia [1].

2.1.2 Batik Menurut Departemen Perindustrian

Karena pengertian umum tentang batik masih bermacam-macam, maka departemen perindustrian membuat definisi sebagai berikut :

"Batik adalah kain tekstil hasil pewarnaan, pencelupan rintang menurut corak khas ciri batik indonesia, dengan menggunakan lilin batik sebagai zat perintang".

Dengan pengertian definisi batik seperti diatas, batik digolongkan menurut dua sistem [1]:

1. Penggolongan menurut cara pelekatan lilin batik yaitu batik tulis, batik cap dan batik lukis.
2. Penggolongan menurut cara proses penyelesaian batik, yaitu batik kerokan, batik lorodan, batik bedesan, batik radion dan batik remukan.

2.1.3 Batik Menurut Pengertian Internasioanl

Dahulu beberapa negara yaitu India, Thailand,Turkestan, Jepang mempunyai mempunyai hasil karya yang menyerupai kain batik yang di mana dibuat berdasarkan proses penceluban rintang. Demikian pula zat perintang dan jenis-jenis pewarna yang dipakai, motif dan namanya pun berbeda-beda pula. Di India hasil pencelupan rintangnya disebut kalamkari, di Thailand disebut phanung, di Turkentan disebut bhokara, dan di Jepang disebut rokechi. Sedangkan hasil pencelupan rintang dari Cina agak sedikit berbeda yaitu zat perintang yang digunakan terbuat dari getah tumbuhan dan selalu dicelup dengan warna biru dengan bahan dasar menggunakan kain sutra atau

disebut kain biru.

Berdasarkan perkembangan yang terjadi, karya celup rintang dari negara-negara tersebut diatas tidak berkembang bahkan ada yang sudah tidak dibuat lagi dan ada pula yang tinggal sisa-sisanya seperti kalamkari dari India. Berbeda halnya dengan di Indonesia batik dapat berkembang dari generasi kegenerasi baik itu dalam segi motif, warna maupun teknologinya [13].

2.1.4 Sejarah Batik

Seni batik diperkirakan sudah ada di Indonesia pada zaman kebudayaan logam atau kebudayaan perunggu, dan zaman ini dimulai kira-kira pada 500 tahun sebelum masehi. Hal ini dapat dilihat dari adanya kesamaan bentuk-bentuk hiasan motif-motif ornamen dengan motif batik, yaitu seperti [1] :

a) Garis-garis sejajar yang menyerupai sawud atau galaran atau rawan.
b) Garis-garis miring yang menyerupai dasar motif lerek atau lereng.
c) Lingkaran kecil-kecil yang menyerupai cecek-cecek atau titik-titik.
d) Garis-garis lengkung bersambung yang menyerupai pilin atau pilin berganda.
e) Segitiga berderet yang menyerupai motif pinggir untu walang atau motif tumpal.
f) Meander menyerupai motif pinggir awan batik klasik pantai utara Jawa (Cirebon).
g) Roset, seperti motif dasar dari motif-motif ceplok.
h) Planet, seperti dasar ornamen pohon hayat.
i) Swastika, seperti pada motif banji dalam batik.
j) Lingkaran seperti pada motif ceplok, nitik dan kawung.
k) Cecek sawut, seperti pada bentuk motif isen dalam motif batik.

Berdasarkan dari kesamaan bentuk-bentuk hiasan tersebut maka para ahli menyimpulkan bahwa di zaman itu dasar seni batik sudah dimiliki oleh bangsa

Indonesia sebelum mendapat pengaruh kebudayaan dari luar. Sesudah itu yaitu antara tahun 200 dan 300 sesudah Masehi terjadi perpindahan penduduk dari daerah Godawari dan Kalinga (Keling) di India, untuk gelombang pertama orang Hindu ke Indonesia yaitu ke Jawa Barat. Hal itu membuat pengetahuan rakyat setempat akan batik menjadi bertambah, antara lain hasil perbatikan mereka menjadi lebih halus.

Perkembangan motifnyapun mengalami perubahan-perubahan dan lebih mengacu pada ragam hias motif di candi dan arca, hingga sampai akhir kerajaan Majapahit pada tahun 1292 bentuk motif yang ada seperti :

a) Motif dengan titik-titik dan lingkaran seperti yang terdapat pada patung Padmapani dari abad ke-8 di Jawa Tengah.
b) Motif dengan ornamen bentuk lingkaran dan roset kecil yang terdapat pada patung Ganesa dari candi Banon dekat candi Borobudur pada abad ke-9.
c) Motif garis miring dengan deretan lingkaran pada bidang-bidang miring seperti dasar motif lerek, terdapat pada candi Dieng abad ke-9.
d) Dan banyak motif lainnya.

Macam-macam motif seni batik pada zaman kebudayaan Hindu ini cukup lengkap dan beraneka ragam bentuknya, dan mengalami perkembangan lagi setelah masuknya seni kebudayaan Islam yang banyak menonjolkan pada bentuk bangunan mesjid, yaitu bentuk kubah, menara,dan bentuk turbah. Perpaduan antara rasa dan pikiran cukup mendominasi dalam seni Islam, dan pada zaman ini terdapat perkembangan beberapa gaya motif seperti:

a) Motif gaya simbolis stiliran yang timbul pada waktu peralihan kebudayaan Hindu ke Islam.
b) Motif lung-lungan atau motif naturalistis adalah motif yang tersusun dari ornamen tumbuh-tumbuhan.Motif ini berkembang didaerah pantai utara Jawa, Madura, dan Bali.

c) Motif look-can yaitu motif yang terjadi karena pengaruh Cina seperti motif ornamen burung phoenix dan bentuk binatang atau tumbuhan dengan rumbai bergelombang.

Pada zaman Mataram tahun 1586-1654 seni batik dan kebudayaan lainnya terus berkembang dan menyebar keseluruh nusantara. Pada tahun 1646 yaitu pada zaman raja-raja Jawa, seni batik berkembang dalam dua arah yaitu pertama dalam kalangan Keraton dengan motif tetap bergaya simbolis stileran dan kedua berkembang dikalangan rakyat, terutama di kalangan rakyat daerah pantai dan bandar perdagangan dengan corak seni setempat seperti gaya look-can dan phoenix yang bersifat naturalis. Kerena potensi batik yang cukup besar, pada zaman pendudukan Belanda di Indonesia pengembangan dan pembinaan batik cukup maju. Namun itu dilakukan bukan untuk meningkatkan taraf hidup rakyat Indonesia pengrajin batik tapi semata-mata untuk meningkatkan perdagangan negeri Belanda seperti penyediaan kain putih atau mori untuk bahan batik yang dibuat dinegeri Belanda dan pengadaan zat pewarna batik yang dipasarkan di Indonesia, juga pemasaran batik-batik Indonesia yang dibeli dengan harga murah dan dipasarkan kenegara-negara lain dengan harga tinggi. Perkembangan motif batik cukup mengagumkan hingga saat ini [1].

2.1.5 Dasar Pemilihan Motif-motif Barat

Sebelum melangkah lebih jauh alangkah baiknya kita mengetahui terlebih dahulu pengertian tentang motif batik atau corak batik yaitu gambar pada kain batik yang berupa perpaduan antara garis, bentuk, dan isen menjadi satu kesatuan yang membentuk satu unit keindahan. Dan keindahan pada batik pada umumnya mempunyai arti atau simbol tertentu yang diciptakan oleh penciptanya, namun juga ada yang merupakan ciptaan biasa saja.

Dalam menerapkan motif-motif batik biasanya para pengrajin mengikuti patokan –patokan yang ada yang berlaku pada jamannya dan menurut penggolongan motifny amisalnya :

1. Motif sidomukti dapat berupa gambar sawah ladang yang mempunyai filosofi menggambarkan kemakmuran, binatang ternak yang menggambarkan raja kaya, kotak-kotak pematang sawah, binatang, candi, dan sayap yang menggambarkan rumah dan kewibawaan yang semuanya tersusun menurut bidang-bidang berbentuk belah ketupat, atau lebih dikenal dengan nama sidomukti sala yang biasa digunakan sebagai motif pada batik untuk penganten pada upacara perkawinan [1].
2. Motif nitik termasuk dalam motif batik kuno, yang tersusun dari garis putus-putus dan titik-titik sehingga sepintas lalu seperti motif anyaman, kain tenun atau kain sulam. Oleh sebab itu motif nitik disebut juga motif anyaman, dan dibuat dengan warna dasar hitam khusus daerah Yogya-Solo dan warna dasar biru, hijau untuk daerah Pekalongan.Kain dengan motif itu disebut kain jlamprang atau yang lebih dikenal dengan nama kain cinden, dan kain cinden ini dahulu dianggap keramat karena filosofi dari motif-motifnya yang dianggap memiliki kekuatan menolak bahaya. Kain ini biasanya digunakan sebagai pakaian, perlengkapan kraton, dan pakaian menari wayang wong [1].
3. Motif lerek atau lereng termasuk motif klasik yang hingga kini masih digunakan, dan terdiri dari motif parang dan motif liris.Motif parang tersusun dari deretan parang menurut garis miring dan variasi motif terletak pada bentuk parangnya, besar-kecil parangnya, dan isen pada motif. Motif parang yang paling terkenal adalah parang rusak karena motif ini dahulu hanya boleh digunakan pada pakaian keluarga keraton karena motif ini dianggap motif keturunan para raja dan dikeramatkan, sehingga orang biasa tidak boleh memakainya namun sekarang motif ini sudah boleh dipakai oleh siapa saja [14].
4. Motif semen yang berasal dari kata semi atau tumbuh yang menunjukkan jenis tumbuh- tumbuhan atau bagian tumbuhan seperti kuncup, daun bunga, bunga, tumbuhan merambat atau lung dan pohon hayat bentuk

stiliran, juga lidah api, balai atau candi dan garuda. Motif- motif tersebut mempunyai filosofi menggambarkan kebesaran alam yaitu bumi, api, air, dan angin [1].

5. Motif buketan berasal dari daerah utara pulau jawa seperti Lasem, Yuwono, Pekalongan, Cirebon dan Indramayu. Motif buketan menggambarkan tumbuhan dalam bentuk lengkap atau utuh, yaitu dengan mengisikan pangkal batang, batang, ranting, daun serta bunganya dan pada perwujudannya motif disusun berderet menurut panjang kain. Motif - motif tersebut disusun sedemikian rupa untuk menggambarkan bagaimana keinginan dari masyarakat atau penduduk pada waktu itu untuk memperoleh atau mendapatkan kesuburan pada tanaman- tanaman diladang atau dikebun mereka dan biasanya digunakan pada saat upacara menanam padi atau tanaman lainnya [14].
6. Motif terang bulan , terdapat didaerah Yogya, Solo, Pekalongan, Tasikmalaya, dan daerah Jawa barat. Motif ini berbentuk hiasan tumpal berupa bentuk stiliran yang menyerupai gapura Bali atau tugu, atau juga berupa gubahan pohon hayat menyerupai gunungan wayang kulit dan disusun berderet.Motif ini memberikan filosofi bagaimana kekuatan dan kejayaan kerajaan-kerajaan pada masa itu, dan kain batik dengan motif ini digunakan biasanya oleh para prajurit-prajurit kraton [1].
7. Motif peksi atau burung, jenis motif ini cukup banyak ragamnya seperti peksi dares, gagak, garuda, handon, huk, kablak, kingkin, kirana, kurung, kuwun, makuta, sikatan, dan urang-urangan. Motif-motif peksi ini cukup digemari dan berkembang didaerah Solo, Yogya, dan Pekalongan, disamping itu terus divariasi dan dikembangkan oleh para seniman-seniman batik yang ada hingga sampai saat ini, dan motif peksi ini diyakini dapat memberikan kemampuan memimpin bagi yang memakainya, khusus peksi makuta [14].

8. Motif nogo atau naga, jenis motif ini sudah agak jarang ditemui karena kurang begitu diminati.Motif nogo ada beberapa bentuknya seperti nogo bisikan atau naga raja, nogo pertolo, dan nogo puspa atau naga bunga yaitu ular yang kulitnya belang, dan batik dengan jenis motif ini dahulu berkembang didaerah Solo dan Yogya.

2.2 Pengertian Kebudayaan Menurut Bahasa dan Para Ahli

2.2.1 Pengertian Kebudayaan Menurut Bahasa

Kebudayaan berasal dari perkataan latin "Colere" yang berarti mengolah, mengerjakan, menyuburkan dan mengembangkan, terutama mengolah tanah atau bertani. Dari segi arti ini berkembanglah arti culture sebagai "segala daya dan aktivitas manusia untuk mengolah dan mengubah alam."

Menurut Bahasa Indonesia, kebudayaan berasal dari Bahasa Sansekerta "Buddhayah". Yaitu bentuk jamak dari buddhi yang berarti budi atau akal. Pendapat lain mengatakan, bahwa kata budaya adalah sebagai suatu perkembangan dari kata majemuk : budi daya, yang berarti daya dari budi. Karena itu mereka membedakan antara budaya dan kebudayaan. Budaya adalah daya dari budi yang berupa cipta, karsa dan rasa, sedangkan kebudayaan adalah hasil dari cipta, karsa dan rasa tersebut [15].

2.3 Pengertian Kebudayaan Menurut Para Ahli

Kebudayaan merupakan keseluruhan yang kompleks, yang didalamnya terkandung pengetahuan, kepercayaan, kesenian, moral, hukum, adat istiadat, dan kemampuan-kemampuan lain yang didapat oleh seseorang sebagai anggota masyarakat [16].

Kebudayaan mencakup keseluruhan yang meliputi bentuk teknologi social, ideologi, religi, dan kesenian serta benda, yang kesemuanya merupakan warisan social.

Kebudayaan adalah keseluruhan sistem gagasan, tindakan, dan hasil karya manusia dalam rangka kehidupan masyarakat yang dijadikan milik diri manusia dengan relajar [17].

Kebudayaan merupakan sistem gagasan yang menjadi pedoman dan pengarah

bagi manusia dalam bersikap dan berperilaku, baik secara individu maupun kelompok.

Kebudayaan adalah seperangkat peraturan dan norma yang dimiliki bersama oleh para anggota masyarakat, yang jika dilaksanakan oleh para anggotanya akan melahirkan perilaku yang dipandang layak dan dapat di tarima ole semua masyarakat.

Kebudayaan terdiri dari Pola-pola perilaku yang di hasilkan oleh interaksi social Semua perilaku dan semua produk yang dihasilkan oleh sesorang sebagai anggota suatu masyarakat yang di temukan melalui interaksi simbolis.

Kebudayaan adalah sesuatu yang terbentuk oleh pengembangan dan transmisi dari kepercayaan manusia melalui simbol-simbol tertentu, misalnya simbol bahasa sebagai rangkaian simbol yang digunakan untuk mengalihkan keyakinan budaya di antara para anggota suatu masyarakat. Pesan-pesan tentang kebudayaan yang di harapkan dapat di temukan di dalam media, pemerintahan, intitusi agama, sistem pendidikan dan semacam itu.

Kebudayaan adalah sebagian perulangan keseluruhan tindakan atau aktivitas manusia dan produk yang dihasilkan manusia yang telah memasyarakat secara sosial dan bukan sekedar di alihkan secara genetikal.

Kebudayaan adalah segala sesuatu yang di peroleh individu dari masyarakat, mencakup kepercayaan, adat istiadat, norma-norma artistic, kebiasaan makan, keahlian yang di peroleh bukan dari kreatifitasnya sendiri melainkan merupakan warisan masa lampau yang di dapat melalui pendidikan formal atau informal.

2.4 Game

Game berasal dari kata bahasa inggris yang berarti dasar permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*Intellectual Playability Game*) yang juga bisa diartikan sebagai arena keputusan dan aksi pemainnya.Dalam *game,* ada target-target yang ingin dicapai pemainnya.Permainan adalah kegiatan yang kompleks yang didalamnya terdapat peraturan, *play* dan budaya.Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau

buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan.

Game bertujuan untuk menghibur, *game* banyak disukai oleh anak-anak hingga orang dewasa. *Game* sebenarnya penting untuk perkembangan otak, untuk meningkatkan konsentrasi dan melatih untuk memecahkan masalah dengan tepat dan cepat karena dalam *game* terdapat berbagai konflik atau masalah yang menuntut kita untuk menyelesaikannya dengan cepat dan tepat. Tetapi *game* juga bisa merugikan karena apabila kita sudah kecanduan *game* kita akan lupa waktu dan akan mengganggu kegiatan atau aktivitas yang sedang kita lakukan [6].

2.4.1 Sejarah Game

Game pertama kali dibuat pada tahun 1966 oleh Ralph Baer bersama dengan timnya yang berjumlah 500 orang yang terdiri dari para insinyur danteknisi. Permainan yang pertama sekali ditemukan hanya dapat dimainkan oleh komputer seharga US$ 40.000. Dimana pengembangan dan pembuatan *game* tersebut didanai oleh Pentagon.

Pada tahun 1965, pihak militer datang kepada Baer dan meminta menciptakan simulasi komputer yang dapat membantu pasukan untuk belajar strategi dan mengukur

kemampuan reflek. Proyek ini dikerjakan dengan tingkat keamanan yang sangat ketat di tengah situasi perang dingin.

Setelah sebulan bekerja keras, Bear berhasil menampilkan dua titik putih yang berkejar-kejaran di layar. Hal ini membuat pihak militer merasa kagum dan memberikan dana yang lebih besar lagi sehingga ia dapat menyewa asisten lebih banyak. Tim ini kemudian berhasil membuat permainan antara papan dan bola yang pada akhir tahun 1966 dipresentasikan di depan para pejabat Pentagon. Namun sayang, ternyata mereka tak merasa tertarik dengan penemuan Bear. Hingga akhirnya ia berusaha memperoleh izin agar dapat memproduksi mesin permainan secara komersial.

Tahun 1970, Bill Enders yang tergabung dalam sebuah organisasi dengan nama Megnavox mencoba menyakinkan edukatif Megnavox untuk memberikan kesempatan pada Bear dan mesin permainannya. Hasilnya adalah munculnya *video game* komersial pertama, yaituMagnavox Odyssey.

Tahun 1972, muncul nama baru dalam dunia *game*. Nolan Bushnel. Pada tanggal 27 Juni 1972, ia mendirikan perusahaan Atari, dan membuat *game arcade* Pong. Era baru dalam perkembangan dunia *game* terjadi pada tahun 1988, yang di dominasi oleh perusahaan di Jepang. Nintendo, yang awalnya hanya memproduksi mesin fotocopi, beralih memproduksi *game* [7].

2.4.2 Jenis - Jenis Game

Berdasarkan jenisnya *game* dibagi menjadi beberapa kategori diantaranya: Berdasarkan *platfrom* atau alat yang digunakan [6] :

1. Arcade *game*s, yaitu video *game* dengan mesin yang memang khusus di desain untuk jenis video *game*s tertentu dan memiliki fitur seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil dan lain-lain.
2. PC *Game*s , yaitu video *game* yang dimainkan menggunakan Personal Computers.
3. Console *game*s, yaitu video *game*s yang dimainkan menggunakan console tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.

4. Handheld *game*s, yaitu yang dimainkan di console khusus video *game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
5. Mobile *game*s, yaitu yang dapat dimainkan atau khusus untuk *mobile phone* atau PDA.

2.4.3 Genre – Genre Game

Genre atau ragam permainan video digunakan untuk menggolongkan permainan video berdasarkan interaksi bidang permainannya, bukan hanya perbedaan visual maupun naratif. Berdasarkan genre *game* dibagi menjadi beberapa kategori diantaranya:

1. *Shooting*

   Shooting merupakan salah satu *genre* dalam *game* biasanya mensimulasikan tembak-tembakan , atau hajar-hajaran bisa juga tusuk-tusukan, tergantung cerita dan tokoh di dalamnya, *video game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari game jenis ini adalah tembak.

2. *Fighting*

   Fighting jenis ini memang memerlukan kecepatan refleks dan koordinasi mata-tangan, tetapi inti dari game ini adalah penguasaan *skill*karakter yang di mainkan, pengenalan karakter dan timing sangatlah penting, dan berbeda seperti game Aksi pada umumnya yang umumnya hanya melawan *Artificial Intellegence* atau melawan komputer saja, pemain jenis *fighting game* ini baru teruji kemampuan sesungguhnya dengan melawan pemain lainnya.

3. Petualangan

   Refleks dan kelihaian pemain dalam bergerak, berlari, melompat hingga memecut atau menembak tidak diperlukan di sini.*Video Game* murni petualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter hingga

penggunaan benda-benda tepat pada tempat yang tepat.termasuk didalamnya petualangan dengan teks atau sistem tunjuk dan klik.

4. Simulasi, Konstruksi dan manajemen

   Video Game jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Dari mencari jodoh dan pekerjaan, membangun rumah, gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. Dunia kehidupan rumah tangga sampai bisnis membangun konglomerasi, dari jualan limun pinggir jalan hingga membangun laboratorium cloning. *Video Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas.

5. *Role Playing*

   Video game jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya *level*, baik dari status kepintaran, kecepatan dan kekuatan karakter dan lain-lain.

6. Strategi

   Kebalikan dari *video game* jenis action yang berjalan cepat dan perlu refleks secepat kilat, *video game* jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. *Video game* strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusal pelatihan tempur, tergantung dari tema ceritanya. Pemain game strategi melihat dari sudut pandang lebih meluas dan lebih

kedepan dengan waktu permainan yang biasanya lebih lama dan santai dibandingkan game action. Unsur-unsur permainannya biasanya berkisar sekitar, prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumberdaya uang, besi, kayu, minyak dan lain-lain, hingga ke pembelian dan peng-upgrade-an pasukan atau teknologi.

7. Olahraga

   Biasanya *game genre* ini permainannya diusahakan serealistik mungkin di buat se mirip mungkin dengan aslinya walau kadang ada yang menambah unsur fiksi dalam *game* olahraga tersebut.

8. Edukasi

   Video Game jenis ini dibuat dengan tujuan spesifik sebagai alat pendidikan, baik itu untuk belajr mengenal warna, mengenal huruf dan angka, matematika, belajar bahasa asing dan lain-lain. *Developer* yang membuatnya, harus memperhitungkan berbagai hal agar game ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan desain visual ataupun animasinya.

2.4.4 Game Adventure

Adventure game atau permainan pertualangan adalah suatu permainan yang mengajak kita menjelajah kedalam permainan tersebut. Memberikan petualangan yang berbeda disetiap *level*nya. Biasanya didalam petualangan tersebut user atau karakter pemain tersebut diminta untuk mengumpulkan poin dengan cara mengumpulkan koin-koin.

2.4.5 Game Edukasi

Sesuai dengan arti bahasa Indonesia, *Game* berarti permainan, sedangkan edukasi adalah pendidikan.

Game edukasi adalah salah satu genre *game* yang digunakan untuk memberikan pengajaran atau menambah pengetahuan penggunanya melalui suatu media unik dan

menarik. Genre ini biasanya ditunjukan untuk anak-anak TK sampai SD maka permainan warna sangat diperlukan disini [8].

2.5 Algoritma Greedy

Algoritma *Greedy* adalah algoritma yang untuk membentuk solusi langkah perlangkah. Pada setiap langkah tersebut akan dipilih keputusan yang paling optimal. Keputusan tersebut tidak perlu memperhatikan keputusan selanjutnya yang akan diambil, dan keputusan tersebut tidak dapat diubah lagi pada langkah selanjutnya.

Definisi dari Algoritma Greedy secara umum adalah suatu algoritma pencarian solusi yang memecahkan suatumasalah langkah per langkah, dimana pada setiap langkah [9]:

1. Mengambil pilihan terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan.
2. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

2.5.1 Identifikasi dan Elemen-Elemen Algoritma Greedy

Dari skema umum Algoritma Greedy, terdapat 5 elemen identifikasi masalah dalam konteks Algoritma Greedy. Yang diantaranya adalah [9]:

1. Himpunan Kandidat

   Himpunan yang berisi kemungkinan - kemungkinan yang bisa menjadi solusi dari permasalahan.
2. Himpunan Solusi

   Himpunan yang berisi kandidat yang telah terpilih sebagai solusi.
3. Fungsi Seleksi

   Fungsi untuk melakukan seleksi terhadap kandidat agar menghasilkan solusi yang diharapkan
4. Fungsi Kelayakan dan

   Fungsi untuk memastikan bahwa solusi yang dipilih memenuhi syarat.

5. Fungsi Objektif

   Memilih solusi paling optimal dari himpuna solusi.

Algoritma greedy akan membentuk solusi langkah per langkah. Pada setiap langkah algoritma mengambil pilihan terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan nya, sebagaimana prinsip algoritma greedy. Dengan demikian, secara umum algoritma greedybekerja dengan cara melakukan pencarian himpunan solusi pada himpunan kandidat dengan fungsi seleksi dan diverifikasi dengan fungsi kelayakan kemudian memilih solusi paling optimal dengan fungsi objektif.

2.6 Metode

OOP (*Object Oriented Programming*) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam kelas-kelas atau objek-objek.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat [10].

Pemrograman orientasi-objek menekankan konsep berikut:

1. Kelas (*Class*)

   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 fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer 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.

2. Objek (*Object*)

   Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

3. Abstraksi (*Abstract*)

   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.

4. Enkapsulasi (*Encapsulation*)

   Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin 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.

5. Polimorfisme (*Polimorfism*)

   Melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepadaobjek ertentu

di mana pesa 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.

6. Inheritas (*Inheritance*)

   Mengatur *polimorfisme* dan *enkapsulasi* dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki *inheritas*).

2.7 UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualiasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap system informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan system [11].

2.7.1 Sejarah UML

Model biasanya digambarkan dalam bahasa visual, kebanyakan informasi dalam model diekspresikan oleh simbol grafik dan sambungan. Seperti kata pepatah : 'picture says more than a thousands words" juga relevant dengan pemodelan. Dengan menggunakan deskripsi visual lebih mudah untuk mengkomunikasikan relasi yang kompleks, dan membuat pemodelan praktis lebih mudah dikerjakan. Bagaimanapun, tidak ada sesuatupun yang sesuai untuk menggambarkan secara visual, dan beberapa informasi dalam model perlu diekspresikan dalam bentuk teks. Syarat model yang baik adalah [18] :

1. Benar untuk menggambarkan sistem yang dibangun.
2. Konsisten (dalam tinjauan yang berbeda seharusnya tidak mengekspresikan sesuatu yang menimbulkan konflik dengan tinjauan yang lain).
3. Mudah dikomunikasikan dengan yang lain.
4. Mudah diubah.
5. Mudah dimengerti (sesederhana mungkin tapi tidak lebih sederhana).

UML adalah bahasa pemodelan visual yang memiliki semua kemungkinan untuk menjadi standart formal dalam pembuatan model yang berkorespondensi dengan persyaratan model yang baik diatas. Salah satu ide awal dibelakang UML adalah sebagai akhir dari perang metode pada komunitas OOP. OO pertama kali ditemukan

oleh bahasa pemrograman Simula, tapi tidak begitu populer, sampai akhir tahun 80-an dengan bahasa pemrograman seperti C++ dan Smalltalk. Dengan suksesnya OOP, diperlukan metode untuk mendukung pengembangan software. Metode OOP menjadi populer dan digunakan pada tahun 90-an. Beberapa metode lain yang populer adalah Booch, OMT, OOSE/Objectory, Fusion dan Coad/Yourdon.

Tiap metode memiliki notasi tersendiri (semisal simbol yang digunakan untuk menggambarkan model berorientasi obyek), proses (aktifitas apa yang dikerjakan pada bagian yang berbeda dalam pengembangan), dan alat bantu/tools (CASE-tools yang mendukung notasi dan proses). Ini membuat pemilihan metode menjadi keputusan yang sangat penting, dan seringkali menimbulkan argumentasi dan debat sampai ditemukan metode mana yang terbaik atau lebih advanced atau yang benar untuk digunakan dalam suatu proyek khusus. Sebagaimana halnya argumentasi : yang mana sistem operasi yang terbaik ? bahasa pemrograman yang terbaik ? seringkali tidak tersedia jawaban yang baik karena semua metode memiliki kekuatan dan kelemahan sendiri-sendiri. Pengembang yang berpengalaman seringkali kali menetapkan suatu metode sebagai dasar, dan memasukkan ide dan solusi dari metode yang lain. Pada prakteknya, perbedaan antara metode tidak benar-benar harus ada, metode2 tersebut akan terus berkembang dan menjadi mirip satu sama lain. Hal ini ditemukan oleh beberapa pemilik metode dan mereka mulai berusaha untuk bekerja sama.

Grady Booch dan James Rumbaugh di perusahaan Rational Software mulai kerjanya dengan UML di tahun 1994, tujuan mereka adalah membuat metode baru, "Unified Method" yang menggabungkan metode Booch dan metode OMT-2. Dan pada tahun 1995, Ivar Jacobson penemu metode OOSE dan metode Objectory bergabung. Rational juga mengajak Objective Systems, perusahan Swedia yang mengembangkan dan mendistribusikan Objectory. Pada titik ini, pengembang UML juga merealisasikan tujuan mereka untuk membuat bahasa pemodelan standart dan yang diberi nama dengan "Unified Modelling Language". Kemudian UML versi 1.0 dilempar ke pasaran pada tahun 1997.

Selain menggabungkan tiga metode utama dari UML yaitu dari metode Booch,

OMT dan OOSE, mereka juga membawa ide dari metode lain. Ide David Harel tentang state chart juga diadopsi dalam UML state diagram, bagian dari notasi Fusion untuk memberi nomer operasi dimasukkan dalam Collaboration Diagram, dan ide Gamma Helm Johnson Vlissides pada pola dan bagaimana cara mendokumentasinya untuk mengilhami untuk membuat Class Diagram.

Selama tahun 1996, sejumlah organisasi yang bergabung dengan Rational untuk membentuk konsorsium. Organisasi tersebut melihat UML sebagai strategi menguntungkan bagi bisnis mereka dan berharap dapat memberikan konstribusi untuk mendefinisikan UML. Perusahaan tersebut antara lain : DEC, HP, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Texas Instruments, Unisys, dan tentu saja Rational. Perusahaan tersebut juga mendukung proposal untuk mengadaptasi UML sebagaiObject Management Group, organisasi yang mempelopori standart pada bidang berorientasi obyek), sebagai standart untuk bahasa pemodelan.

Rational juga membuat perjanjian dengan Microsoft, yang menyatakan bahwa perusahaan akan bekerjasama untuk mengembangkan dan memasarkan pengembangan tools. Ini berarti bahwa mereka berharap untuk menerima lisensi teknologi dari yang lain, dan itu berarti bahwa produk mereka mudah untuk bekerja sama dengan yang lain (Sebagai contoh Rational Rose akan mengintegrasikan high-end CASE tool untuk lingkungan pengembangan semisal Visual C++ atau Visual Basic.

Pada saat UML mulai menjadi standart, maka akan semakin dikenal oleh pengembang sebagai bahasa pemodelan standart. Bagaimanapun, pada saat OMG direncanakan sebagai bahasa pemodelan standart, UML juga dapat diterima sebagai standart formal. Ini menjadikan lebih banyak permintaan akan UML dan definisi formal yang lebih presisi, sehingga menaikkan kualitas bahasanya. Standardisasi formal juga merupakan hal yang penting bagi banyak industri sebelum mereka menggunakan teknologi baru.

Pada saat sekarang ini, standart OMG UML dipandang sebagai pilihan yang menyakinkan dan pasti untuk pemodelan visual, dan menggantikan bahasa pemodelan

visual lainnya seperti : STEP Express/G, ER, EER, dll. Semua pemasok CASE tool bergabung untuk kontes pendukung UML. Dan setelah sebelumnya terjadi perang metode, sekarang beramai-ramai mengadakan perjanjian untuk suatu notasi yang s*tandart* [18].

2.7.2 Kegunaan UML

UML merupakan bahasa pemodelan general-purpose yang evolusioner, mudah diaplikasikan, dan merupakan standar industri. Dapat diaplikasikan pada bermacam tipe sistem, domain, dan metode atau proses [18].

1. Sebagai bahasa pemodelan yang general-purpose, difokuskan pada pokok himpunan konsep yang dapat dipakai bersama, dan menggunakan pengetahuan bersama dengan mekanisme perluasan.
2. Sebagai bahasa pemodelan yang mudah diaplikasikan, dapat diaplikasikan untuk bermacam tipe sistem (software dan non- software), domain (bisnis vs. software), dan metode atau proses.
3. Sebagai bahasa pemodelan standart industri, bukan merupakan bahasa yang tertutup atau satu-satunya, tapi bersifat terbuka dan sepenuhnya dapat diperluas.

UML berfungsi memungkinkan untuk menangkap, mengkomunikasikan, mengembangkan strategi, taktik dan mengoperasikan pengetahuan untuk meningkatkan nilai tambah melalui peningkatan kualitas, mereduksi biaya, mereduksi time-to-market.

2.7.3 Mekanisme Pendekatan Pemodelan

Mekanisme pendekatan pemodelan dibagi menjadi beberapa bagian yaitu[18]:

1. Perspektif (konseptualisasi, spesifikasi dan realisasi)
    a. Memperjelas asosiasi dengan diagram.
    b. Digunakan untuk mengefektifkan komunikasi.
2. Dikotomi (type-instance, spesifikasi-realisasi, dan statik-dinamik)
    a. Memungkinkan sesuatu ditampilkan dari multiple perspektif.

b. Digunakan untuk menemukan inkonsistensi dalam model.

3. Layer atau *level* abstraksi (*level sistem*, *level sub sistem*, *level class* dan *level* metode).
    a. Memungkinkan perhatian dan konsentrasi difokuskan sehubungan dengan subyek (persoalan atau solusi)
    b. Digunakan untuk mengorganisasi semua diagram, berkenaan dengan model tunggal ke dalam dasar pengetahuan yang mudah dimengerti.
4. Mekanisme perluasan (stereotype, properti, tagged value, dan constraint)
    a. Memungkinkan UML untuk dapat diatur dan diperluas.
    b. Digunakan untuk memastikan bahwa UML akan dapat berkembang bukannya didefinisi ulang untuk memenuhi perubahan kebutuhan.

2.7.4 Tipe Diagram UML

UML mempunyai dua tipe diagram, struktural dan behavioural. Struktural Diagram menggambarkan bagian statik dari sistem. Behavioural Diagram menggambarkan bagian dinamik dari sistem. Behavioural Diagram diklasifikasikan lebih lanjut ke dalam Interaction dan State Diagram [18].

1. Struktural
    1. *Class Diagram* - *Classes*, *interfaces*, *collaboration*, *relationship.*
    2. *Object Diagram* - *object* dan relasi sebagai instance dari prototipe.
    3. *Component Diagram* -komponen dan relasi yang mengilustrasikan implementasi sistem.
    4. *Deployment Diagram* - konfigurasi *runtime* dari node dan objek yang ada node.
2. Behavioural

1. *Use case Diagram* - *use cases*, *actors*, dan relasi. Digunakan untuk mengorganisasikan use case dan *behaviours*.
2. *Interaction*
3. *Sequence Diagram* - waktu urutan *message* dan object *lifeline.*
4. *Collaboration Diagram* - urutan *message* dan organisasi objek dalam interaksi
5. *State Diagram* untuk memodelkan perlaku *objects* di dalam sistem.
6. *Activity Diagram* - arus kerja dari aktifitas, difokuskan pada operasi yang dilewatkan antar objek.
7. *Statechart Diagram* - *life cycle* dari objek sebagai perubahan dari satu *state* ke *state* lain, di *trigger* oleh *message*.

2.7.5 Standar Notasi Dalam UML

Standar notasi dalam UML adalah sebagai berikut [18] :

1. *Actor*

   Actor bukan merupakan bagian dari system. Actor merepresentasikan seseorang atau sesuatu yang berinteraksi dengan system. Actor bisa jadi :

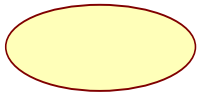
   3. Seorang/sesuatu yang memasukkan Input informasi ke system.
   4. Seseorang/sesuatu menerima informasi dari system.
   5. Atau kedua-duanya.



Gambar 2. 1 Actor

2. *Use Case*

Model Use Case adalah dialog antara actor dan system. Use Case merepresentasikan fungsionalitas yang disediakan oleh system, berupa kemampuan apa yang akan disediakan oleh system untuk actor.



Gambar 2. 2 Use Case

3. *Use Case Diagram*

   Use Case diagram memberikan tinjauan grafis dari semua atau beberapa actor, use case, dan interaksi diantara keduanya untuk suatu system.

4. *Object*

   Objek adalah representasi entitas, baik dunia dunia nyata maupun konseptual. Suatu obyek dapat merepresentasikan sesuatu yang konkrit, semisal truk atau komputer, atau konsep semisal proses kimia, transaksi bank, order pembelian. Objek adalah konsep, abstraksi atau sesuatu yang mempunyai batasan yang didefinisikan dengan baik dan dimaksudkan untuk suatu aplikasi. Tiap objek dalam sistem mempunyai tiga karakteristik berikut : status, behavior dan identitas. Status dari obyek adalah salah satu kondisi mungkin yang harus ada. Status dari suatu objek biasanya dapat berubah sewaktu-waktu, dan didefinisikan oleh himpunan properties yang disebut atribut, dengan nilai properti, plus relasi antara objek dengan obyek yang lain. Sebagai contoh, obyek course offering dalam sistem registrasi kemungkinan mempunyai salah satu status : open atau closed. Jika jumlah mahasiswa yang mendaftar untuk suatu course kurang dari 10, statusnya adalah open, jika lebih dari 10 status akan di-*closed*.

   Behavior menentukan bagaimana suatu objek merespon request dari obyek yang lain, dan segala sesuatu yang harus dikerjakan oleh objek. *Behavior* diimplementasikan oleh himpunan operasi untuk obyek. Dalam sistem

registrasi, *objek course offering* mempunyai *behavior* : *add a student* dan *delete a student*. Identitas berarti tiap objek bersifat unik. Sebagai contoh, Algebra 101 Section 1 dan Algebra 101 Section 2 adalah dua objek dalam sistem registrasi. Meskipun kedua objek merupakan *course offering*, tapi mempunyai identitas yang unik. Dalam UML, obyek direpresentasikan dengan gambar segiempat, dan nama obyek diberi garis bawah.



Gambar 2. 3 Notasi untuk Objek

5. *Class*

   Class adalah deskripsi sekumpulan objek dengan properti (atribut), behavior (operasi), relasi ke objek lain, dan semantik. Sehingga *class* merupakan suatu template untuk membuat objek.

   Class CourseOffering didefinisikan dengan karakteristik berikut :

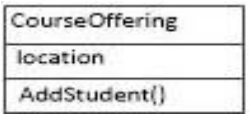
   - Atribut - lokasi, waktu yang disediakan.
   - Operasi - retrieve lokasi, retrieve waktu, tambahkan mahasiswa yang mengambil.

   Algebra 101, Section 1 dan Algebra 101, Section 2 adalah obyek yang menjadi milik *class CourseOffering*. Tiap obyek memiliki nilai untuk suatu atribut dan mengakses operasi yang dispesifikasi oleh class *CourseOffering*.

   Class yang baik menangkap satu abstraksi - seharusnya mempunyai satu tema. Sebagai contoh, suatu class mempunyai kemampuan untuk memaintain informasi tentang student dan informasi mengenai semua course yang tersedia yang harus harus diambil oleh s*tudent*, ini bukan merupakan class yang baik karena mempunyai lebih dari satu tema utama. *Class* tersebut seharusnya dibagi menjadi dua class yaitu class *StudentInformation* dan class *StudentHistory*.

Class harus diberi nama dengan menggunakan kosa kata domain. Nama seharusnya merupakan kata benda tunggal dan merupakan karakteristik terbaik untuk suatu abstraksi. Jika class diberi nama dengan singkatan, maka namanya yang lengkap harus didokumentasi dalam dokumentasi class.

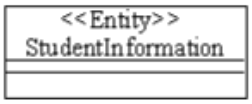
Batasan tool juga mempengaruhi himpunan standart nama. Pada Rational Rose 4.0, actor adalah suatu class yang mempunyai stereotype <<*Actor*>>. Sehingga actor dan class tidak boleh memiliki nama yang sama. Pilihan lain dengan menambahkan kata Actor untuk setiap actor dan class bisa berisi nama yang lebih singkat (misal *StudentActor* dan *Student*).



Gambar 2. 4 Notasi UML untuk Class

6. *Class with Stereotype*

Class juga mempunyai stereotype, yang menyediakan kemampuan untuk membuat elemen pemodelan yang baru. Disini, akan dibuat macamclass yang baru. Beberapa stereotype yang umum untuk suatu class adalah : entity, boundary, control, utility dan exception. Stereotype untuk suatu class ditunjukkan diatas namaclass dengan warna merah.



Gambar 2. 5 Class dengan suatu Stereotype

Entity Classes:

EntityClasses memodelkan informasi danasosiasi behavioryang secara umum aktif dalam waktu lama. Tipe class ini merefleksikan entiti dunia-nyata, yang diperlukan untuk membentuk task internaldarisuatu sistem. Biasanya independent dari sekitarnya, dan tidak tahubagaimana cara sekitarnya

berkomunikasi dengan sistem. Aplikasinya independent, artinya bisa digunakan untuk lebih dari satu aplikasi.

Tahap awal adalah menguji responsibility/tanggungjawab yang didokumentasi dalam arus event untukidentifikasi use case (apa yang harus dikerjakan oleh sistem). Entity class adalah class yang diperlukan oleh sistem untuk menyelesaikan beberapa responsibility/tanggung jawab. Noun dan noun phrases digunakan untuk menggambarkan suatu responsibility. Daftar noun pada awalnya harus disaring jika noun berisi sesuatu diluar domain persoalan, noun yang merupakan ekspresi bahasa, noun yang redundant dan noun yang merupakan deskripsi dari struktur class.

Entity Class biasanya ditemukan padaFase Uraian/Elaborasi, dan disebut dengan domain classes karena berhubungan dengan abstraksi entiti dunia nyata.

Boundary Classes:

Boundary Classes menangani komunikasi antara lingkungan sistemdan yang ada di dalamsistem, menyediakan interface bagi user atau sistemlain (misal interface ke actor). Merupakan bagian yang tergantung pada lingkungan sistem.

Tiap pasang actor/scenario secarafisik diuji untuk menemukan boundary classes. *Boundary Classes* dipilih dalamfase Uraian pada *level* atas. Misal, anda memodelkan suatu jendela tapitidak memodelkan masing-masing kotak dialog dan tombol. Pada titik ini, akan didokumentasi user *interface* yang dibutuhkan, bukan cara mengimplementasikan *interface*.

Selamaproses desain, class-class ini diperbaiki untuk memilih mekanisme user interface - dengan pertimbangan bagaimana cara mereka diimplementasikan.

Boundary classes juga ditambahkan untuk fasilitas komunikasi dengan sistemlain. Selamaproses desain, *class-class* ini diperbaiki dengan pertimbangan protokol komunikasi yang dipilih.

Control Classes

Control Classes memodelkan urutan behaviorspesifik terhadapsatu atau lebih usecase. *Control classes* mengkoordinasi *event* yang diperlukan untuk merealisasikan behaviorspesifik dalamuse *case*. *Control class* biasanya merupakan class yang tergantung pada aplikasi.

Pada awal fase Uraian, control class ditambahkan ke tiap pasang actor/use Case. Control class bertanggungjawab terhadap arus event dalamuse case. Penggunaan dari control class sangat subyektif. Jika penggunannya mengakibatkan behavior menjadi terpisahdari data, maka ini terjadi karena control class mengerjakan terlalu banyak, lebih dari sekedar mengurutkan, dan ini bukanlah hal yang baik. Sebagai contoh, dalamsistemregistrasi course, student memilih course yang tersedia, kemudian jikacourse tersebut tersedia, student menambahkannya.

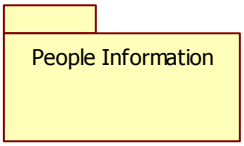
7. *Package*

   Package adalah tinjauan logika dari suatu model yang berupa kumpulan *package* dan/atau *classes*. Dengan mengelompokkan *class-class* ke dalam *package*, makadapat dilihat *level* atas dari tinjauan suatu model, atau model dapat dilihat lebih jauh dengan mengetahui isi *package*.

   Tiap package berisi interface yang direalisasikan oleh *public class* yaitu *class* yang berkomunikasi dengan *class* pada *package* yang lain. Sisanya adalah class implementasi - yaitu *class* yang tidak berkomunikasi dengan *class* pada *package* lain.

   Jika sistemnya kompleks, *package* dapat dibuat pada awal fase Uraian untuk fasilitas komunikasi. Untuk sistem yang lebih sederhana class ditemukan pada awal fase analisis dapat dikelompokkan kedalam satu *package*, sistem itu sendiri. Kemudian setelah analisis dan desain mengalami kemajuan, konsep dari *package* akan digunakan untuk mengelompokkan sekumpulan *class* yang diperlukan untuk pengambilan keputusan arsitektural dari suatu sistem.

Dalam UML, package direpresentasikan dalamgambar seperti folder.

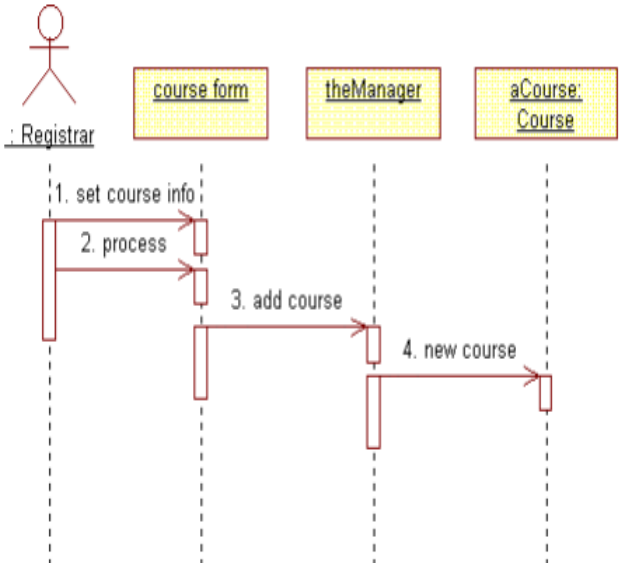


Gambar 2. 6 Notasi UML untuk Package

8. *Sequence Diagram*

Sequence Diagrammenunjukkan interaksi obyek yang diatur dalam satuan waktu. *Sequence Diagram* menangkap objek dan class yang terlibat dalam scenario dan urut-urutan message yang ditukar diantara objek diperlukan untuk melaksanakan fungsionalitas scenario. *Sequence Diagram* berasosiasi dengan *use case* selama proses pengembangan.

Dalam UML, obyek dalam *Sequence Diagram* digambar dengan segiempat yang berisi namaobyek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara : nama objek, nama objek dan kelas atau hanya nama class.



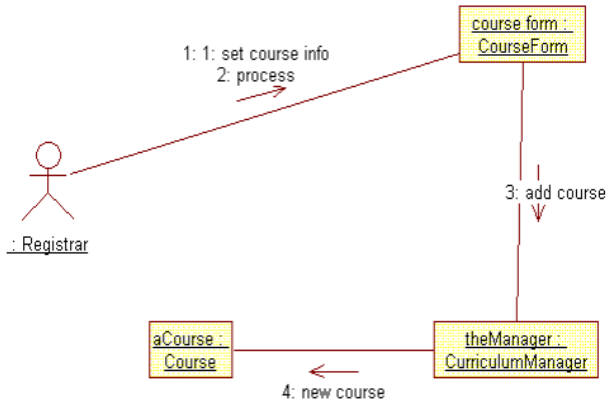
Gambar 2. 7 Sequnce Diagram

9. *Collaboration Diagram*

   Collaboration Diagram adalah alternatif lain untuk menunjukkan suatu scenario. Tipe dari diagram yang menunjukkan interaksi objek yang diorganisasi sekitar objek dan relasi satu sama lain.

   Collaboration Diagram, berisi :

   a. Objek yang digambar dalam persegi panjang.
   b. Link diantara objek yang ditunjukkan sebagai garis yang menghubungkan obyek yang di-link.
   c. Message yang menunjukkan teks dan garis yang mengarah dari client ke supplier.



Gambar 2. 8 *Collaboration Diagram*

2.8 Aplikasi Pembangunan Perangkat Lunak

2.8.1 Adobe Flash

Adobe Flash adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe Systems. Adobe Flash digunakan untuk membuat gambar vektor maupun animasi gambar tersebut. Berkas yang dihasilkan dari perangkat lunak ini mempunyai file extension .swf dan dapat diputar di penjelajah web yang telah dipasangi Adobe Flash Player. Sebelum tahun 2005, Flash dirilis oleh Macromedia. Flash 1.0 diluncurkan pada tahun 1996 setelah Macromedia membeli program animasi vektor bernama Future Splash. Versi terakhir yang diluncurkan di pasaran dengan

menggunakan nama ‘Macromedia’ adalah adalah Macromedia Flash 8. Pada tanggal 3 Desember 2005 Adobe Systems mengakuisisi Macromedia dan seluruh produknya, sehingga nama Macromedia Flash berubah menjadi Adobe Flash. Adobe Flash merupakan sebuah program aplikasi pembuat animasi yang popular untuk saat ini, dimana semua pekerjaan yang berhubungan dengan animasi 2D bahkan 3D pada Flash versi terbaru saat ini, dapat dilakukan dengan mudah melalui program aplikasi ini.
Berikut merupakan kegunaan program aplikasi Adobe Flash :

1. Flash Untuk Membuat Animasi
2. Flash untuk e-card
3. Flash untuk presentasi
4. Flash untuk web
5. Flash untuk CD Pembelajaran Interaktif

Kelebihan Adobe Flash :

1. Merupakan teknologi animasi web yang paling populer saat ini sehingga banyak didukung oleh berbagai pihak.
2. Ukuran file yang kecil dengan kualitas yang baik.
3. Dapat membuat website, cd-interaktif, animasi web, animasi kartun, kartu elektronik, iklan TV, banner di web, presentasi interaksi, permainan, aplikasi web [12].

2.9 Pengujian Perangkat Lunak

Pengujian perangkat lunak Adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean.

Pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi di mana peluang terjadinya kesalahan manusia sangat besar dan arena ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas.

Meningkatnya visibilitas (kemampuan) perangkat lunak sebagai suatu elemen sistem dan “biaya” yang muncul akibat kegagalan perangkat lunak, memotivasi dilakukannya perencanaan yang baik melalui pengujian yang teliti. Pada dasarnya, pengujian merupakan satu langkah dalam proses rekayasa perangkat lunak yang dapat dianggap sebagai hal yang merusak daripada membangun.

Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah [19]:

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan
2. Test case yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya

Sasaran itu berlawanan dengan pandangan yang biasanya dipegang yang menyatakan bahwa pengujian yang berhasil adalah pengujian yang tidak ada kesalahan yang ditemukan. Data yang dikumpulkan pada saat pengujian dilakukan memberikan indikasi yang baik mengenai reliabilitas perangkat lunak dan beberapa menunjukkan kualitas perangkat lunak secara keseluruhan, tetapi ada satu hal yang tidak dapat dilakukan oleh pengujian, yaitu *pengujian tidak dapat memperlihatkan* tidak adanya cacat, pengujian hanya dapat memperlihatkan bahwa ada kesalahan perangkat lunak.

Sebelum mengaplikasikan metode untuk mendesain test case yang efektif, perekayasa perangkat lunak harus memahami prinsip dasar yang menuntun pengujian perangkat lunak, yaitu:

1. semua pengujian harus dapat ditelusuri sampai ke persyaratan pelanggan, maksudnya mengungkap kesalahan dari cacat yang menyebabkan program gagal.

2. Pengujian harus direncanakan lama sebelum pengujian itu mulai, maksudnya semua pengujian dapat direncanakan dan dirancang sebelum semua kode dijalankan.
3. Prinsip Pareto berlaku untuk pengujian perangkat lunak, maksudnya dari 80% kesalahan yang ditemukan selama pengujian dapat ditelusuri sampai 20% dari semua modul program.
4. Pengujian harus mulai "dari yang kecil*" dan* berkembang ke pengujian "yang besar", Selagi pengujian berlangsung maju, pengujian mengubah focus dalam usaha menemukan kesalahan pada cluster modul yang terintegrasi dan akhirnya pada sistem.
5. Pengujian yang mendalam tidak mungkin karena tidak mungkin mengeksekusi setiap kombinasi jalur skema pengujian dikarenakan jumlah jalur permutasi untuk program menengah pun sangat besar.
6. Untuk menjadi paling efektif, pengujian harus dilakukan oleh pihak ketiga yang independent

Dalam lingkungan yang ideal, perekayasa perangkat lunak mendesain suatu program computer, sebuah sistem atau produk dengan testabilitas dalam pikirannya. Hal ini memungkinkan individu yang berurusan dengan pengujian mendesain test case yang efektif secara lebih mudah. Testabilitas adalah seberapa mudah sebuah program computer dapat diuji. Karena sangat sulit, perlu diketahui apa yang dapat dilakukan untuk membuatnya menjadi lebih mudah. Procedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi.

Sasaran utama desain test case adalah untuk mendapatkan serangkaian pengujian yang memiliki kemungkinan tertinggi di dalam pengungkapan kesalahan pada perangkat lunak. Untuk mencapai sasaran tersebut, digunakan 4 kategori yang berbeda dari tehnik desain test case: *Pengujian white-box*, *pengujian black-box*, *Integrasi Bottom-Up* dan *Integrasi Top-Down.*

2.9.1 Pengujian White-Box

Pengujian *white-box* berfokus pada struktur control program. Test case dilakukan untuk memastikan bahwa semua statemen pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian basic path, tehnik pengujian white-box, menggunakan grafik (matriks grafiks) untuk melakukan serangkaian pengujian yang independent secara linear yang akan memastikan cakupan.

Pengujian aliran data dan kondisi lebih lanjut menggunakan logika program dan pengujian loop menyempurnakan tehnik white-box yang lain dengan memberikan sebuah prosedur untuk menguji loop dari tingkat kompleksitas yang bervariasi. Pengujian black-box didesain untuk mengungkap kesalahan pada persyaratan fungsional tanpa mengabaikan kerja internal dari suatu program.

2.9.2 Pengujian Blak-Box

Pengujian black-box berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut [19]:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan perilaku (behavior) atau kesalahan kinerja.
5. Inisialisasi dan pemutusan kesalahan.

Tes ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan berikut ini:

1. Bagaimana validitas fungsional diuji?
2. Bagaimana perilaku dan kinerja sistem diuji?
3. Apa kelas input akan membuat kasus uji yang baik?
4. Apakah sistem sensitive terhadap nilai input tertentu?
5. Bagaimana batas-batas kelas data yang terisolasi?

6. Kecepatan dan volume data seperti apa yang dapat ditolerir sistem?
7. Efek apakah yang akan menspesifikasikan kombinasi data dalam sistem operasi?

2.9.2.1 Teknik Pengujian Black-Box

Teknik pengujian black-box berfokus pada domain informasi dari perangkat lunak, dengan melakukan test case dengan menpartisi domain input dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode pengujian graph-based mengeksplorasi hubungan antara dan tingkah laku objek-objek program. Partisi ekivalensi membagi domain input ke dalam kelas data yang mungkin untuk melakukan fungsi perangkat lunak tertentu. Analisis nilai batas memeriksaa kemampuan program untuk menangani data pada batas yang dapat diterima.

Metode pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. GUI, arsitektur client/ server, dokumentasi dan fasilitas help dan sistem real time masing-masing membutuhkan pedoman dan tehnik khusus untuk pengujian perangkat lunak [19].