BAB II
TINJAUAN PUSTAKA

2.1 Arsitektur

Arsitektur sebagai salah satu ciri kebudayaan manusia, merupakan seni dan teknik bangunan sama halnya dengan seni sastra, seni suara, seni drama, seni tari dan bahasa dapat mencerminkan keadaan suatu kebudayaan masyarakat di lokasi tertentu. Karya-karya arsitektur (yang sebagian besar merupakan bangunan – bangunan monumental) sampai saat ini masih merupakan tanda kebesaran dari cipta dan karsa (kebudayaan) manusia yang pernah ada dalam sejarah perkembangan manusia.[11]

2.1.1 Pengenalan Dunia Arsitektur

Telah banyak pandangan atau pengertian serta batasan tentang ‘arsitektur’ dari beberapa ahli / pakar. Mulai dari pandangan *Marcus Vitrovius Fillio* tentang arsitektur yaitu: arsitektur adalah proses penciptaan dan pewujudan bangunan yang didalamnya memiliki tiga pilar utama, yaitu: kegunaan atau fungsionalitas, kekokohan / kekuatan dan estetika atau keindahan. Pandangan dari *Van Rommondt* tentang arsitektur, yaitu: proses penciptaan bangunan bagi kehidupan manusia, yang didalamnya tercapai rasa, aman untuk dihuni, nyaman untuk digunakan dan bahagia/puas dari aspek visual.

Pandangan yang terkini tentang arsitektur, yaitu dari *Ammos Rapoport* yang menyatakan bahwa: arsitektur adalah proses penciptaan dan pewujudan dari ruang dalam bentuk ‘lingkungan binaan’ (build environment) bagi kehidupan manusia, dimana lingkungan yang diciptakan tersebut hendaklah adaptif dan selaras dengan lingkungan sernya. Dan yang terakhir dari *Christian Normberg Schulz*, yaitu: arsitektur dalam proses penciptaan ruang atau bangunan yang didalamnya

mengandung tiga unsur penting, yaitu: kegunaan / fungsi, teknik / keteknikan dan ekspresi bentuk.

James Snyder mengungkapkan bahwa terdapat tingkatan ‘skala’ atau ‘besaran’ ruang yang mesti dipahami dalam kerangka kerja perancangan arsitektur. Tingkatan skala atau besaran ruang dalam arsitektur tersebut adalah: Skala / Besaran Ruang Tingkat Kecil (*Mikro*), yang meliputi: *the part of the room scale, the room scale, the multi rooms scale*. Skala / Besaran Ruang Tingkat Menengah (*Midle*), yang meliputi: *the part of the building scale, the building scale* dan *the building with site scale*. Skala / Besaran Ruang Tingkat Besar (*Makro*) yang meliputi: *the multi building scale, the district scale* dan *the part of urban scale*.

Dengan memahami tingkatan skala atau besaran ruang dalam arsitektur, maka akan mengenal dan paham sampai seberapa detail pekerjaan perancangan arsitektur dilakukan. Atau pada skala makro dapat dilihat seberapa luas kaitan antara perancangan arsitektur dengan profesi lain yang terkait didalamnya. Sekali lagi wujud dari ruang dalam arsitektur dapat beraneka ragam, seperti: ruangan dalam (*interior*), bagian dari bangunan (*part of the building*), satu bangunan utuh (single building) hingga ke lingkungan / kawassan (*district*) bahkan sampai skala lingkungan kota.

2.2 Jembatan

Jembatan secara umum adalah suatu konstruksi yang berfungsi untuk menghubungkan dua bagian jalan yang terputus oleh adanya rintangan-rintangan seperti lembah yang dalam, alur sungai, danau, saluran irigasi, kali, jalan kereta api, jalan raya yang melintang tidak sebidang dan lain-lain.[10] Berikut ini proses saat pembangunan jembatan.

2.2.1 Proses Persiapan

Proses persiapan yang dimaksud disini adalah suatu proses persiapan yang untuk mengerjakan suatu pembangunan yang mana dalam pembangunan itu ada suatu pekerjaan yang nantinya harus memenuhi standar kualitas mutu, jadi untuk mendapatkan hasil yang maksimal yang harus dilakukan adalah:

1. Survey ke lapangan untuk melihat lokasi yang akan menjadi objek pembangunan.
2. Pengetesan material.
3. Design metoda kerja
4. Traffic management
5. Penyimpanan material
6. Kalibrasi alat ukur
7. Mix design
8. Job mix formula
9. Pengendalian drainase ke aliran sungai

2.2.2 Proses Pemancangan

Proses pemancangan yang dimaksud disini adalah suatu proses pemancangan tiang untuk tiang penyangga lantai jembatan.

Dalam proses pemancangan inilah yang harus betul2 diamati karena ini akan mengkibatkan resiko yang besar kedepannya nanti dan untuk mengurangi resiko yang akan terjadi, proses pemancangan harus memenuhi standar kualitas mutu sebagai berikut:

1. Jumlah pukulan harus sampai pada tanah karang.
2. Jenis alat pancang harus standar SNI.
3. Final set
4. Loading test
5. PDA test
6. Pile spacing
7. Koordinat

Kedua proses inilah yang harus dilakukan dalam perencanaan pembuatan jembatan ,karena apabila proses ini tak di lakukan maka resiko runtuhnya jembatan tidak dapat diperhitungakan.

Jenis jembatan berdasarkan fungsi, lokasi, bahan konstruksi dan tipe struktur sekarang ini telah mengalami perkembangan pesat sesuai dengan kemajuan jaman dan teknologi, mulai dari yang sederhana sampai pada konstruksi yang mutakhir. Berdasarkan fungsinya, jembatan dapat dibedakan sebagai berikut.

1. Jembatan jalan raya (highway bridge).
2. Jembatan jalan kereta api (railway bridge).
3. Jembatan pejalan kaki atau penyeberangan (pedestrian bridge).

Berdasarkan lokasinya, jembatan dapat dibedakan sebagai berikut.

1. Jembatan di atas sungai atau danau.
2. Jembatan di atas lembah.
3. Jembatan di atas jalan yang ada (fly over).
4. Jembatan di atas saluran irigasi/drainase (culvert).
5. Jembatan di dermaga (jetty).

Berdasarkan bahan konstruksinya, jembatan dapat dibedakan menjadi beberapa macam, antara lain :

8. Jembatan kayu (log bridge).
9. Jembatan beton (concrete bridge).
10. Jembatan beton prategang (prestressed concrete bridge).
11. Jembatan baja (steel bridge).
12. Jembatan komposit (compossite bridge).

Berdasarkan tipe strukturnya, jembatan dapat dibedakan menjadi beberapa macam, antara lain :

1. Jembatan plat (slab bridge).
2. Jembatan plat berongga (voided slab bridge).
3. Jembatan gelagar (girder bridge).
4. Jembatan rangka (truss bridge).
5. Jembatan pelengkung (arch bridge).
6. Jembatan gantung (suspension bridge).
7. Jembatan kabel (cable stayed bridge).

8. Jembatan cantilever (cantilever bridge).

2.2.3 Struktur Jembatan

Secara umum struktur jembatan dapat dibedakan menjadi dua bagian yaitu struktur atas dan struktur bawah.[10]

1. Struktur Atas (Superstructures)

   Struktur atas jembatan merupakan bagian yang menerima beban langsung yang meliputi berat sendiri, beban mati, beban mati tambahan, beban lalu-lintas kendaraan, gaya rem, beban pejalan kaki, dll.

   Struktur atas jembatan umumnya meliputi :

   A. Trotoar:
      1) Sandaran dan tiang sandaran.
      2) Peninggian trotoar (Kerb).
      3) Slab lantai trotoar.
   B. Slab lantai kendaraan.
   C. Gelagar (Girder).
   D. Balok diafragma.
   E. Ikatan pengaku (ikatan angin, ikatan melintang).
   F. Tumpuan (Bearing).

2. Struktur Bawah (Substructures)

   Struktur bawah jembatan berfungsi memikul seluruh beban struktur atas dan beban lain yang ditumbulkan oleh tekanan tanah, aliran air dan hanyutan, tumbukan, gesekan pada tumpuan dsb. untuk kemudian disalurkan ke fondasi. Selanjutnya beban-beban tersebut disalurkan oleh fondasi ke tanah dasar.

   Struktur bawah jembatan umumnya meliputi :

   A. Pangkal jembatan (Abutment).
      1) Dinding belakang (Back wall).
      2) Dinding penahan (Breast wall).
      3) Dinding sayap (Wing wall).

4) Oprit, plat injak (Approach slab).
5) Konsol pendek untuk jacking (Corbel).
6) Tumpuan (Bearing).

B. Pilar jembatan (Pier).
1) Kepala pilar (Pier Head).
2) Pilar (Pier), yg berupa dinding, kolom, atau portal.
3) Konsol pendek untuk jacking (Corbel).
4) Tumpuan (Bearing).

3. Fondasi

Fondasi jembatan berfungsi meneruskan seluruh beban jembatan ke tanah dasar. Berdasarkan sistimnya, fondasi abutment atau pier jembatan dapat dibedakan menjadi beberapa macam jenis, antara lain :

A. Fondasi telapak (spread footing).
B. Fondasi sumuran (caisson).
C. Fondasi tiang (pile foundation)
1) Tiang pancang kayu (Log Pile).
2) Tiang pancang baja (Steel Pile).
3) Tiang pancang beton (Reinforced Concrete Pile).
4) Tiang pancang beton prategang pracetak (Precast Prestressed Concrete Pile), spun pile.
5) Tiang beton cetak di tempat (Concrete Cast in Place), borepile, franky pile.
6) Tiang pancang komposit (Compossite Pile).

2.3 Game

Video game merupakan salah satu media hiburan yang paling popular untuk semua kalangan usia. Sejak pertama kali ditemukan sampai saat ini, teknologi game telah mengalami kemajuan yang pesat. Hal ini ditandai dengan berkembangnya jenis,

produk, alat, dan jenis interaksi gamedengan penggunaan yang semakin beragam bentuknya.

2.3.1 Pengertian Game

Game berasal dari kata bahasa inggris yang memiliki arti dasar Permainan. Permainan dalam hal ini merujuk pada pengertian "kelincahan intelektual" (*intellectual playability*). *Game* juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana *game* itu menarik untuk dimainkan secara maksimal. *Game* merupakan sebuah bentuk seni dimana penggunanya, disebut dengan pemain (*player*), diharuskan membuat keputusan-keputusan dengan tujuan untuk mengelola sumber daya yang diperoleh dari kesempatan-kesempatan bermain (token) miliknya, untuk mencapai suatu tujuan tertentu.

Video *game* adalah bentuk *game* yang interaksi utamanya melibatkan media video (dan biasanya melibatkan audio). Berdasarkan representasi visualnya, *game* dapat dibedakan menjadi 2 jenis, yaitu game 2 dimensi (2D) dan 3 dimensi (3D). *Game* 2D adalah game, yang secara matematis, hanya melibatkan 2 elemen koordinat kartesius, yaitu xdan y, sehingga konsep kamera pada game 2D hanya menentukan "gambar" mana pada game yang dapat dilihat oleh pemain. Sedangkan game 3D adalah game yang selain melibatkan elemen xdan y, juga melibatkan elemen zpada perhitungannya, sehingga konsep kamera pada game 3D benar-benar menyerupai konsep kamera pada kehidupan nyata, yaitu selain digeser (seperti pada game 2D), juga dapat diputar dengan sumbu tertentu.

Menariknya akan tantangan maupun visualisasi yang ada pada game sering mengakibatkan ketergantungan terhadap game tersebut (*addicted)*. Ini merupakan salah satu akibat buruk yang disebabkan oleh game (*game addicted).*[6]

2.3.2 Sejarah Game Komputer

Game komputer adalah sebuah permainan interaktif yang dijalankan menggunakan komputer. Program komputer menerima input dari si pemain melalui pengendali dan menampilkan lingkungan buatan melalui TV atau layar monitor. Sejarah teknologi *game* komputer secara langsung berhubungan dengan perkembangan komputer itu sendiri.

Game komputer generasi pertama, tahun 1952, adalah sebuah permainan OXO (tic-tac-toe) dalam versi grafik yang diciptakan oleh seorang mahasiswa Universitas Cambridge bernama A.S. Gouglas. *Game* ini diprogram dengan memakai komputer EDVAC vacuum tube yang memiliki layer berupa *cathode ray tube* (CRT). Pada tahun 1958. William Haginbotham menciptakan *video game* pertamanya, dan *game* komputer pertama yang benar-benar diciptakan menggunakan komputer itu sendiri adalah Spacewar oleh Steve Russel yang terinspirasi dari kisah fiksi ilmiah berjudul Skylark karangan Edward E Smith. Steve Russel bekerja menggunakan sebuah komputer mainframe bernama MIT PDP-1 yang biasa dipakai untuk perhitungan statistic pada tahun 1961. Saat itu di tahun 60-an, komputer adalah barang sangat mewah. Komputer umumnya dipakai untuk kepentingan riset dalam dunia militer.

Game generasi kedua, di tahun 1971, Nolan Bushnell bersama Ted Dabney menciptakan *game* bergenre *arcade* yang pertama yang diberi judul Komputer Space, yang didasari oleh Spacewar. Tahun 1972, Nolan dan Ted memulai Atari komputer. Mereka mengembangkan *game* berjudul Pong yaitu *game* pertama yang tersedia untuk publik. Karena sebelumnya *game* hanya ada didalam komputer mainframe yang digunakan untuk kesenangan sendiri. Asal usul Pong dimulai saat Nolan ingin membuat *game* sederhana dan mudah dimengerti. Menggunakan memory dan micro processor kelas rendah, dengan kemampuan grafis yang sederhana, akhirnya dia membuat versi elektronik dari permainan ping pong yang kemudian menjadi Pong. Pong kemudian berevolusi menjadi sebuah *game* bernama Breakout. *Game* itu diciptakan oleh Steve Jobs untuk Atari dan bersama Steve Wozniak mulai berpikir

untuk menciptakan sebuah PC. Mereka meminjam semua peralatan yang dipakai dalam proyek Breakout untuk membuat sebuah prototype bernama Apple I. dan itu adalah cikal bakal komputer Apple Macintosh yang ada sekarang.

Pada tahun 1980, Atari mengeluarkan *game* berjudul Asteroid dan Lunar Lander. Kedua *game* tersebut adalah *game* pertama yang didaftarkan kepada perusahaan untuk mendapatkan hak cipta yang paten. Asteroid merupakan *game* yang penuh inovasi baru dalam grafisnya, dengan menggunakan metode raster, *game* ini merupakan *grafis vector line* seperti yang ada pada *osciloscope*. Pengenalan Atari Video komputer *System* (Atari 2600) dengan CPU biasa dan slot untuk kasetnya, yang menjadi suatu era baru dalam dunia *game*. Di tahun 1980 itu juga menunjukkan adanya peningkatan penjualan PC yang biasa dipakai untuk *game*.

Kini komputer bersaing dengan mesin *console* seperti PS atau Xbox. Komputer tidak hanya sebagai mesin untuk kepentingan bisnis semata, tapi juga untuk hiburan seperti *game*.

2.3.3 Elemen Dasar Game

Dalam suatu game terdapat beberapa elemen yang menyusun game tersebut, seperti jenis game, karakter game, background, elemen sound/suara, dan gerakan-gerakan dasar.

2.3.4 Jenis "*Platform"/* Alat yang digunakan

Beberapa alat yang digunakan untuk menjalankan suatu game, diantaranya adalah :

1. Arcade Games

   Game yang dijalankan pada mesin dengan input dan output audio visual yang telah terintegrasi dan tersedia ditempat-tempat umum, seperti mal, bandara dan sebagainya. Contoh permainan ding-dong.

2. PC Games

   Game yang dimainkan pada PC (Personal Computer) yang memiliki kelebihan yaitu memiliki tampilan antarmuka yang baik untuk I/O, output visual kualitas tinggi karena layar computer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televisi biasa.

3. Console Games

   Game yang dijalankan pada suatu mesin spesifik yang biasanya tersedia di rumah pribadi, seperti Playstation, Microsoft Xbox 360, Nintendo Wii, Sega dan sebagainya.

4. Mobile Games

   Game yang dapat dimainkan khusus untuk mobile phone atau PDA.

5. Handheld Games

   Game yang dimainkan di console khusus *video game* yang dapat dibawa kemana-mana, contoh *GameBoy*, Nintendo DS dan Sony PSP.

2.3.5 Jenis-jenis *Game* Komputer

A. RPG (Role Playing Game)

RPG adalah salah satu game yg mengandung unsur experience atau leveling dalam gameplay nya. Biasanya dalam game ini memiliki kebebasan untuk menjelajah dunia game tersebut, dan kadang kala dalam beberapa game, dapat menentukan endingdari game tersebut RPG terbagi 2 : Action RPG & Turn Based RPG (Menunggu giliran untuk berjalan seperti Monopoly).

B. FPS (First Person Shooting)

FPS adalah game yg tembak menembak yg memiliki ciri utamanya adalah penggunaan sudut pandang orang pertama dibuat dibelakang senjata

C. Third Person Shooter

TPS adalah game yg mirip dengan FPS yaitu memiliki gameplay tembak menembak hanya saja sudut pandang yg digunakan dalam game ini adalah orang ketiga.

D. Strategy

Strategy adalah genre game yg memiliki gameplay untuk mengatursuatu unit atau pasukan untuk menyerang markas musuh dalam rangkamemenangkan permainan. biasanya di dalam game Strategy, dituntut untuk mencari gold untuk membiayai pasukan. Games Strategy dibagi 2 :

1. Real Time Strategy (RTS).

   Pada game ini, dapat dikendalikan pasukan secara langsung, dari mencari sumber daya, hingga menghancurkan musuh. Semua pertempuran ini dapat disaksikan secara langsung. Contoh : Warcraft 3

2. Turn Based Strategy (TBS).

   Sistemnya seperti Turn Based RPG, tetapi disini selain mengendalikan character utama, dikendalikannya pasukan dan kota untuk memenangkan pertarungan. Biasanya memainkan game nya di atas peta. Contoh : Civilization

E. Simulation

Adalah genre yang mementingkan realisme. Segala faktor pada game ini sangat diperhatikan agar semirip didunia nyata. Segala nilai, material,referensi, dan faktor lainnya adalah berdasarkan dunia nyata. Cara memainkannya juga berbeda, karena biasanya kontrol yang dimiliki cukup rumit. Genre simulasi meliputi game racing, flight, sampai militer.

F. Tycoon

Tycoon adalah game yang menjadikannya sebagai seorang businessman yang akan mengembangkan sesuatu Property untuk dikembangkan hingga laku di pasaran

G. Racing

Racing Game adalah game berjenis balapan yang memungkinkan untuk mengendalikan sebuah kendaraan untuk memenangkan sebuah balapan.

H. Action Adventure

Action Adventure adalah game berupa petualangan salah seorang karakter yang penuh dengan penuh aksi yg akan terus ada hingga game tersebut tamat. (Biasanya Action dimasukan kategori RPG)

I. Arcade

Arcade game adalah genre game yang tidak terfokus pada cerita, melainkan hanya dimainkan "just for fun" atau untuk kejar-mengejar point / highscore

J. Fighting Game

Fighting adalah genre game bertarung. Seperti dalam arcade, pemain dapat mengeluarkan jurus-jurus ampuh dalam pertarungannya. Genre fighting biasanya one on one dalam sebuah arena yang sempit

K. Sports

Adalah genre bertema permainan olahraga. Sistem permainan akan berbeda-beda tergantung jenis olahraga yang menjadi tema game tersebut [6]

2.4 Game Edukasi

Game edukasi adalah game yang khusus dirancang untuk mengajarkan user terhadap suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya.[7]

2.4.1 Kriteria Game Edukasi

Menurut Hurd dan Jenuings, perancangan yang baik haruslah memenuhi kriteria dari *education* game itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education* game, yaitu:

1. Nilai Keseluruhan (*Overall Value*)

   Nilai keseluruhan dari suatu game terpusat pada desain dan panjang durasi game. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur timer.

2. Dapat Digunakan *(Usability)*

   Mudah digunakan dan diakses adalah poin penting bagi pembuat game. Aplikasi ini merancang sistem dengan interface yang user friendly sehingga user dengan mudah dapat mengakses aplikasi.

3. Keakuratan *(Accuracy)*

   Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah game dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model game pada tahap perencanaan.

4. Kesesuaian *(Appropriateness)*

   Kesesuaian dapat diartikan bagaimana isi dan desain game dapat diadaptasikan terhadap keperluan user dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan user untuk membantu pemahaman user dalam menggunakan aplikasi.

5. Relevan *(Relevance)*

   Relevan artinya dapat mengaplikasikan isi game ke target user. Agar dapat relevan terhadap user, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak-anak maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Objektifitas *(Objectives)*

   Objektifitas menentukan tujuan user dan kriteria dari kesuksesan atau kegagalan. Dalam aplikasi ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik *(Feedback)*

   Untuk membantu pemahaman user bahwa permainan (performance) mereka sesuai dengan objek game atau tidak, feedback harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

2.4.2 Poin-Poin pada Game Edukasi

Adapun hal-hal yang harus ada pada game edukasi adalah sebagai berikut:

1. Produktif

   Game edukasi harus dapat mengembangkan sikap produktif dari para pemain game. Jadi inti dari game tersebut akan tersimpan di memori sebagai pemain sehingga suatu saat bisa menginovasi dan merangsang kreativitas dalam menciptakan sesuatu yang baru.

2. Aktif

   Game edukasi perlu dapat mengembangkan sikap aktif sehingga secara tidak langsung mengembangkan syaraf-syaraf motorik kasar dan motorik halus pada pemain.

3. Efektif dan Efisien

   Sebuah game edukasi tentunya perlu punya dampak dalam kehidupan para pemainnya, misal pada *Brain Challenge*, lama kelamaan pemain dapat menjadi makin cepat berpikir, atau makin analitif. Selain itu, dari sisi waktu, pemain yang ingin segera belajar tidak harus menunggu kelas dimulai, bisa dimainkan kapan saja dan di mana saja sehingga game edukasi bisa dikatakan efisien.

4. Kreatif

   Melalui game edukasi, setidaknya pemain bisa memikirkan beberapa cara lain untuk menyelesaikan masalah, dan tidak harus selalu menggunakan satu macam cara, sehingga merangsang pemain untuk kreatif dan tidak hanya menunggu orang lain menyelesaikannya dan menirunya.

5. Menyenangkan

   Game edukasi tetap adalah game, jangan sampai kehilangan unsur menyenangkan dalam game tersebut karena padatnya isi materi dan monotonnya gameplay. Jadi game sebisa mungkin bisa membuat pemain

makin pandai perlahan tapi pasti tanpa meninggalkan kesenangan bermain. Jadi sebaiknya game edukasi tanpa disadari bermanfaat untuk mengembangkan IQ, EQ danSQ .

2.4.3 Manfaat Game Edukasi

Adapun manfaat dari Game Edukasi yaitu:

1. **Mampu melatih konsentrasi**

   Semakin tinggi usia pemain maka semakin terbatas daya tangkapnya. Oleh karena itu permainan dan pengajaran yang menggunakan alat dan media yang baik dan tepat akan membantu mempertahankan daya tangkapnya

2. **Mengajarkan sesuatu dengan lebih cepat dengan waktu relatif singkat**

   Bila pelajaran hanya disampaikan dengan kata-kata saja, mungkin bisa tersampaikan atau salah paham. Namun dengan bantuan alat dan media yang baik seperti game edukasi, guru bisa menjelaskan dalam waktu cepat dan mencapai indikator kebnerhasilan belajar lebih cepat.

3. **Menambah daya paham dan ingatan**

   Dalam menjelaskan sesuatu jika menggunakan media yang tepat tentu akan lebih mudah dimengerti dan memperdalam pengalaman belajar serta ingatan anak. Melalui indera penglihatan dan pendengaran anak yang berusia dini dapat memahami perbedaan arti, warna, serta bentuk melalui game edukasi. Ini salah satu kegunaan game edukasi bagi para pemain yang berusia muda.

4. **Membuat proses belajar menyenangkan**

   Cara mengajar yang monoton tentu akan membosankan. Tetapi bila didisampaikan dalam bentuk yang berbeda, media yang berbeda tentu akan menyenangkan dan mampu membangkitkan motivasi belajar .

5. **Membangkitkan emosi**

   Menyampaikan suatu materi dengan media-media yang menarik tentunya akan lebih berhasil daripada menggunakan ceramah saja. Dengan media yang menarik tentu akan membangkitkan emosi , perhatian pada materi dan juga pada media tersebut. Misal ada game edukasi yang memiliki kisah dan cerita, tentu akan lebih mudah menyerap inti yang ingin diajarkan.

6. **Mampu Mengatasi Keterbatasan Bahasa**

   Perbedaan kebudayaan sering menimbulkan kesalahpahaman, namun dengan media mampu mengatasi kesalahpahaman akan keterbatasan untuk mengerti suatu bahasa. Beberapa game edukasi juga diarahkan untuk mempelajari bahasa.

7. **Meningkatkan kemampuan berkomunikasi**

   Dengan permainan edukatif pasti merangsang para pemain untuk berhubungan dan berkomunikasi dengan orang lain. Paling tidak dengan permainan edukasi akan menimbulkan banyak pertanyaan dan imajinasi yang tentunya akan ditanyakan pada guru, orangtua atau teman sesama pemain.

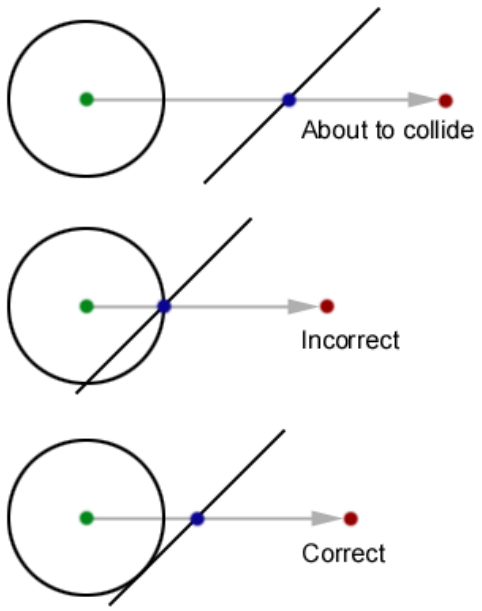
2.5 Algoritma *Collision Detection*

Algoritma *Collision Detection* adalah sekumpuluan perintah dalam computer yang digunakan untuk mengatur benturan suatu objek dengan objek lain agar terlihat lebih rapih dan realistis, contohnya adalah beririsan antara dua benda padat. Didalam *game*, *collision* merupakan hal yang penting untuk mengatur segala bentuk benturan diantaranya digunakan untuk membuat mobil tidak akan menembus jalan/jembatan, terjadinya ledakan bila mobil masuk ke jurang/ tabrakan.

Pada awalnya video games adalah game berbasis 2D dan memiliki karakter yang terbatas sehingga collision detection secara manual dilakukan untuk mendeteksi adanya tabrakan. Namun, ketika dunia game mulai merambah dunia 3D dan dengan

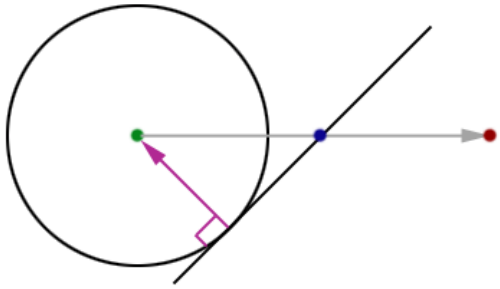
bertambahnya karakter pada game maka dibutuhkan algoritma pengecekan collision detection. Jika tidak ada collision detection maka karakter pada game bisa berjalan menembus tembok dan semua pembatas lainnya.

Untuk menerapkan Algoritma *Collision Detection*, *Game Engine* biasanya sudah menyediakannya dalam bentuk *library*, pengguna hanya tinggal memanggil *library* untuk *collision* tersebut pada objek game yang diinginkan. Seperti pada *Game Engine* Unity yang memberikan *collider* ke setiap game objek yang akan dibuat sebagai *collision*.



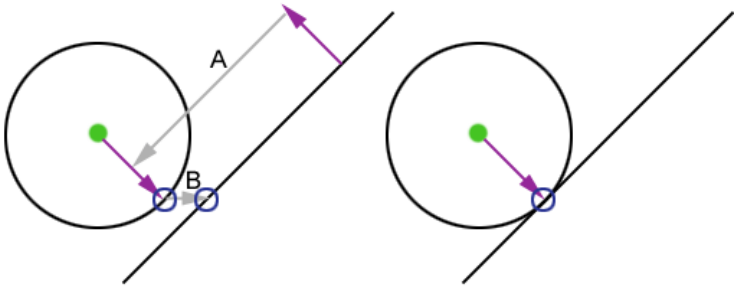
Gambar 2.1 Sebuah lingkaran menabrak sebuah permukaan(garis)

Pada gambar 2.1 bisa melihat bahwa lingkaran bisa menabrak permukaan. Namun tabrakan lingkaran dengan permukaan tidak semudah yang bayangkan. Tabrakan tersebut bisa saja terjadi tidak benar yaitu ketika lingkaran sudah melewati garis.



Gambar 2.2 Tabrakan yang benar antara garis dan lingkaran

Pada gambar 2.2, bisa dilihat bahwa vector dibentuk dari titik perpotongan dan pusat dari lingkaran, adalah terjadi bersamaan dengan garis normal dari garis. bisa membalik proses ini untuk memutuskan titik pada garis yang terdapat pada lingkaran yang akan berpotongan dengan garis sebelum melakukan tabrakan. Pada gambar 3 akan diperlihatkan lebih jelas lagi



Gambar 2.3 Menggunakan garis normal dari garis untuk menemukan titik potong lingkaran

Gambar 2.3 menunjukkan bahwa garis normal dari garis bisa digunakan untuk mendapatkan titik pada permukaan lingkaran yang akan beririsan dengan garis. Disebut dengan *titik potong lingkaran.*

Untuk mendapatkan titik potong lingkaran harus menentukan garis normal dari garis ke jari-jari lingkaran lingkaran. Lalu tarik garis abu-abu dari garis normal menuju garis jari-jari yang disebut sebagai garis A dan akan melihat bahwa hasilnya adalah vector titik potong lingkaran (yang berupa garis yang berlawanan arah dari garis normal dan diambil dari titik tengah lingkaran).

Dari titik yang didapatkan, pindahkan velocity vector (garis B berwarna abu-abu), hingga memotong garis. Sehingga lingkaran akan menuju garis dengan baik.

2.6 Game Engine

Game engine adalah perangkat lunak yang dirancang untuk membuat dan mengembangkan video game. Fungsi utama yang diberikan oleh game engine meliputi rendering untuk 2D atau 3D graphic, collision detection, sound, scripting, animasi, artificial intelligence, networking, memory management, threading dan scene graph. Game engines memberikan perangkat untuk visual development dengan tambahan komponen perangkat lunak yang dapat dipakai berulang kali. Perangkat ini pada umumnya memberikan integrated development environment yang dapat mempermudah, serta mempercepat pengembangan game. Game engine juga sering disebut juga "Game Middleware". Game Middleware adalah software yang didalamnya sudah terdapat elemen-elemen yang dibutuhkan oleh seorang game developer, kebanyakan dari game middleware memberikan fasilitas guna menunjang pengembangan game dengan mudah seperti graphics, sound dan Artificial intelligence.

Berikut beberapa *Game engine* yang beredar sekarang ini :

1. *OGRE*
2. *Panda 3D*
3. *crystal space*
4. *blender Game engine*
5. *OPEN SceneGraph*
6. *C4 Engine*
7. *Game Maker*
8. *torque Game engine*
9. *3DGamestudio*
10. *DXStudio*
11. *neoExis engine*
12. *esenthel engine*

2.6.1 Unity 3D

Unity 3D merupakan sebuah *game* engine yang dapat bekerja secara multi platform. Digunakan sebagai alat untuk membangun sebuah video *game* untuk *plugin web, desktop platform, console* dan perangkat*mobile.* Dikembangkan awalnya hanya untuk mensuport OS X pada tahun 2005 hingga saat ini menjadi *game engine* yang multi *platform.*

Perkembangan Unity saat ini telah sampai pada versi Unity 4.2.1, yang dikeluarkan pada September 2013. Saat ini dapat mensuport pembuatan *game* pada iOS, Android, Windows, Blackberry 10, OS X, Linux, *web browser*, Flash, Playstaation 3, Xbox 360, Windows Phone 8, dan Wii U.

Scripting pada Unity 3D dikembangkan melalui Mono, sebuah *software open-source* yang mengimplementasikan .NET Framework. Programmer dapat menggunakan UnityScript (sebuah bahasa yang dikembangkan dengan ECMA Script syntax, atau dikenal dengan JavaScript), C#, atau Boo.

2.7 OOP (Object Oriented Programming)

OOP adalah teknik memodelkan suatu sistem dunia nyata dalam perangkat lunak berdasarkan objek.Objek tersebut adalah konsep inti.Suatu objek adalah sebuah perangkat lunak entitas atau konsep model dunia nyata.

Ketika sebuah program berjalan, objek individu biasanya tidak dapat berdiri sendiri. Mereka termasuk ke sebuah koleksi dari objek lain yang serupa yang merupakan anggota dari grup yang sama, atau *class*. Sebuah program akan terbentuk dari banyak *class* yang berbeda, setiap *class* terbentuk dari objek-objek yang serupa.
Beberapa sistem perangkat lunak orientasi objek akan memiliki sifat berikut:

1. Abtraksi dengan objek.

   Abtraksi adalah mekanisme yang memungkinkan kompleks, situasi dunia nyata dapat diwakili menggunakan model yang disederhanakan. Orientasi objek abstrak dunia nyata didasarkan pada objek dan interkasi antar objek lainnya.

2. Enkapsulasi *class*.

   Enkapsulasi adalah proses menyembunyikan semua bagian rinci sebuah objek dari dunia luar.

3. Interaksi lewat pesan.

   Untuk memenuhi suatu perintah, objek butuh berinteraksi dengan objek lain. Interaksi dapat antara objek di *class* yang sama, atau objek di *class* lain. Interaksi ini ditangani dengan cara mengirimkan pesan ke objek lain untuk melewati informasi atau meminta aksi.

4. Masa hidup objek.

   Semua objek memiliki masa hidup.Mereka dibuat dan diinisialisasi sebagaimana mereka dibutuhkan pada saat program dijalankan, hidup dan membawa keluar fungsi mereka, dan akhirnya dihancurkan.Sementara mereka ada, mereka mempertahankan identitas mereka sendiri dan kondisi. Banyak objek yaitu contoh dari *class* yang sama bisa hidup pada waktu tertentu. Setiap objek memiliki atribut yang berbeda dari yang lain yaitu contoh objek dari *class* yang sama.

5. Hirarki *class*.

   Dalam desain orientasi objek, *class* objek tersebut diatur ke dalam hirarki yang memodelkan dan menggambarkan hubungan antar *class*.Hubungan yang paling sederhana adalah sebuah asosiasi.

6. Polimorfisme

   Polimorfisme adalah karakteristik sistem orientasi objek. Ketika warisan digunakan untuk memperpanjang *class* umum ke *class* yang lebih khusus, biasanya akan mencakup memperluas beberapa perilaku dari *class* umum. *Class* khusus sering akan menerapkan perilaku yang agak berbeda dengan *class* umum, tetapi nama yang digunakan mendefinisikan perilaku akan menjadi sama. Adalah penting bahwa contoh yang diberikan dari sebuah objek menggunakan perilaku yang benar, dan *property* polimorfisme memungkinkan

ini terjadi secara otomatis dan mulus. Polimorfisme sebenarnya lebih mudah digunakan daripada dijelaskan.

2.8 UML (Unified Modeling Language)

UML adalah penerus gelombang berorientasi objek alnalisis dan desain (OOA & D) metode yang muncul di akhir ‘80-an dan ’90-an awal. Hal yang paling langsung menyatukan metode Booch, Rumbaugh (OMT), dan Jacobson, tapi jangkauan lebih luas dari itu. UML pergi melalui proses standarisasi dengan OMG (*Obyek Manajement Group*) dan sekarang menjadi standar OMG.

UML disebut bahasa permodelan, bukan metode.Sebagian besar metode terdiri, setidaknya pada prinsipnya, kedua model sebuah bahasa dan proses.Bahasa pemodelan adalah notasi (terutama grafis) bahwa metode digunakan untuk mengekspresikan desain. Proses ini menyarankan mereka apa langkah yang harus diambil dalam melakukan desain.

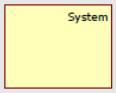
Bagian proses dalam buku banyak metode yang agak samar. Selain itu, kebanyakan orang, ketika mereka mengatakan bahwa mereka menggunakan metode, menggunakan bahasa pemodelan, tapi jarang mengikuti proses.Jadi dalam banyak hal pemodelan bahasa adalah bagian paling penting dari meode ini.Hal ini tentu bagian penting untuk komunikasi. Jika anda ingin mendiskusikan desain anda dengan seseorang, itu adalah bahasa pemodelan yang anda berdua perlu pahami, bukan proses yang digunakan untuk sampai ke desain tersebut.

2.8.1 Use Case Diagram

Use-case diagram adalah gambaran graphical dari beberapa atau semua aktor, use-case, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. Use-case diagram menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

Use-case diagram dapat digunakan selama proses analisis untuk menangkap requirement sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, use-case diagram berperan untuk menetapkan perilaku (behavior) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa use-case diagram. Kebutuhan atau requirements system adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model use-case yang menggambarkan fungsi sistem yang diharapkan (use-case), dan yang mengelilinginya (actor), serta hubungan antara actor dengan use-case (use-case diagram) itu sendiri.

Komponen yang terdapat di dalam *Use Case Diagram*, yaitu:

SYMBOL	NAME OF SYMBOL	EXPLANATION
	Actor	Accessing use case
	Use Case	Show what the system do
	Association	Relate the actor with use case
	System Boundary	Show boundary between system and its environment

Gambar 2.4 *Use Case Diagram*[2]

2.8.2 Activity 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. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktur. Sangat bermanfaat apabila membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.

Activity diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram.[2]

2.8.3 Sequence Diagram

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu. Sequence Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan use case diagram. Sequence diagram juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case. Sequence diagram juga dapat merubah atribut atay method pada class yang telah dibentuk oleh class diagram, bahkan menciptakan sebuah class baru. Sequence diagram memodelkan aliran logika dalam sebuah system dalam cara yang visual.

Sequence diagram biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi method didalam sebuah system. Sequence diagram biasanya dipakai untuk memodelkan :

1. Deskripsi tentang system yang ada pada sebuah / beberapa use case pada use case diagram, yang menggambarkan hubungan antara actor dan use case diagram.
2. Logika dari method (operation, function atau procedure).
3. Logika dari service (high level method).

2.9 Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean[5].

2.9.1 Pengujian *Black Box*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* menungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* bukan merupakan alternatif dari teknik *white box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white box*. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

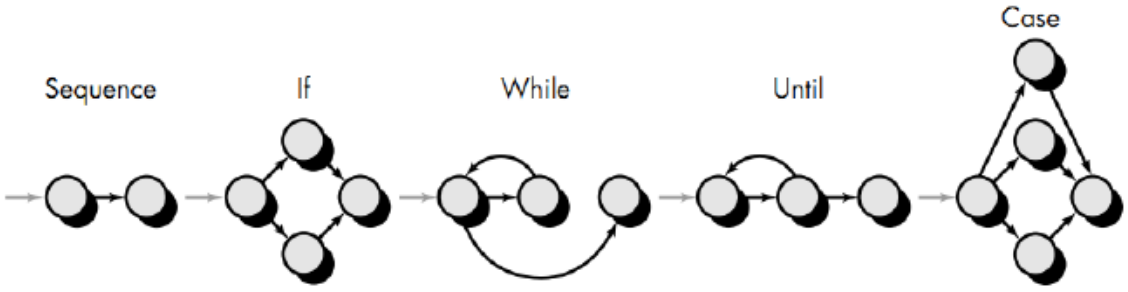
1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan dalam interface
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

2.9.2 Pengujian *White Box*

Pengujian *white box*, yang kadang-kadang disebut pengujian *glass box*, adalah metode desain *test case* yang menggunakan struktur kontrol desain procedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white box*, perekayasa sistem dapat melakukan *test case* sebagai berikut :

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*
3. Mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka
4. Menggunakan struktur data internal untuk menjamin validitasnya.

Pengujian *white box* yang berupa notasi diagram alir dapat dilihat pada gambar 2.



Gambar 2.5 Notasi Diagram Alir

2.10 Skala Likert

Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu

pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti:

a. Sangat tidak setuju
b. Tidak setuju
c. Netral
d. Setuju
e. Sangat setuju

Selain pilihan dengan lima skala seperti contoh di atas, kadang digunakan juga skala dengan tujuh atau sembilan tingkat. Suatu studi empiris menemukan bahwa beberapa karakteristik statistik hasil kuesioner dengan berbagai jumlah pilihan tersebut ternyata sangat mirip.

Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Empat skala pilihan juga kadang digunakan untuk kuesioner skala Likert yang memaksa orang memilih salah satu kutub karena pilihan "netral" tak tersedia[7].