

BAB II

LANDASAN TEORI

II.1. Game

II.1.1. Perkembangan Game

Pengertian *game* adalah suatu sistem atau program di mana satu atau lebih pemain mengambil keputusan melalui kendali pada obyek di dalam *game* untuk suatu tujuan tertentu. *Game* pada awal tahun 1970-an merupakan awal dari perkembangan *game* dunia, di mana *game* yang dibuat terdiri atas teks dan *grafik* standar. Beberapa *game* pada awal perkembangannya antara lain (Dewi lestari, 2011:1).

1. Teks Game

Jenis *game* ini hanya terdiri atas teks, tidak ada grafik apa pun dalam game ini. *User* berinteraksi dengan mengetikan teks contoh: *The Adventurer's Museum, The Castle, Adventure in The Island.*

2. Grafik Game

Jenis *game* ini sudah menggunakan grafik meskipun masih dalam bentuk grafik standar yaitu VGA, SVGA, dll. Contoh: *The Murderer*. Pembuatan sebuah game tidak terlepas dari mamfaat yang diterima oleh para pemain *game* yang menggunakannya. Beberapa mamfaat yang didapat saat memainkan *game* antara lain:

a. *Entertainment*

Game ini hanya diutamakan untuk permainan yang menghibur. Sebagian *game* semacam ini dimanfaatkan sebagai alternatif hiburan sehingga dapat menyegarkan kembali pikiran yang stress

b. *Expand Skill*

Game ini juga dibuat untuk lebih dikhususkan melatih ketangkasan dan kecepatan berfikir pemainnya. *Game* ini tidak hanya terbatas pada *game* personal computer, namun meliputi *platform game* lainnya.

c. *Education*

Ada beberapa *game* dibuat untuk mendukung proses pembelajaran dengan konsep belajar sambil bermain *game* jenis ini banyak ditemukan di beberapa *platform game* seperti personal computer, playstation dan lainnya. Contoh: *Edu Games Learning*.

d. *Embed Messages*

Ada beberapa *game* yang sengaja dibuat untuk menyampaikan pesan tertentu, misalnya kebenaran selalu menang melawan kejahatan, jangan mudah menyerah, dan sebagainya. Biasanya *game* seperti ini memiliki unsur penceritaan yang kompleks

II.1.2. Game Berdasarkan Jenis (*Genre*)

Pada dasarnya jenis-jenis *game* yang lebih dikenal dengan istilah "*Genre game*". *Genre* juga berarti *format* atau gaya dari sebuah *game*. *Format* sebuah

game bisa murni sebuah *genre* atau bisa merupakan campuran (*hybrid*) dari beberapa *genre* lain. Beberapa game berdasarkan jenis atau genre antara lain:

1. *Maze Game*

Jenis *game* ini adalah jenis game yang paling awal muncul. Contoh yang paling populer di Indonesia adalah *game* Pacman dan Digger. *Game* jenis ini sangat mudah dimainkan.

2. *Board Game*

Jenis *game* ini sama dengan *game board* tradisional, seperti Monopoly. Umumnya *game* ini lebih menekankan kepada kemampuan komputer menjadi lawan tanding dari pemain. Ini melibatkan kemampuan AI (*Artificial Intelligence*) yang andal untuk bisa menjadikan *game* ini menantang pemain dengan baik. Contoh: *Chess*, *Chess Master 2000*, *Monopoly*, *Scrabble*.

3. *Card Game*

Hampir sama dengan board *game*, jenis ini tidak memberikan perubahan berarti dari *game* versi tradisional yang sejenis. Variasi yang ada adalah kemampuan *multi player* dan tampilan yang lebih bervariasi dari versi tradisional. *Game* ini termasuk game yang muncul pada awal *game* komputer. Contoh: *Hearts*, *Poker*, dan *Solitaire*.

4. *Trading Card Game*

Jenis *game* ini jarang masuk ke Indonesia. Ada beberapa stasiun televisi di Indonesia yang menayangkan kartun tentang *trading card* ini. Versi *game*

elektroniknya sangat digemari di luar negeri. Contoh: *Battle Card Pokemon, Yu Gi Oh!, Duel Masters, Magic The Gathering.*

5. *Quiz Game*

Jenis *game* ini juga jarang di Indonesia. *Game* ini sederhana dalam cara bermain yaitu hanya perlu memilih jawaban yang benar dari beberapa pilihan jawaban. Biasanya pertanyaan yang diberikan memang memiliki topik tertentu tetapi ada juga yang memiliki topik yang beragam. Contoh: *Classroom Millionaire, Deal or No Deal.*

6. *Puzzle Game*

Game jenis ini memberikan tantangan kepada pemainnya dengan cara menjatuhkan sesuatu dari sisi sebelah atas ke bawah. Susunan ini dilakukan secepat dan sebaik mungkin. Dalam perkembangannya, jenis ini membebaskan cara bermainnya. Contoh: *Tetris, Magic Inlay, Adventure Inlay, Chip Challenge*

7. *Shooting Game*

Secara standar bahwa semua *game* yang bertipe atau dimainkan dengan cara menembak objek termasuk dalam jenis ini. Baik *game* yang bertipe dua dimensi atau pun tiga dimensi. Contoh: *Deer Hunt, Hunting Unlimited*

8. *Shoot Them up*

Jenis ini banyak diminati karena mudah dimainkan. Pada awalnya bentuk *game* ini adalah *game* dua dimensi namun berkembang menjadi tiga dimensi. Contoh: *Twin Bee, Astro Raid, Arcanoid, Macross, Blackhawk Striker 2.*

9. *Adventure Game*

Game ini adalah game petualangan. Biasanya pemain menuju ke suatu tempat dan menemukan banyak hal dan peralatan yang akan kita simpan. *Game* jenis ini tidak berfokus kepada pertarungan. Umumnya *game* ini lebih menekankan kepada pemecahan misteri daripada pertarungan sampai mati. Contoh: *Beyond Good and Evil, Sam & Max, Maniac Mansion*.

10. *Side Scroller Game*

Game jenis ini sangat terkenal dengan banyaknya *game* yang dikeluarkan. *Game* ini didasarkan kepada dua dimensi dimana pemain bergerak ke sepanjang jalur permainan ke satu arah dan menyelesaikan tugasnya. Contoh: *Prince of Persia, Sonic the Hedgehog, Super Mario Bros, Megaman Series*

11. *Fighting Game*

Game ini memberikan kesempatan untuk pemain dalam bertarung menggunakan berbagai kombinasi gerakan dalam pertarungan. Contoh: *Street Fighter, Samurai Showdown, Mortal Kombat, Tekken*.

12. *Sport Game*

Game jenis ini hanya dibuat berdasarkan jenis cabang olahraga baik secara dua dimensi ataupun tiga dimensi. Contoh: *Winning Eleven, NBA, NFL, Virtual Tennis*.

13. *Racing Game*

Game ini memberikan permainan lomba kecepatan dari kendaraan yang dimainkan oleh pemain. Contoh: *Need For Speed, Driver, Taxi, Test Driver*.

14. *Simulation Game*

Game ini berfokus kepada simulasi penerbangan. Simulasi yang diberikan meniru kondisi penerbangan sebenarnya, baik kondisi pesawat dan peralatannya maupun kondisi pemandangan. Contoh: *Microsoft Flight Simulator, IL-2Sturmovik, Apache Air Assault*

15. *Real Time Strategy(RTS) Game*

Pada game jenis ini pemain harus melakukan berbagai gerakan sesuai dengan strategi yang dibuat. Contoh: *Warcraft, Starcraft*.

16. *Role Playing Game (RPG)*

Game ini pemain akan berperan menjadi sebuah karakter. Contoh: *Blade of Sword, Kingdo m Hearts*. *Game* jenis ini yang penulis buat untuk menjadi sebuah software game

II.1.3. Elemen-Elemen Game

Setiap *game* tertentu mempunyai ciri khas dari masing-masing pembentuk *game* tersebut menjadi sebuah game yang unik dan mudah diingat masyarakat. Misalnya Mario Bros, hampir semua pemain *game* akan mengenali bahwa seseorang sedang memainkan game tersebut hanya dengan mendengarkan lagu dari temanya.

Namun setiap *game* memiliki elemen-elemen penting yang membangun *game* tersebut menjadi sebuah *game*, dan bukan sekedar perangkat lunak (*software*) biasa. Sebagian *game* yang diproduksi pada masa sekarang memiliki semua elemen-elemen tersebut (JP Marbun,2010:3).

II.1.4. Elemen Pokok Game

Fungsi tampilan dari berbagai *game* ada yang serupa atau mirip satu sama lain, baik fungsi maupun bentuknya. Walaupun jenis *game* atau genre satu sama lain berbeda, namun elemen itu tetap memiliki fungsi dan maksud yang sama. Sama halnya dengan buku, apa pun isi dari buku tersebut tetap memiliki aturan menampilkan judul, kata pengantar, nomor halaman, dan sebagainya. Berikut adalah keterangan berbagai elemen-elemen yang ada dalam sebuah *game* yang terdiri atas (JP Marbun,2010:3).

1. *Title* (Judul)

Judul adalah elemen penting dalam pengembangan sebuah *game*. Membuat sebuah judul sebuah *game* tidaklah mudah. Judul harus singkat namun menyiratkan isi dari maksud *game* tersebut.

2. *Title Screen* (Layar Judul)

Judul memang merupakan hal yang penting, namun untuk membuat judul lebih menarik adalah tampilan grafis yang sesuai dengan judul *game* tersebut. Pada *game* masa kini, sebelum *title screen* muncul terkadang sudah ada beberapa animasi pembuka yang muncul.

3. *Credits* (Daftar Nama Individu yang Terlibat)

Maksud dari *credits* adalah menunjukkan bahwa yang ikut serta dalam membuat *game* tersebut dengan penghargaan kepada pihak yang membantu dalam pengembangan *game* tersebut. Pada *game* masa kini *credits* berisikan banyak *individu* yang terlibat dimulai dari *designer*, *programmer*, *sound engineer*, *two dimension and three dimension artist* dan sebagainya. Di sini juga terdapat kata ucapan terima kasih yang disampaikan pada pihak-pihak tertentu yang dianggap telah membantu dan berjasa kepada pengembang *game* tersebut.

4. *Cutscene/Intro* (Pengenalan Cerita)

Tujuan dari *cutscene/intro* adalah untuk membuka awal perkenalan sebelum pemain memulai *game*, suatu penting di dalam alur cerita *game* maupun suatu transisi antara tingkat satu dengan tingkat selanjutnya. Dengan adanya *cutscene/intro* ini maka pemain secara perlahan akan masuk ke alur permainan atau kepada bagian selanjutnya dari *game*.

5. *Control Panel*

Tidak kalah pentingnya adalah control panel dimana pemain dapat mengatur permainan. Contoh: memulai *game* baru, menghentikan sementara *game*, mengulangi *game*, menyimpan *game*, dan sebagainya.

6. *User Interface* (Antarmuka)

Berbeda dengan control panel yang hanya aktif ketika dibutuhkan. Sementara user interface tetap aktif selama pemain memainkan *game* dan tidak hanya tampilan pada layar, *keyboard* dan *mouse* yang digunakan

adalah *user interface*. Untuk layar maka *user interface* adalah simbol atau tampilan khusus yang membantu pemain memainkan *game*. Contoh pada *game jenis First Person Shooter (FPS)* *user interface*-nya adalah jumlah peluru dan nyawa yang ada pada tampilan layar.

7. *Help* (Bantuan)

Tidak semua *game* dapat dimengerti dengan cepat oleh pemain *game* tersebut. Untuk itulah elemen *help* ini menjadi salah elemen yang cukup penting. Di dalam *help* ini semua instruksi yang bisa membantu pemakai ditampilkan dengan cepat dan sederhana sehingga mudah dipahami dan bisa langsung digunakan.

8. *Mouse Pointer* (Ikon Penunjuk Mouse)

Bentuk *pointer mouse* bisa bervariasi sesuai dengan tema dari *game* yang dibuat. Terkadang bentuk *pointer* bisa berubah jika menyentuh area tertentu di dalam *game* (disebut *hotspot*).

9. *Music and Sound* (Musik dan Efek Suara)

Semua jenis *game* saat ini sewajarnya dilengkapi musik dan suara yang cocok dengan tema dari *game* yang akan dikembangkan. Namun jika *game* yang dikembangkan tidak memiliki suara dan musik akan terkesan tidak *profesional* dan kurang bermutu.

10. *Art* (Seni)

Pada dasarnya *game* yang baik dihasilkan dari gambar yang memang memiliki seni yang baik pula. Terkadang *game* yang alur permainannya kurang baik dapat tertutupi sebagian oleh tampilan grafis yang bagus.

Gambar juga memegang peranan penting dalam elemen *game* yang akan dikembangkan. Itu sebabnya kenapa *game* masa kini khususnya di bidang *art* (seni) dipegang oleh beberapa orang yang memiliki keahlian tersendiri.

11. *Storyline* (Cerita)

Alur cerita tidak digunakan pada semua *game*, hanya jenis *game* tertentu yang menggunakan alur cerita misalnya *adventure game*, *RPG game*, *Real Time Strategi* (RTS) *game* dan *action game*.

12. *Playability* (Kemampuan untuk Dimainkan)

Game yang dikembangkan harus mampu dimainkan oleh seorang pemain pemula, dalam arti cukup mudah dan cepat dimengerti bagaimana cerita dan *game* itu berjalan. Terkadang sebagian *game* yang sukses dipasaran lebih berdasarkan faktor ini dibandingkan faktor lain seperti grafis yang hebat.

13. *Levels* (Tingkatan)

Hampir setiap *game* memiliki beberapa level untuk dimainkan. Maksudnya supaya pada tahap awal pemain bisa menguasai permainan dengan baik. Pada tingkat selanjutnya pemain tidak akan merasa bosan karena tantangan akan meningkat

14. *Demo*

Pada dasarnya *game* muncul dengan mode demo terlebih dahulu sehingga bisa dimainkan sebelum *software* aslinya muncul di pasaran. Memang tidak semua *game* menyediakan *demo game*, tetapi keuntungan promosi bisa didapatkan dengan menyediakan *demo game*.

15. *Invincible Mode* (Mode Tak Terkalahkan)

Mode tak terkalahkan ini bertujuan untuk pemeriksaan dari *game* itu sendiri di mana beberapa pihak diizinkan untuk menjalankan *game* dalam bentuk *invincible mode* di mana karakter *game* pemain tidak bisa kalah atau mati.

16. *Exit Screen* (Layar Keluar)

Layar keluar ini merupakan elemen penting dalam pengembangan *game*. *Game* yang baik biasanya memiliki *exit screen* pada *game* yang dikembangkan.

17. *Documentations* (Dokumentasi)

Game yang baik tidak dianggap selesai tanpa adanya dokumentasi yang cukup. Dokumentasi ini ditujukan kepada para pemain. Pada dokumentasi biasanya berisikan cara memainkan *game*, menangani masalah yang timbul dari segi *software* maupun *hardware*.

18. *Copyright* (Hak Cipta)

Hal paling penting dalam pengembangan sebuah perangkat lunak *game* ini adalah hak cipta yang diberikan oleh pembuat *game*. Hal ini untuk mencegah masalah yang berkaitan dengan hak cipta di masa depan.

19. *Setup* (Pemasangan/instal)

Sudah menjadi kewajiban bagi pembuat perangkat lunak untuk membuat *game* dalam bentuk *setup*. Dengan program ini maka *game* akan dikompilasi terlebih dahulu menjadi paket yang padat

II.1.5. Elemen Tambahan

Game Umumnya game ini ada untuk tipe *shareware*. *Game tipe shareware* biasanya menjadi pilihan bagi pengembang tunggal atau pun pengembang kecil. Elemen berikut ini biasanya ada pada versi cetak maupun versi *file* elektronik yang dimuat di dalam CD *game* itu sendiri yaitu:

1. *Readme.txt*

Pada file ini *user* akan bisa mendapatkan informasi lengkap bagaimana membuka dan menginstalnya ke dalam komputer. Selain itu bisa juga berisi informasi mengenai websitedan panduan singkat untuk pemain jika mereka mencari dokumentasi lengkapnya.

2. *File Description* (Deskripsi File)

File ini merupakan deskripsi singkat tentang semua *game* yang dibuat untuk membantu pembaca dari sebuah situs membuat review singkat untuk daftar file yang mereka simpan di website.

3. *License Agreement* (Persetujuan Lisensi)

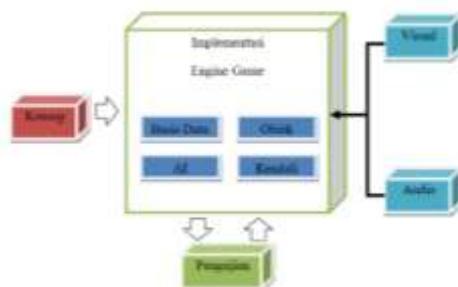
Pada umumnya pembuat game di Indonesia belum memiliki standarisasi untuk lisensi perangkat lunak, walaupun pada masa sekarang ini sebagian dari pengembang perangkat lunak di Indonesia sudah membuatnya.

4. *Order Form* (Formulir Pemesanan)

Pada file ini bisa memuat cara mendaftar dan memesan game. Jadi pemain akan dengan gampang mendapatkannya dan mengisinya. Umumnya para pengembang *shareware* membuat pilihan untuk langsung mencetaknya ke printer

II.1.6. Konsep Pengembangan Game

Dari elemen-elemen game tersebut dapat dibentuk menggunakan konsep pengembangan game secara garis besar. Bagan umum dari konsep pengembangan game dapat dilihat pada gambar di bawah.



Gambar II.1. Bagan Umum Konsep Pengembangan Game
(Sumber : JP Marbun, 2010)

Secara garis besar, konsep pengembangan *game* terbagi menjadi beberapa bagian, yaitu:

1. Perancangan Konsep *Game*

Pada bagian ini penulis menentukan ide dasar *game* secara keseluruhan dan jenis dari *game* yang dikembangkan yaitu jenis *Role Playing Game*(RPG). Dan pada bagian ini juga ditentukan tujuan dan sasaran *game* yang dikembangkan misalnya anak-anak, remaja atau dewasa.

2. Implementasi Engine Game

Engine game merupakan inti atau mesin utama dari *game* yang mengendalikan seluruh aktivitas dan interaksi yang terjadi dalam *game*. Dalam pengembangan *game* ini penulis menggunakan RPG Maker VX sebagai *engine game*. Biasanya terdapat empat komponen utama yang membentuk *engine game*. Komponen-komponen tersebut antara lain:

a. *Database* (Basis Data)

Basis data dalam *game* berguna untuk mendaftarkan dan menyimpan data tentang objek-objek maupun kejadian di dalam *game*. Misalnya daftar karakter, daftar peta, tingkatan, atau pun data yang disimpan ke dalamnya.

b. *Game Object* (Objek Game)

Objek *game* merupakan objek yang akan pemain kendalikan di dalam *game* tersebut, bagian ini mengimplementasikan objek *sprite game* dan lingkungan pada *game*. *Sprite* adalah objek yang dapat dikendalikan oleh pemain atau *Artificial Intelligence*. Objek lingkungan pada *game* yaitu latar belakang dan benda. Latar belakang yaitu wadah atau tempat di mana objek *sprite* akan dikendalikan. Benda adalah objek yang dapat berinteraksi dengan *sprite*.

c. *Artificial Intelligence* (Kecerdasan Buatan)

Artificial intelligence adalah kecerdasan buatan yang diberikan pada suatu objek agar dapat berlaku secara acak ataupun teratur. Tujuan pemberian kecerdasan buatan adalah agar pemain tidak sembarangan dalam mengambil keputusan dan mengembangkan semangat berkompetisi.

d. *Game Control* (Kendali Game)

Pengendalian *game* merupakan interaksi antara pemain dengan objek *game* sebagai hasil keputusan pemain tersebut. Kendali juga membatasi hal apa yang dapat dilakukan oleh objek *game*.

3. Penanaman Visual dan Audio pada Game

Pada bagian ini para seniman pada bidang grafis maupun suara bekerja. Para pengembang dapat mengolah dan menghasilkan aspek visual maupun audio dari *game* sehingga *game* tersebut dapat dinikmati oleh pemainnya. Aspek visual merupakan segala sesuatu yang dapat dilihat oleh pemain. Aspek visual dapat berupa model dua dimensi atau tiga dimensi. Model dua dimensi dibuat dengan garis-garis pada bidang, pewarnaan, masking, gradasi dan efek-efek percampuran warna. Model tiga dimensi dibuat dari rangka grid, proporsi, rendering dan pencahayaan.

4. Pengujian

Pengujian dilakukan untuk menjaga kualitas *game* yang akan dibuat. Pengujian ini meliputi pengujian terhadap *engine game*, pengujian interaksi dengan pemain melalui pengendali *game*, *debugging* dan perbaikan.

II.1.7. Para Pengembang *Game*

Sebuah *game* sebenarnya dapat dibuat oleh perorangan baik konsep cerita, grafik sampai kepada suara dan pengujiannya dapat dilakukan sendiri yaitu dengan menggunakan RPG Maker VX.

Pada dasarnya *game* dibuat dengan sekelompok tim pengembang yang bekerja pada masing-masing bidang misalnya desainer yang mengembangkan desain grafis *game* yang akan ditampilkan. Untuk mengenali siapa saja yang

terlibat dalam pengembangan *game* sampai dengan keahlian yang harus dimiliki oleh masing-masing (JP Marbun,2010:3).

II.1.8. Tim Utama Pengembang *Game*

Ada beberapa bagian yang terlibat dalam pengembangan *game* baik dari segi pekerjaan dan keahlian dari masing-masing yang dikuasai. Beberapa bagian tersebut antara lain:

1. *Designer*

Pada bagian desainer diharapkan bisa membuat alur cerita di dalam *game* menjadi hidup. Kemampuan komunikasi, pengetahuan, dan wawasan yang luas serta bekerja secara tim sangat membantu seorang *designer game* yang berperan dalam menjembatani antara dunia imajinasi yang dibuat ke dalam *game* dengan dunia nyata. Dalam bagian *designer* ini juga terdapat beberapa posisinya sendiri yang dapat berperan pada masing-masing posisi. Pembagian posisinya sendiri bisa beberapa macam seperti:

a. *Lead Designer*

Perancangan konsep dari keseluruhan sistem *game*, mulai dari konsep yang utuh sebuah *game*, pengawasan isi keseluruhan *game* sampai dengan strategi pemasaran *game*.

b. *Game Designer*

Pada posisi *game designer* ini dilakukan perancangan konsep alur cerita, karakter dan segala aspek yang berhubungan dengan *game* yang akan dikembangkan.

c. *Level Designer*

Perancang tingkatan dalam game juga merupakan peranan posisi yang penting dalam pengembangan sebuah game sehingga tidak menimbulkan efek bosan namun tidak lepas dari tujuan game yang dibuatnya.

2. *Programmer*

Dalam bidang ini sangat berpengaruh dalam bidang pengembangan *game*. Semua alur cerita dan gaya permainan serta grafik yang memukau digerakan oleh perintah-perintah kode yang dituliskannya. Seorang programmer menerjemahkan semua keinginan para pembuat *game* lainnya menjadi bentuk kode yang akan menggerakan *game*. *Game* itu sendiri adalah *software* dan programmer adalah orang yang bertanggung jawab penuh jika *game* itu tidak dapat berjalan semulus yang diminta.

Pembagian jabatan di posisi ini pun bervariasi. Posisi ini akan mengurus bagian-bagian khusus dari *programming* seperti:

a. *Graphic*

b. *Physics*

c. *Game Engine*

d. *Artificial Intelligence*

e. *Scripting*

f. *Sound*

3. *Artist*

Bagian ini mengkhususkan diri ke elemen grafik di dalam *game* baik grafik dua dimensi atau tiga dimensi. Beberapa posisi yang berhubungan dengan art adalah:

a. *Character Artist*

Bagian ini menciptakan berbagai model karakter dalam *game* yang akan bergerak seperti karakter pemain, pesawat, kendaraan dan sebagainya. Selain membuat model, posisi ini juga bertugas dalam menciptakan animasi game.

b. *Three Dimention Modeller*

Hampir sama dengan character artist tetapi posisi ini lebih memfokuskan diri ke bagian model yang tidak bergerak seperti lokasi maupun tempat dimana karakter yang dibuat akan bermain.

c. *Texture Artist*

Untuk melengkapi model dan karakter yang dibuat maka posisi inilah yang akan mengerjakan bagian *tekstur* untuk model. Tekstur yang dibuat akan memberikan sentuhan nyata dan hidup dari model dan karakter yang dibuat oleh artist lain.

4. *Sound/Audio Enginner*

Posisi ini tidak kalah pentingnya dengan posisi yang lain. *Game* tidak lagi menggunakan beberapa jenis suara, bahkan bisa menggunakan *efek orkestra* yang dibuat oleh seorang atau beberapa ahli musik. Ada beberapa yang bertugas secara khusus untuk menangani berbagai bidang sebagai berikut:

a. *Music Composer*

Pada posisi ini khusus membuat musik tema dan musik latar belakang dalam game yang akan dikembangkan sehingga menjadi lebih menarik dan tidak membosankan jika dimainkan.

b. *Sound Effects Engineer*

Posisi ini bertugas membuat efek suara seperti suara pukulan, tembakan, suara kendaraan, suara angin, dan lain-lain.

c. *Voice Engineer*

Beberapa *game* menggunakan percakapan dan tugas bagian ini adalah membuat rekaman suara untuk percakapan itu. Sering menggunakan *artist* yang asli untuk memberikan rekaman suara ke dalam game tersebut

II.2. Permainan Pac-man

Permainan *Pacman* adalah sebuah permainan beraliran labirin (*maze*) dengan konsep yang sangat sederhana diberikan suatu labirin dengan berbagai dinding rintangan di dalamnya, suatu titik yang merupakan posisi awal munculnya karakter Pacman, titik-titik yang harus dilewati, dan yang paling menarik beberapa musuh yang harus dihindari.

Objektif permainan *Pacman* adalah pemain diharuskan menggerakkan karakter Pacman menyusuri labirin tersebut melewati semua titik-titik yang telah dispesifikasikan sebelumnya tanpa satu kali pun bertemu dengan musuh. Titik-titik dalam labirin yang harus dilewati oleh Pacman digambarkan sebagai makanan Pacman.

Ketika permainan *Pacman* dimulai, selama beberapa detik musuh (digambarkan sebagai karakter yang berwarna biru, merah, merah muda, coklat muda, dan memiliki mata) akan terkurung di kotak di bagian tengah permainan, setelah beberapa detik, karakter-karakter tersebut akan bebas dan mulai berjalan secara acak (*random*).

Untuk meningkatkan kemewahan dan kesulitan permainan *Pacman*, karakter musuh dapat diberi suatu kecerdasan khusus agar pada setiap tahap permainan, karakter musuh tersebut dapat menentukan rute yang paling pendek (*minimum*) agar semakin mendekati posisi karakter *Pacman*.

Persoalan mendekati karakter *Pacman* ini dapat diselesaikan dengan berbagai macam cara, salah satu cara yang cukup sangkil (walaupun tidak selalu menghasilkan hasil yang paling optimal) adalah dengan menggunakan algoritma *greedy*. Persoalan karakter musuh *Pacman* dalam menentukan arah mana yang harus dijalaninya untuk semakin mendekati dirinya kepada karakter *Pacman* dapat dikategorikan sebagai persoalan optimasi, dan persoalan optimasi cukup efektif dipecahkan dengan menggunakan algoritma *Greedy*. Persoalan optimasi pada musuh *Pacman* ini termasuk persoalan minimasi, yaitu mencari *rute* terpendek saat ini dari posisi karakter musuh ke posisi karakter *Pacman* (Timotius Nugroho, 2010:8).

II.3. Algoritma Greedy

Algoritma *Greedy* merupakan metode yang paling populer dalam memecahkan persoalan optimasi. Hanya ada dua macam persoalan optimasi, yaitu

maksimasi dan minimasi. Algoritma *Greedy* adalah algoritma yang memecahkan masalah langkah per langkah. Algoritma *Greedy* membentuk solusi langkah per langkah. Pada setiap langkah, terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkahnya merupakan pilihan, untuk membuat langkah optimum lokal (*local optimum*) dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global (*global optimum*). Prinsip *Greedy* adalah "*take what you can get now*", mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (Muhammad Ardhin, 2011:2).

II.3.1. Skema Umum Algoritma Greedy

Persoalan *optimasi* dalam konteks algoritma *greedy* disusun oleh elemen-elemen sebagai berikut :

1. Himpunan kandidat

C.Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.

2. Himpunan solusi

Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.

3. Fungsi seleksi (*selection function*)

Fungsi ini dinyatakan dengan predikat seleksi. Merupakan fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

4. Fungsi kelayakan (*feasible*)

Fungsi ini dinyatakan dengan predikat layak. Fungsi kelayakan ini merupakan fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi objektif

ini merupakan sebuah fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Dengan kata lain, algoritma greedy melibatkan pencarian sebuah himpunan bagian, S , dari himpunan kandidat, C ; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimasi oleh fungsi *objektif*.

Adakalanya, solusi optimum global yang diperoleh dari algoritma *greedy* yang diharapkan sebagai solusi optimum dari persoalan, belum tentu merupakan solusi *optimum* (terbaik), tetapi solusi *sub-optimum* atau *pseudo-optimum*. Hal ini dikarenakan algoritma *greedy* tidak beroperasi secara

menyeluruh terhadap semua alternatif solusi yang ada dan terdapat beberapa fungsi seleksi yang berbeda, yaitu jika fungsi seleksi tidak identik dengan fungsi *objektif* karena fungsi seleksi biasanya didasarkan pada fungsi *objektif*. Sehingga harus dipilih fungsi yang tepat jika menginginkan algoritma menghasilkan solusi *optimal* atau nilai yang optimum. Jadi, pada sebagian masalah algoritma *greedy* tidak selalu berhasil memberikan solusi yang benar-benar optimum, tetapi algoritma *greedy* pasti memberikan solusi yang mendekati (*approximation*) nilai optimum (Vandy Putrandika, 2010:4).

II.4. Perangkat Lunak (Software) Pendukung

Adapun perangkat lunak yang digunakan dalam perencanaan perancangan sistem diantaranya menggunakan NetBeans IDE (versi 7.0) dan SQL Server 2000 sebagai basis data (database).

II.4.1. Sekilas tentang NetBeans IDE (versi 7.0)

NetBeans adalah *Integrated Development Environment* (IDE) berbasiskan Java dari *Sun Microsystems* yang berjalan di atas *Swing*. *Swing* sebuah teknologi Java untuk pengembangan aplikasi *Desktop* yang dapat berjalan di berbagai macam *platforms* seperti Windows, Linux, Mac OS X and Solaris. Netbeans merupakan *software development* yang *Open Source*, dengan kata lain *software* ini di bawah pengembangan bersama, bebas biaya

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas komunitas yang terus tumbuh, dan memiliki hampir

100 mitra. *Sun Microsystems* mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama.

Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun *Graphic User Interface* (GUI), suatu text atau kode editor, suatu *compiler* atau *interpreter* dan suatu debugger.

The NetBeans IDE adalah sebuah lingkungan pengembangan-sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java-namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2S E, web, EJB, dan aplikasi *mobile*). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan refactoring (Jubilee Enterprise, 2014:2).

II.5. UML (*Unified Modeling Language*)

UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *OO (Object-Oriented)*. *UML* sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*.

Metode *objek-oriented* banyak digunakan dalam pengembangan perangkat lunak. *Unified Modeling Language* (UML) adalah bahasa yang dapat digunakan untuk spesifikasi, visualisasi dan dokumentasi sistem *software* pada fase pengembangan. *UML* merupakan unifikasi metode-metode *Booch*, *OMT* dan *Objectory* serta beberapa metode lainnya, *de facto UML* merupakan standar bidang analisis dan desain sistem *objek-oriented*. *View* adalah jendela (*window*) yang merupakan aspek pandang *UML* terhadap sistem. *UML* memperkenalkan lima buah *view* untuk memandang sistem yaitu: *Use-Case View*, *Logical View*, *Componentview*, *Deployment View* dan *Concurrency View* (Eriksson dan Penker, 1998). *Booch* (1998) menyebut *view* ini sebagai arsitektur 4+1' dan menyebut *Concurrency View* sebagai *Process View*. Tahap pengembangan sistem perangkat lunak di dalam *UML* meliputi:

Analisis Kebutuhan (*Requirement Analysis*), Analisis Sistem (*Analysis*), Desain (*Design*), Implementasi (*Implementation*) dan *Testin* (Angga Dharma, 2011:13).

II.5.1. Diagram UML

Tabel II.1. Diagram UML

Major Area	View	Digram	Main Concept
Structural	Static View	Class Diagram	Class, Assosiation, Generalization, defendency, Realization, Interface
	Use Case View	Use Case Diagram	Use Case, Actor, Association, Extends, Include, Use Case Generalization,

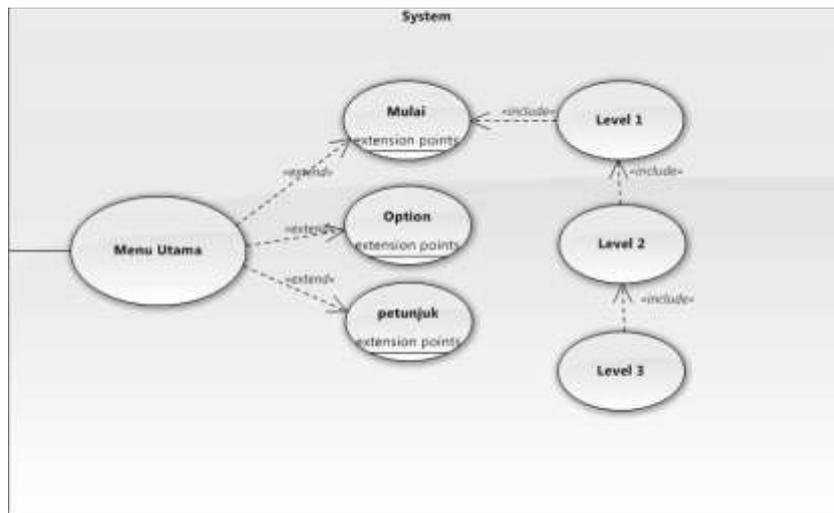
	Implementation View Deployment View	Componen Diagram Deployment Diagram	Component, Interface, Dependency, Location State, Event, Transaction, Fork, Join
Dinamic	State Machine View Activity View Interaction View	Statechart Diagram Activity Diagram Sequence Diagram Colaboration Diagram	State Event, Transition, Action State , Activity, Completion transition, Fork, join Colaboration, Interaction, Colaboration Role, Messages
Model Management	Model Management View	Class Diagram	Packages, Subsystem, Model
Extensibility	All	All	Contrait, Steretype, Tagged Values

(Sumber : Sri Dharmayanti, 2010)

II.5.1.1. Use Case Diagram

Use case diagram menggambarkan sebuah fungsi yang dibutuhkan oleh sebuah sistem. Dalam hal ini ada kondisi yang berbeda, yaitu tingkah laku dan domain sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara

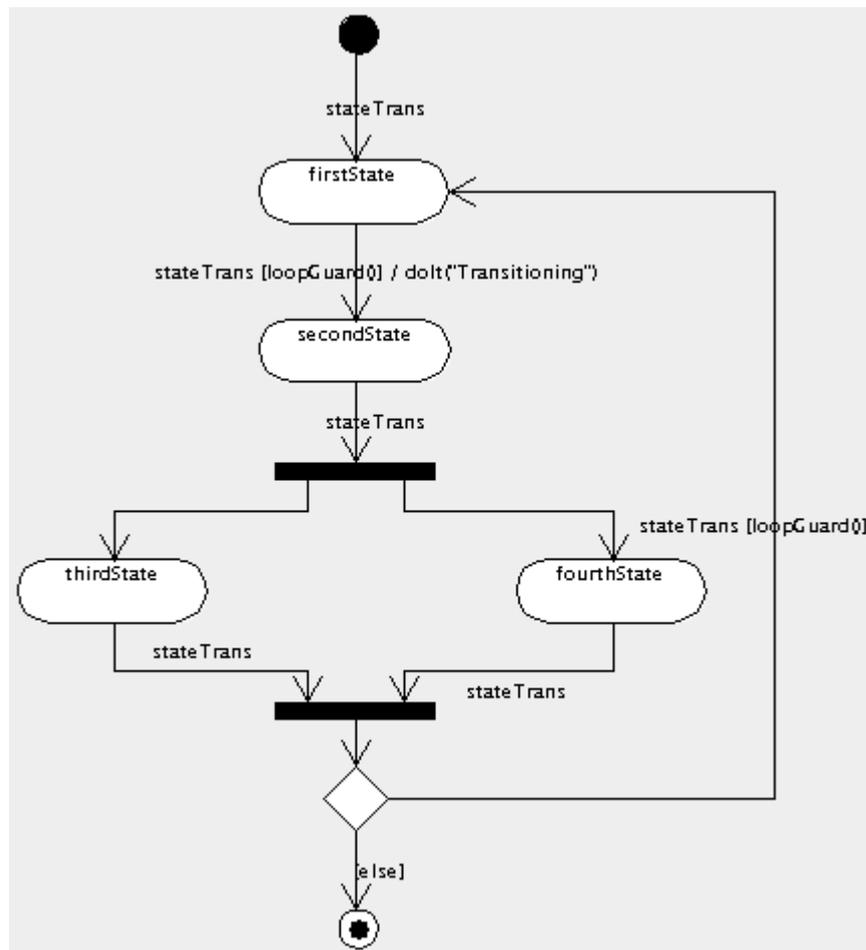
pengguna dengan sebuah sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, membuat sebuah daftar aktifitas, dan sebagainya. Pada Gambar II.2 diperlihatkan contoh diagram *use case*



Gambar II.2. Use Case Diagram
(Sumber : Angga Dharma ,2011)

II.5.1.2. Activity Diagram

Activity Diagram adalah teknik untuk menggambarkan logika *procedural*, proses bisnis, dan alur kerja. Dalam beberapa masalah, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dengan notasi diagram alir adalah diagram ini mendukung *behavioral parallel*. *Activity Diagram* memungkinkan siapa pun melakukan proses untuk memilih urutan dalam melakukannya, atau menyebutkan aturan-aturan yang harus diikuti.



Gambar II.3. Activity Diagram
(Sumber : Angga Dharma ,2011)