BAB 2

LANDASAN TEORI

2.1. Sejarah Kerajaan Singosari

Kerajaan Singosari didirikan Ken Arok dari tahun 1222-1227. Ken Arok dibunuh Anusapati atas dasar balas dendam karena ayah Anusapati dibunuh oleh Ken Arok. Berita pembunuhan Ken Arok sampai ke telinga Tohjaya, anak Ken Arok.

Pada tahun 1248, Anusapati dibunuh oleh Tohjaya ketika keduanya sedang sabung ayam. Setelah membunuh Anusapati, Tohjaya menjadi raja Singosari. Namun kemudian Tohjaya dikalahkan anak Anusapati yaitu Ranggawuni.

Sejak Ranggawuni naik tahta kerajaan Singosari aman tentram. Kemudian Ranggawuni diganti oleh Kertanegara, ia raja Singosari paling terkenal. Ia bercita-cita untuk memperluas wilayah ke seluruh nusantara.

Pada tahun 1275, ia mengirimkan sebuah ekpedisi ke kerajaan Melayu, yang dikenal dengan Ekspedisi Pamalayu. Tujuannya untuk menghancurkan kerajaan Sriwijaya lewat jalinan persahabatan dengan kerajaan Melayu. Ekspedisi Singosari bertujuan pula untuk menyaingi Kerajaan Mongol yang dipimpin Kubilal Khan. Raja Kubilal Khan merasa tersaingi, lalu kerajaan Mongol mengirim utusan bernama Meng-Chi, agar Kertanegara mengakui kekuasaan Kubilal Khan. Utusan tadi dikirim kembali ke Mongol dengan muka yang cacat, karena Kartanegara tidak mau tunduk kepada raja Mongol.

Tentu saja hal tersebut membuat marah Kubilal Khan. Ia ingin membalas dendam untuk menyerang Singosari. Pada kenyataannya serangan Mongol salah sasaran, sebab raja Kertanegara telah gugur oleh Jayakatwang keturunan Kediri[3].

2.2. Ken Arok

Ken Arok adalah putra Dewa Brahma dengan seorang wanita desa Pangkur bernama Ken Ndok. Oleh ibunya, bayi Ken Arok dibuang di sebuah pemakaman, hingga kemudian ditemukan dan diasuh oleh seorang pencuri bernama Lembong.

Ken Arok tumbuh menjadi berandalan yang lihai mencuri & gemar berjudi, sehingga membebani Lembong dengan banyak hutang. Lembong pun mengusirnya. Ia kemudian diasuh oleh Bango Samparan, seorang penjudi pula yang menganggapnya sebagai pembawa keberuntungan.

Ken Arok tidak betah hidup menjadi anak angkat Genukbuntu, istri tua Bango Samparan. Ia kemudian bersahabat dengan Tita, anak kepala desa Siganggeng. Keduanya pun menjadi pasangan perampok yang ditakuti di seluruh kawasan Kerajaan Kadiri.

Akhirnya, Ken Arok bertemu seorang brahmana dari India bernama Lohgawe, yang datang ke tanah Jawa mencari titisan Wisnu. Dari ciri-ciri yang ditemukan, Lohgawe yakin kalau Ken Arok adalah orang yang dicarinya.

Tumapel merupakan salah satu daerah bawahan Kerajaan Kadiri. Yang menjadi *akuwu* (setara camat zaman sekarang) Tumapel saat itu bernama Tunggul Ametung. Atas bantuan Lohgawe, Ken Arok dapat diterima bekerja sebagai pengawal Tunggul Ametung.

Ken Arok kemudian tertarik pada Ken Dedes istri Tunggul Ametung yang cantik. Apalagi Lohgawe juga meramalkan kalau Ken Dedes akan menurunkan raja-raja tanah Jawa. Hal itu semakin membuat Ken Arok berhasrat untuk merebut Ken Dedes, meskipun tidak direstui Lohgawe.

Ken Arok membutuhkan sebilah keris ampuh untuk membunuh Tunggul Ametung yang terkenal sakti. Bango Samparan pun memperkenalkan Ken Arok pada sahabatnya yang bernama Mpu Gandringdari desa Lulumbang (sekarang Lumbang, Pasuruan), yaitu seorang ahli pembuat pusaka ampuh.

Mpu Gandring sanggup membuatkan sebilah keris ampuh dalam waktu setahun. Ken Arok tidak sabar. Lima bulan kemudian ia datang mengambil pesanan. Keris yang belum sempurna itu direbut dan ditusukkan ke dada Mpu Gandring sampai tewas. Dalam sekaratnya, Mpu Gandring mengucapkan kutukan bahwa keris itu nantinya akan membunuh 7 orang, termasuk Ken Arok sendiri.

Kembali ke Tumapel, Ken Arok menjalankan rencananya untuk merebut kekuasaan Tunggul Ametung. Mula-mula ia meminjamkan keris pusakanya pada Kebo Hijo, rekan sesama pengawal. Kebo Hijo dengan bangga memamerkan keris itu sebagai miliknya kepada semua orang yang ia temui, sehingga semua orang

mengira bahwa keris itu adalah milik Kebo Hijo. Dengan demikian, siasat Ken Arok berhasil.

Malam berikutnya, Ken Arok mencuri keris pusaka itu dari tangan Kebo Hijo yang sedang mabuk arak. Ia lalu menyusup ke kamar tidur Tunggul Ametung dan membunuh majikannya itu di atas ranjang. Ken Dedes menjadi saksi pembunuhan suaminya. Namun hatinya luluh oleh rayuan Ken Arok. Lagi pula, Ken Dedes menikah dengan Tunggul Ametung dilandasi rasa keterpaksaan.

Pagi harinya, Kebo Hijo dihukum mati karena kerisnya ditemukan menancap pada mayat Tunggul Ametung. Ken Arok lalu mengangkat dirinya sendiri sebagai akuwu baru di Tumapel dan menikahiKen Dedes. Tidak seorang pun yang berani menentang kepustusan itu. Ken Dedes sendiri saat itu sedang mengandung anak Tunggul Ametung.

Pada tahun 1222 terjadi perselisihan antara Kertajaya raja Kadiri dengan brahmana. Para brahmana itu memilih pindah ke Tumapel meminta perlindungan Ken Arok yang kebetulan sedang mempersiapkan pemberontakan terhadap Kadiri. Setelah mendapat dukungan mereka, Ken Arok pun menyatakan Tumapel sebagai kerajaan merdeka yang lepas dari Kadiri. Sebagai raja pertama ia bergelar Sri Rajasa Bhatara Sang Amurwabhumi.

Kertajaya (dalam *Pararaton* disebut Dhandhang Gendis) tidak takut menghadapi pemberontakan Tumapel. Ia mengaku hanya dapat dikalahkan oleh Bhatara Siwa. Mendengar sesumbar itu, Ken Arok pun memakai gelar Bhatara Siwa dan siap memerangi Kertajaya. Perang antara Kadiri dan Tumapel

terjadi dekat Ganter. Pihak Kadiri kalah, Kertajaya diberitakan naik ke alam dewa, yang mungkin merupakan bahasa kiasan untuk mati.

Anusapati merasa heran pada sikap Ken Arok yang seolah menganaktirikan dirinya, padahal ia merasa sebagai putra tertua. Setelah mendesak ibunya (Ken Dedes), akhirnya Anusapati mengetahui kalau dirinya memang benar-benar anak tiri. Bahkan, ia juga mengetahui kalau ayah kandungnya bernama Tunggul Ametung telah mati dibunuh Ken Arok.

Anusapati berhasil mendapatkan Keris Mpu Gandringyang selama ini disimpan Ken Dedes. Ia kemudian menyuruh pembantunya yang berasal dari desa Batil untuk membunuh Ken Arok. Ken Arok tewas ditusuk dari belakang saat sedang makan sore hari. Anusapati ganti membunuh pembantunya itu untuk menghilangkan jejak. Peristiwa kematian Ken Arok dalam naskah *Pararaton* terjadi pada tahun 1247 [4].

2.3. Game Edukasi

Penerapan game edukasi bermula dari perkembangan *video game* yang sangat pesat dan menjadikannya sebagai media efektif yang interaktif dan banyak dikembangkan di perindustrian. Melihat kepopuleran game tersebut, para pendidik berpikir bahwa mereka mempunyai kesempatan yang baik untuk menggunakan komponen rancangan game dan menerapkannya pada kurikulum dengan penggunaan industri berbasis game. *Game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan [1].

2.3.1. Pengertian *Game*

Sebuah *game* adalah aktivitas terstuktur atau semi terstuktur, yang biasanya dilakukan untuk bersenang senang dan kadang juga di gunakan sebagai alat pembelajaran. Sebuah *game* bisa di karakteristik dari “apa yang pemain lakukan.” Hal tersebut adalah sebagai berikut:

a. Peralatan , seperti bola, kartu, atau sebuah komputer
b. Peraturan , biasa menentukan giliran permainan, hak dan kewajiban pemain dan tujuan permainan
c. Skill strategi dan keburuntungan. *Game* dengan skill misalnya kekuatan fisik dan kekuatan mental
d. *Single Player Game* (permainan satu orang). Permainan tunggal ini adalah permaian dengan keahlian, berpacu dengan waktu atau keberuntungan.

Pengertian *game* komputer menurut beberapa ahli :

1. Menurut Chris Crawford, seorang *computer game designer* mengemukakan bahwa *game*, pada intinya adalah sebuah interaktif, aktivitas yang berpusat pada sebuah pencapaian, ada pelaku aktif, ada lawan anda.
2. Menurut David Parlett, *Game* adalah sesuatu yang memiliki "akhir dan cara mencapainya": artinya ada tujuan, hasil dan serangkaian peraturan untuk mencapai keduanya.

3. Menurut Roger Caillois, seorang sosiolog Perancis, dalam bukunya yang berjudul *Les jeux et les hommes* menyatakan *game* adalah aktivitas yang mencakup karakteristik berikut: *fun* (bebas bermain adalah pilihan bukan kewajiban), *separate* (terpisah), *uncertain*, *non-productive*, *governed by rules* (ada aturan), *fictitious* (pura-pura).

2.3.2. Pengertian Edukasi

Edukasi adalah proses yang dilakukan oleh seseoarang untuk menemukan jati dirinya, yang dilakukan dengan mengamati dan belajar yang kemudian melahirkan tindakan dan prilaku. Edukasi sebenarnya tidak jauh berbeda dari belajar yang dikembangkan oleh aliran behaviorisme dalam psikologi. Hanya istilah ini sering dimaknai dan diinterpretasikan berbeda dari learning yang bermakna belajar. Dan istilah ini seringkali digunakan dalam pendekatan pendidikan yang tentu maknanya lebih dari sekedar belajar

Dalam teori belajar dan prilaku sebuah tindakan akan didapatkan dari sebuah aktifitas pengamatan yang dikondisikan dan direkayasa dengan tujuan hasil yang sudah ditentukan dan bisa diukur secara logis dan ilmiah. Dalam belajar ada hubungan erat antara penguat, baik yang positif atau negatif yang terjadi setelah tindakan dilakukan, atau setelah menjadi sebuah prilaku kongkrit yang bisa dilihat dengan kasat mata. Stimulan yang terjadi sebelum prilaku muncul sangat mempengaruhi tujuan yang diharapakan. Ini artinya prilaku seseoarang harus selalu bisa diukur dan selalu pulu diberikan penguat dan

stimulasi yang diciptakan sedimikian rupa sesuai dengan tujuan yang hendak dicapai.

Edukasi dalam pendekatan pendidikan lebih kepada pemberian kondisi dan pengaruh kepada seseorang agar bisa menciptakan prilaku sepontan dalam jangka waktu yang lama sesuai dengan harapan diri sendiri dan masyarakat. Harapan itu diukur dengan kesepakatan-kesepakatan formal yang disebut hukum, dan kesepakatan-kesepakatan yang tidak formal tetapi menjadi kebiasaan yang dianggap baik berlaku di masyarakat yang disebut adat. Inilah prilaku yang diharapkan dari edukasi yang sepadan dengan kata learning dalam pandangan psikologi *behaviorisme.*

2.3.3. Pengertian Game Edukasi

Game edukasi adalah *game* yang khusus dirancang untuk mengajarkan user suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya[1].

2.3.4. Kriteria Edukasi Game

Perancangan game edukasi yang baik haruslah memenuhi kriteria dari game edukasi itu sendiri[4]. Berikut ini adalah beberapa kriteria dari sebuah game edukasi, 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*. Apliksi 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 aplikas 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. Definisi Artificial Intelegence (AI)

Para ahli mendefinisikan AI secara berbeda-beda tergantung pada sudut pandang mereka masing-masing. Ada yang fokus pada logika berpikir manusia saja, tetapi ada juga yang mendefinisikan AI secara lebih luas pada tingkah laku manusia. Stuart Russel dan Peter Norvig mengelompokkan definisi AI, yang diperoleh dari beberapa *textbook,* ke dalam empat kategori [5], yaitu:

1. *Thinking humanly: the cognitive modeling approach*

   Pendekatan ini dilakukan dengan dua cara sebagai berikut:

   1) Melalui instrospeksi: mencoba menangkap pemikiran-pemikiran kita sendiri saat kita berpikir.
   2) Melalui eksperimen-eksperimen psikologi.

2. *Acting humanly: the Turing test approach*

Pada tahun 1950, Alan Turing merancang suatu ujian bagi computer beritelejensia untuk menguji apakah computer tersebut mampu mengelabuhi seorang manusia yang menginterogasinya melalui *teletype* (komunikasi berbasis teks jarak jauh). Jika *interrogator* tidak dapat membedakan yang diinteroragasi adalah manusia atau computer, maka computer berintelenjisia tersebut lolos dari *Turing test.* Komputer tersebut perlu memiliki kemampuan *Natural Language Processing, Knowledge Representation, Automated Reasoning, Machine Learning, Computer Vision, Robotics. Turing test* sengaja menghindari interaksi fisik antara *interrogator* dan komputer karena simulasi fisik manusia tidak memerlukan intelenjisia.

3. *Thinking rationally: the laws of thought approach*

   Terdapat dua masalah dalam pendekatan ini yaitu:

   1) Tidak mudah untuk membuat pengetahuan informasi dan menyatakan pengetahuan tersebut ke dalam *formal term* yang diperlukan oleh notasi logika, khususnya ketika pengetahuan tersebut memiliki kepastian kurang dari 100%.
   2) Terdapat perbedaan besar antara dapat memecahkan masalah "dalam prinsip" dan memecahkannya "dalam dunia maya".

4. *Acting rationally: the rational agent approach*

   Membuat inferensi yang logis merupakan bagian dari suatu *rational agent.* Hal ini disebabkan satu-satunya cara untuk melakukan aksi secara rasional adalah dengan menalar secara logis. Dengan menalar

secara logis, maka bisa didapatkan kesimpulan bahwa aksi yang diberikan akan mencapai tujuan atau tidak. Jika mencapai tujuan, maka *agent* dapat melakukan aksi berdasarkan kesimpulan tersebut.

Thinking humanly dan *acting humanly* adalah dua definisi dalam arti yang sangat luas. Sampai saat ini, pemikiran manusia yang di luar rasio, yakni reflex dan intuitif (berhubungan dengan perasaan), belum dapat ditirukan sepenuhnya oleh computer. Dengan demikian, kedua definisi ini dirasa kurang tepat untuk saat ini. Jika kita menggunakan definisi ini, maka banyak produk komputasu cerdas saat ini yang tidak layak disebut sebagai produk AI.

Definisi *thinking rationally* teras lebihi sempit daripada *acting rationally* dengn pendekatan *rational agent.* Hal ini berdasarkan pemikiran bahwa computer bisa melakukan penalaran secara logis dan juga bisa melakukan aksi secara rasional berdasarkan hasil penalaran tersebut.

2.4.1. Graf Berarah pada *Game*

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Graf direpresentasikan dengan simpul-simpul (*vertex*) yang dihubungkan oleh garis-garis yang disebut sisi (*edge*). Simpul-simpul merepresentasikan objek-objek diskrit dan sisi-sisi merepresentasikan hubungan antara objek-objek tersebut. [6]

$G = \{V, E\}$, dengan:

G : graf

V : himpunan tidak kosong dari simpul-simpul (*vertex*)

E : himpunan sisi-sisi (*edges*) yang menghubungkan sepasang simpul

Graf bisa dikelompokkan menjadi beberapa macam. Berdasarkan ada tidaknya sisi ganda ataupun kalang, graf dikelompokkan menjadi graf sederhana (*simple graph*) dan graf tak-sederhana (*unsimple graph*). Lalu berdasarkan arah pada sisi, graf dikelompokkan menjadi graf tak-berarah (*undirected graph*) dan graf berarah (*directed graph*). Graf sederhana adalah graf yang tidak mengandung kalang maupun sisi ganda, sedangkan graf taksederhana adalah graf yang mengandung sisi ganda atau kalang. Graf tak-berarah adalah adalah graf yang sisinya tidak mempunyai orientasi arah, sedangkan graf berarah adalah graf yang pada setiap sisinya diberikan orientasi arah.

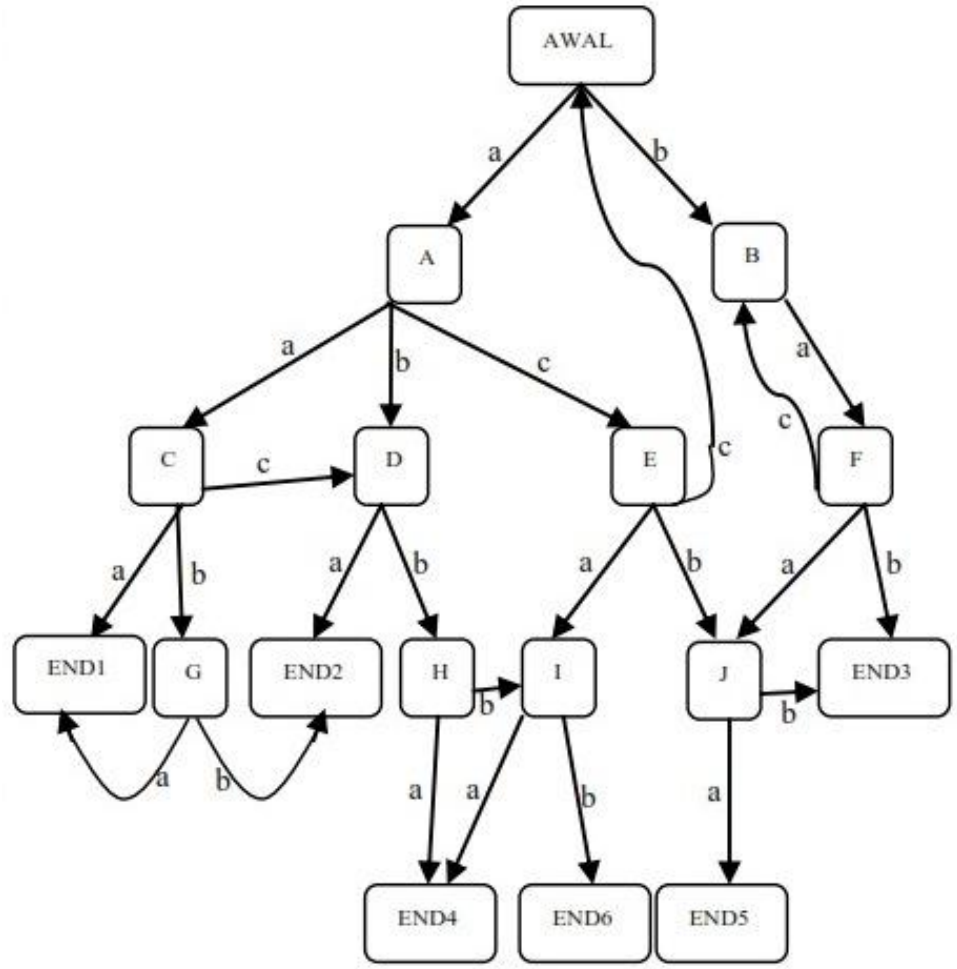
Dari berbagai jenis graf, jenis graf berarahlah yang akan banyak dipakai dalam perancangan nantinya. *Game* beralur kompleks adalah *game* yang pada suatu alur cerita ada pilihan yang membuat alur cerita kembali seperti alur cerita yang sudah pernah dijalani ataupun ada pilihan yang memungkinkan adanya loop yang membuat alur cerita hanya berputar-putar saja. Perancangan alur cerita dari *game* yang beralur kompleks tidak bisa diselesaikan dengan pohon berakar, oleh karena itu digunakan graf berarah yang lebih umum daripada pohon berakar.

Contoh dari algoritma *game* yang beralur:

```
//------------------------
...
IF <KONDISI E> THEN
PILIHAN:
[a] : <KONDISI I>;
[b] : <KONDISI J>;
[c] : <KONDISI AWAL>;

IF <KONDISI F> THEN
PILIHAN:
[a] : <KONDISI J>;
[b] : <KONDISI END3>; END;
[c] : <KONDISI B>;
...
//------------------------
```

Perubahan algoritma di atas menyebabkan adanya pilihan yang memungkinkan alur cerita kembali ke kondisi awal dan juga ke kondisi sebelumnya yang akan membuat sirkuit. Pembentukan sirkuit inilah yang menyebabkan pohon berakar tidak bisa diaplikasikan. Dengan mengaplikasikan graf berarah pada algoritma penyusunan alur cerita tersebut bisa dibentuk graf berarah yang merepresentasikan alur cerita dari *game* tersebut.



Gambar 2.1 Representasi alur cerita yang beralur kompleks dengan graf berarah

Dengan menggunakan graf berarah alur cerita yang mempunyai pilihan untuk kembali ke kondisi sebelumnya bisa direpresentasikan seperti gambar di atas. Dalam pembangunan graf berarah akan lebih mudah dengan membuat semua kondisi yang ada. Lalu satu persatu kondisi, lebih baik jika dari kondisi awal, dibuat sisi-sisi yang menghubungkan kondisi dirinya dengan kondisi berikutnya sesuai algoritma yang diberikan. Jika semua kondisi sudah terhubung sesuai algoritmanya, maka graf berarah yang merepresenatasikan alur cerita sudah terbangun dengan utuh.

2.5. Collision Detection

Collision detection adalah proses pendeteksian tabrakan antara dua objek. Sebenarnya dalam *game* sendiri tabrakan tidak hanya terjadi antara dua objek, tetapi dapat terjadi juga antara satu objek dengan banyak atau banyak objek dengan banyak objek. Dalam *game* membutuhkan *collision detection* yang akurat karena setelah menentukan terjadinya *collision* atau tidak kita harus menentukan apa yang terjadi pada objek yang ditabrak ataupun yang menabrak [7].

Collision detection ini juga berguna untuk menentukan posisi dari satu objek dengan objek yang lain sehingga tidak ada objek yang saling menembus. Sehingga *game* yang akan dibuat memiliki kesamaan dengan realita yang ada.

Teknik *Collision Detection* pada 2D ada 2 cara yaitu :

1. *Rectangles collision detection*
2. *Circle rectangle collision detection*

Untuk game 2D setiap objek yang di perhatikan adalah vector 2D-nya, yaitu panjang dan lebar. Namun tidak setiap objek berbentuk persegi, lingkaran, segilima atau bentuk lain yang beraturan. Seperti bentuk pohon atau kuda dan lainnya, sehingga *collision detection* untuk objek-objek seperti ini memerlukan pandekatan lain yang bisa mempermudah terutama dalah hal pemrograman. Karena untuk bentuk-bentuk yang tidak beraturan sangat sulit untuk dideteksi.

1. Rectangles collision detection

Teknik ini merupakan pendeteksian tabrakan menggunakan *bound* berbentuk persegi, dimana setiap sprite atau objek yang dibuat dalam *game* memiliki *bound* masing-masing. Cara ini cukup mudah karena

hanya perlu membandingkan apakah ada *intersection* (irisan) pada kedua objek pada kordinat tertentu. Namun cara ini kurang akurat karena untuk *sprite* atau objek yang memiliki ruang kosong yang besar dengan *bound*-nya. Hal ini akan muncul saat kondisi kedua batas objek tersebut beririsan walaupun sebenarnya tidak terjadi tabrakan maka akan dianggap terjadi tabrakan. Cara ini cocok untuk *game* yang tidak memperhatikan akurasi deteksi tabrakan atau untuk *game* yang menggunakan *sprite* yang bentuknya mendekati persegi.

Adapun cara lain dari hasil modifikasi teknik ini yaitu *reduced size bounding box*. Dimana kotak batasnya diperkecil dengan mengurangi beberapa pixel dari ukuran *sprite* atau objek yang sebenarnya. Sehingga dengan cara ini pada saat melakukan *collision detection* akan lebih akurat karena ruang kosong antara *sprite* dan *bound* berkurang.

Kemudian cara lain dari modifikasi teknik ini adalah dengan cara menggunakan beberapa *bounding box* secara bersamaan untuk satu *sprite* atau objek. Contohnya untuk *sprite* berbentuk manusia, pertama *bounding box* digunakan untuk bagian kepala kemudian *bounding box* yang lainnya digunakan untuk badan dan kaki. Cara ini lebih akurat karena banyaknya *bounding box* yang digunakan dapat disesuaikan dengan bentuk gambar pada *sprite*. Namun semakin banyak *bounding box* yang digunakan akan semakin banyak pula waktu yang digunakan untuk *collision detection*.

2. Circle rectangle collision detection

Untuk teknik ini memiliki cara yang hampir sama dengan yang sebelumnya hanya saja disini *bound* atau batas yang digunakan berbentuk lingkaran. Cara pengecekannya lebih mudah karena lingkaran memiliki jarak yang sama ke batas luarnya. Caranya yaitu dengan menghitung jarak kedua objek dan membandingkannya dengan jumlah radius masing-masing *bound* dari kedua objek, jika jaraknya lebih kecil dari jumlah radius kedua objek maka terjadi tabrakan (*collision*) dan jika lebih besar maka tidak terjadi tabrakan. Cara sebelumnya yaitu *rectangle collision detection* lebih akurat daripada cara ini.

2.6. Pengertian *Unified Modeling Language* (UML)

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 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, memvisualisasikan, 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 sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem[8].

2.5.1. Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*.
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur *object*.
9. *Component Diagram* untuk memodelkan komponen *object*.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object, yaitu *use case diagram, sequence diagram, collaboration diagram, dan class diagram*.

1. Use Case Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

2. Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

3. Collaboration Diagram

Collaboration diagram dipakai untuk memodelkan interaksi antar object di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration diagram* lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh object.

4. Class Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

2.7. Perangkat Lunak Pendukung

Perangkat lunak (*Software*) sangatlah dibutuhkan dalam sitem aplikasi ini, karena sistem aplikasi yang akan dibangun ini membutuhkan program aplikasi yang sesuai dengan yang dibutuhkan oleh pengguna. Adapun program aplikasi yang digunakan dalam membangun perangkat lunak ini adalah RPG Maker XP.

2.6.1. RPG Maker XP

Ruby *Game Scripting System* (atau RGSS) adalah turunan dari bahasa pemrograman Ruby yang dikhususkan penggunaannya untuk membuat game-game PC. *Game* hasil buatan RPG Maker XP ini memiliki resolusi standar 640*480 yang dapat diubah melalui pengubahan *script* RGSS.

RPG Maker XP menggunakan gambar *untuk title screen, game over screen, character set, tileset,* efek transisi, animasi pertarungan, icon pertarungan, icon item, latar belakang, dan window skin. Gambar-gambar baru dapat dibuat melalui software pengolah gambar. RGSS dapat menerima semua ekstensi gambar secara teknis seperti PNG, BMP, dan JPG. Tetapi, dalam penggunaan animasi GIF, sebuah *script* harus diinputkan kedalam

database agar animasi dapat ditampilkan, jika tidak, maka yang tampil hanyalah sebuah gambar biasa.

Sistem pertarungan yang disediakan secara *default* adalah *first-person* dan bergiliran. Programmer dapat membuat animasi serangan baru dan mengimpor gambar untuk digunakan dalam latar belakang pertarungan. Sistem pertarungan dapat diedit oleh programmer pada databasenya menggunakan script RGSS dan Ruby itu sendiri.

Karakter ditampilkan dalam set-set (dinamakan "Character Set" atau "Charset"). Setiap animasi ditampilkan dalam segi 4x4 "pose" untuk setiap *Charset*. Desain animasi terebut memberikan sebuah ilusi yang mengesankan bahwa karakter tersebut benar-benar berjalan [9].

2.6.2. Star UML

StarUML merupakan *open source* alat UML , berlisensi versi modifikasi dari GNU GPL . Setelah ditinggalkan untuk beberapa waktu, proyek ini memiliki kebangkitan terakhir yang pindah dari Delphi untuk Java / Eclipse dan berhenti lagi. Namun, masyarakat masih aktif dan banyak topik yang dibahas di forum.

Tujuan dari proyek ini adalah untuk menggantikan yang lebih besar, aplikasi komersial seperti Rational Rose dan Borland 's Together . StarUML mendukung sebagian besar jenis diagram ditentukan dalam UML 2.0 . Saat ini tidak ada objek , paket , waktu dan diagram gambaran interaksi (meskipun dua yang pertama bisa cukup dimodelkan melalui diagram kelas editor).

StarUML ditulis dalam Delphi , yang merupakan salah satu alasan mengapa hal itu ditinggalkan untuk waktu yang lama. Dalam StarUML Ada ekspor ke gambar raster format, JPEG , dan vektor format gambar Windows Metafile . Tapi, tidak ada ekspor ke SVG dan PNG. Untuk impor, bisa impor dari XMI , dan Rational Rose format file [10].

2.6.3. Visual Paradigm for UML 10.0 Community Edition

VP-UML Community Edition (VP-UML CE) UML adalah alat bagi Anda untuk membuat diagram UML dengan mudah dan cepat melalui perusahaan pemenang penghargaan antarmuka diagram secara gratis . VP-UML CE tidak hanya mendukung UML tetapi juga ERD dan SysML . VP-UML CE ini sesuai untuk siswa dan pemula yang ingin belajar dan berlatih UML dan keterampilan pemodelan dengan UML. Kelompok open source dan masyarakat menggunakan VP-UML CE untuk mendokumentasikan proyek perangkat lunak mereka dan mempublikasikan output ke halaman Web/ blog mereka [11].