# BAB 2

# TINJAUAN PUSTAKA

## *2.1 Pengertian Game*

*Game* berasal dari kata bahasa Inggris yang mempunyai arti permainan. Permainan dalam hal ini merujuk pada pengertian “kelincahan intelektual” (*intellectual playability*) [3]. *Game* juga dapat 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.

Pada awalnya, *game* identik dengan permainan anak-anak. Orang selalu berpikir *game* merupakan suatu kegiatan yang dilakukan oleh anak-anak yang dapat menyenangkan hati mereka.

Dengan kata lain, segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai *game*.

Berikut pengertian *game* menurut beberapa para ahli [4] :

1. Menurut Chris Crawford, seorang desainer *game* komputer mengemukakan bahwa *game*, pada intinya adalah sebuah interaktif, aktivitas yang berpusat pada sebuah pencapaian, ada pelaku aktif, dan ada lawan.
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).
4. Menurut Clark C. Abt, *game* adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan “dibatasi oleh konteks tertentu” (misalnya, dibatasi oleh peraturan).
5. Menurut Bernard Suits *game* adalah “upaya sukarela untuk mengatasi rintangan yang tidak perlu”. “Rintangan yang tidak perlu”.
6. Menurut Greg Costikyan, *Game* adalah “sebentuk karya seni di mana peserta, yang disebut Pemain, membuat keputusan untuk mengelola sumberdaya yang dimilikinya melalui benda di dalam *Game* demi mencapai tujuan”.
7. Definisi ini dari buku *Rules of Play* karya Katie Salen dan Eric Zimmerman, *game* adalah “sistem tempat pemain melakukan konflik bohongan, ditentukan oleh aturan, yang memberi hasil terukur”.

Permainan video atau dalam istilah asing sering disebut *'video game'* merupakan permainan eleklronik yang melibatkan interaksi antara manusia . dengan *inteface* tertentu yang kemudian hasilnya ditampilkan secara langsung pada perangkat video [7]. Kata video dalam istilah permainan video sebenarnya merujuk secara tradisional pada perangkat *display* berbasis *raster,* namun kini

istilah itu pun berlaku pada perangkat *display* apa saja, bahkan yang tidak berbasis *raste*r sekalipun.

Sistem perangkat elektronis yang digunakan untuk menjalankan permainan video biasa disebut dengan *platform*. Secara garis besar, *platform* terbagi menjadi dua, yaitu komputer personal dan konsol permainan video. *Platform* tersebut sangatlah bervariasi, mulai dari yang berukuran sangat besar seperti komputer *mainframe* sampai dengan yang berukuran kecil seperti *handheld device* 'perangkat genggam'.

Alat *input* 'masukan' yang digunakan untuk memanipulasi permainan video disebut *game controller* dan jenisnya pun bermacam-macam dari setiap *platform*. Jenis kontroler tersebut mulai dari *joystick* sederhana, *keyboard* atau bahkan sampai dengan sensor gerak. Pada masa-masa awal, alat *input* untuk permainan video yang berjalan pada komputer personal hanyalah *keyboard* saja, tapi kini *mouse* pun bisa digunakan. Lebih lagi, kini berbagai jenis *game controller* pun banyak bermunculan seperti *joystick* nirkabel, dan perangkat yang dapat menerima sensor gerak.

### 2.1.1 Jenis-jenis *Game*

Jenis *game* dapat dibagi atas beberapa kategori atau 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 *genre* bisa digabungkan kedalam sebuah *game* untuk membuat unsur permainan lebih bervariasi dan menantang, berikut ini adalah *genre game*[8]

a. *Maze Game*

Secara sederhana pemain hanya mengitari *maze* (lorong-lorong yang berhubungan) dan memakan beberapa item untuk menambah tenaga atau kekebalan. Pemain juga tentunya memiliki musuh yang mengejar. Ketika pendapatkan kekebalan pemain bisa berbalik mengejar mereka. Sederhana, tetapi mengasikan. Contoh : *Digger, Pacman, Doom, Quake.*

b. *Board Game*

Jenis *game* ini sama dengan *game* board tradisional, *Game* ini memindahkan versi tradisional ke layar komputer. *Game* ini melibatkan kemampuan AI (*Artificial Intelligence*) yang andal untuk bisa menjadikan *Game* ini menantang pemain dengan baik. Contoh : *Chess, Monopoly, Scrabble.*

c. *Card Game*

Yaitu *Game* dengan menggunakan kartu sebagai permainan, kartu yang digunakan adalah seperti kartu tradisional.Contoh : *Hearts, Spider, Blackjack, Poker, Solitaire.*

d. *Trading Card Game*

Yaitu *game* dengan set kartu yang dirancang khusus untuk permainan ini. Contoh yang populer adalah *Battle Card Pokemon*. Versi *Game* elektroniknya sangat digemari di luar negeri. Contoh lain : *Yu Gi Oh, Duel Master, Pokemon, Magic The Gathering.*

*e. Shooting Game*

Secara standar bahwa semua *game* yang bertipe atau dimainkan dengan cara menembak objek adalah termasuk *Shooting Game*. Contoh : *Deep Hunter, Hunting Unlimited 3.*

*f. Quiz Game*

Permainan hanya perlu memilih jawaban yang benar dari beberapa pilihan jawaban. Ada yang memilih topik tertentu dan ada juga yang topiknya beragam. Contoh : *Who Wants To Be Millionare, Deal or No Deal.*

*g. Puzzle Game*

Yaitu *game* dengan menyusun item sedemikian rupa dan penyususnan dilakukan secepat dan sebaik mungkin. Contoh : *Tertis, Magic Inlay, Adventure Inlay, Rocket Mania, Chip Challenge.*

*h. Action*

*Game genre* ini lebih mengutamakan ketangkasan koordinasi tangan dan mata, dimana kecepatan dan refleks menjadi andalan utama sang pemain.

*i. Arcade*

Jenis ini dapat dikatakan seperti *action* yang hanya memiliki pola main yang lebih sederhana dari pada jenis action

*j. Adventure*

Jenis ini bersifat petualangan dimana pemain mengendalikan sebuah tokoh yang mengikuti suatu alur cerita dengan menghadapi berbagai tantangan dan teka-teki.

k. *Role Playing Game* (RPG)

Di *genre* ini permainan akan berperan menjadi sebuah karakter. Pemain akan menjalankan peran dengan berbagai atribut, seperti kesehatan, inteligensi, kekuatan dan keahlian. *Genre game* ini berkembang menjadi beberapa jenis variasi RPG seperti *action* RPG. Contoh : *Ragnarok, Final Fantasi Series, Kingdom Hearts, Beyond Divinity, Dragon Quest Series.*

l. *Real Time Strategy* (RTS) *Game*

Pada *game* ini pemain harus melakukan berbagai gerakan dan strategi. Pada *genre* ini pemain tidak harus saling menunggu, malah pemain tercepatlah yang besar kemungkinannya untuk menang. Contoh : *Warcraft, Starcraft, Stronghold Crusader, Command and Conquer*.

m. *First Person Shooter* (FPS) *Game*

*Game* ini mengutamakan kecepatan gerakan di dalam permainan. Banyak baku tembak dan pemain harus bertahan selama mungkin. Disebut *First Person Shooter* karena panndangan pemain adalah pandangan orang pertama *(First Person).* Pemain melihat tampilan dilayar seperti pemain melihat dari mata pemain sendiri. Contoh : *Call of Duty, Medal of Honor, Counter Strike, Doom, Quake 4, Alien Vs Predator.*

*n. Third Person Action Games*

Sebenarnya *genre* ini sama dengan FPS, hanya sudut pandang yang berbeda. Pada *genre* in pemaian melihat dari sudut pandang orang ketiga. Contoh : *Assasin's Creed, Prince of Persia, Tomb Raider, Metal Gear Solid.*

o. *First Person 3D Vehicle Bused*

*Game* ini sama FPS hanya bedanya pandangan pemain bukan dari mata tetapi dari sudut pandang kendaraan atau mesin yang dinaiki pemain. Kendaraan bisa berupa kapal, tank atau robot raksasa.

p. *Full Motion Video Game* (FMV)

*Game* ini meminta pemain memecahkan misteri. Caranya gampang, hanya mengklik beberapa objek dari layar dan animasi atau film akan muncul. Lalu layar baru akan muncul sejalan dengan pilihan pemain. Contoh : *Riven, Myst.*

*q.* *Real Word Situation*

*Game* ini meliputi permainan olahraga dan simulasi masalah kendaraan termasuk kendaraan militer. *Game* ini kebanyakan melibatkan masalah fisik dan taktik, tetapi tidak masalah eksplorasi, ekonomi dan konseptual. Contohnya seperti *Game Footbal Manager.*

*r.* Construction and Management

Seperti *Game Roller Coster Tycoon* dan *The Sims*. Pada dasarnya adalah *game* yang menitik beratkan pada masalah ekonomi dan konseptual. *Game* ini jarang yang melibatkan konflik dan eksplorasi, dan hampir tidak pernah meliputi tantangan fisik.

### 2.1.2 Klasifikasi *Game*

Berikut ini adalah klasifikasi permainan berdasarkan pembatasan usia dari pemainnya, menurut ESRB (*Entertainment Software Rating Board*) adalah sebagai berikut :

1. *Early Childhood (3+)* memiliki isi yang mungkin cocok untuk usia 3 keatas. Tidak mengandung materi yang tidak pantas seperti adegan kekerasan.
2. *Everyone (6+)* memiliki konten yang mungkin cocok untuk usia 6 keatas. Judul-judul dalam kategori ini mungkin berisi sedikit kartun, fantasi atau kekerasan ringan dan tidak jarang menggunakan bahasa yang ringan.
3. *Everyone (10+)* memiliki konten yang mungkin cocok untuk usia 10 keatas. Judul-judul dalam kategori ini mungkin berisi lebih banyak kartun, fantasi atau kekerasan ringan, bahasa ringan dan juga minimal tema sugestif.
4. *Teen (13+)* memiliki konten yang mungkin cocok untuk usia 13 keatas. Judul-judul dalam kategori ini mungkin mengandung kekerasan, tema sugestif, humor kasar, sedikit darah, simulasi perjudian, dan tidak jarang menggunakan bahasa yang kuat.
5. *Mature (17+)* memiliki konten yang mungkin cocok untuk orang usia 17 dan lebih tua. Judul-judul dalam kategori ini mungkin berisi intens kekerasan, darah, konten seksual dan bahasa yang kuat.
6. *Adults Only (18+)* memiliki konten yang hanya boleh dimainkan oleh orang-orang 18 tahun keatas. Judul-judul dalam kategori ini mungkin termasuk adegan yang intens, kekerasan dan juga mengandung konten seksual.

7. *RP (Rating Pending*) telah disampaikan kepada ESRB dan sedang menunggu penilaian akhir. (Simbol ini hanya muncul dalam iklan sebelum permainan dibebaskan) [6].

## *2.2 Artificial Intelligence*

Sebagian kalangan menerjemahkan *Artificial Intelligence* sebagai kecerdasan buatan, kecerdasan artifisial, intelijensia artifisial, atau intelijensia buatan. Dalam karya tulis ini, istilah *Artificial Intelligence* sengaja tidak diterjemahkan kedalam bahasa Indonesia karena istilah tersebut sudah sangat akrab bagi orang Indonesia. Begitu juga dengan singkatan istilah tersebut, yaitu AI, sudah sangat melekat di berbagai media ilmiah maupun no ilmiah.

Stuart Russel dan Peter Norvig mengelompokkan definisi AI, yang diperoleh dari beberapa *textbook* yang berbeda, kedalam empat kategori, yaitu [6]:

1. *Thinking humanly : the cognitive modeling approach*
2. *Acting humanly : the turing test approach*
3. *Thinking rationally : the laws of thought approach*
4. *Acting rationally : the rational agent approach*

Para ilmuwan memiliki dua cara sudut pandang yang berlainan tentang AI. Yang pertama adalah AI sebagai bidang ilmu yang hanya fokus pada proses berpikir. Sedangkan yang kedua adalah memandang AI sebagai bidang ilmu yang fokus pada tingkah laku.

### 2.2.1 Definisi AI

Definisi AI yang paling tepat saat ini adalah *acting rationally* dnegan pendekatan *human agent* [1]. Hal ini berdasarkan pemikiran bahwa komputer

dapat melakukan penalaran secara logis dan dapat melakukan aksi secara rasional berdasarkan hasil penalaran tersebut.

### 2.2.2 Perkembangan AI

Sejak pertama kali dikemukakan istilah AI pada tahun 1965 di konferensi Darthmouth, AI terus dikembangkan melalui berbagai penelitian mengenai teori-teori dan prinsip-prinsipnya. Perkembangan AI mengalami pasang surut mengikuti antusias para peneliti dan dana penelitian yang tersedia. Pada periode 1966 sampai 1974, perkembangan AI melambat, tetapi sejak tahun 1980, AI menjadi sebuah industri skala besar yang melakukan investasi besar-besaran dalam bidang AI.

Saat ini, dengan semakin cepatnya perkembangan perangkat keras dan perangkat lunak pada komputer, berbagai produk AI telah berhasil dibangun dan digunakan dalam kehidupan sehari-hari. Dengan teknologi perangkat keras yang performansinya semakin tinggi dan berukuran kecil serta didukung teknologi perangkat lunak yang semakin beragam dan kuat, produk-produk berbasis AI semakin dekat dengan kehidupan manusia [1].

Pada masa mendatang, AI ditantang untuk membuat suatu kecerdasan yang hampir menyerupai kecerdasan manusia. Ray Kuzwell memprediksi bahwa hal itu akan semakin mungkin terwujud melalui tahapan-tahapan prediksi yang dibuatnya secara bertahap sampai sekitar 90 tahun (tahun 2099).

Sebagian prediksi pada tahapan pertama sudah mulai berhasil diwujudkan. Perangkat komputer yang semakin kecil dalam genggaman, seperti PDA dan *smartphone*, sudah banyak digunakan secara komersil dengan harga terjangkau.

### 2.2.3 Teknik Pemecahan Masalah AI

Terdapat empat teknik dasar dalam pemecahan masalah yang terdapat pada bidang academic/*Traditional Artificial Intelligence* [1] diantaranya adalah :

*a. Searching*

Pada teknik *searching* atau pencarian ini terdiri dari beberapa langkah untuk merealisasikannya. Langkarh pertama adalah mendefinisikan ruang masalah untuk suatu masalah yang dihadapi. Langkah kedua adalah mendefinisikan aturan produksi yang digunakan untuk mengubah suatu keadaan ke keadaan lainnya. Langkah terakhir adalah memilih metode pencarian yang tepat sehingga dapat menemukan solusi terbaik.

*b. Reasoning*

Teknik *reasoning* atau penalaran merupakan teknik penyelesaian masalah dengan cara mempresentasikan masalah ka dalam basis pengetahuan (*knowledge base*) menggunakan *logic* atau bahasa formal (bahasa yang dipahami komputer). Teknik ini melakukan proses penalaran berdasarkan basis pengetahuannya untuk menemukan solusi.

*c. Planning*

*Planning* adalah suatu metode penyelesaian masalah dengan cara memecah masalah ke dalam sub-sub masalah yang lebih kecil, menyelesaikan sub-sub masalah satu demi satu, kemudian mengabungkan solusi-solusi dari sub-sub masalah tersebut menjadi sebuah solusi lengkap dengan tetap mengingat dan menangani interaksi yang terdapat pada sub-sub masalah tersebut.

*d. Learning*

Pada ketiga teknik sebelumnya, seseorang harus mengetahui aturan yang berlaku untuk sistem yang akan dibangunnya. Tetapi, pada masalah tertentu terkadang suatu aturan tidak bisa didefinisikan secara benar ataupun lengkap. Melalui teknik yang disebut learning ini, secara otomatis aturan yang diharapkan bisa berlaku umum untuk data-data yang belum pernah diketahui dapat ditemukan.

## *2.3 Algoritma A**

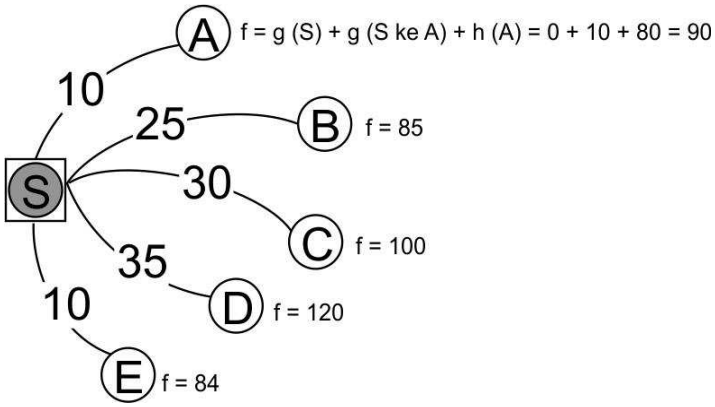
Dalam ilmu komputer, metode A* (A Star) adalah *graph search algorithm* yang mencari *path* (jalur) dari titik awal yang diberikan menuju titik tujuan [1]. Metode A* adalah metode yang merupakan hasil pengembangan dari metode dasar *Best First Search*. Metode ini mengevaluasi setiap titik dengan mengkombinasikan dengan g(n), nilai untuk mencapai titik n dari titik awal, dan h(n), nilai perkiraan untuk mencapai tujuan dari titik n tersebut. Dalam notasi matematika dituliskan sebagai : f(n) = g(n) + h(n). Dengan perhitungan biaya seperti ini, algoritma A* adalah *complete* dan optimal.

Ketika g(n) memberikan hasil evaluasi nilai untuk mencapai titik n, dan h(n) memberikan nilai estimasi untuk mencapai tujuan dari titik n, maka didapatkan f(n) = nilai estimasi yang terkecil yang melewati titik n.

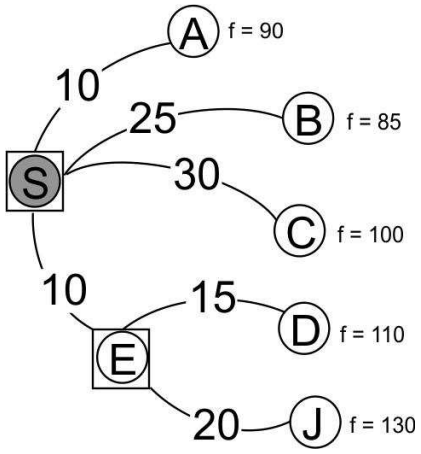
### 2.3.1 Cara Kerja Algoritma A*

Untuk memperjelas pemahaman mengenai algoritma A* (A Star), maka berikut ini adalah langkah-langkah pencarian rute berdasarkan algoritma ini yang

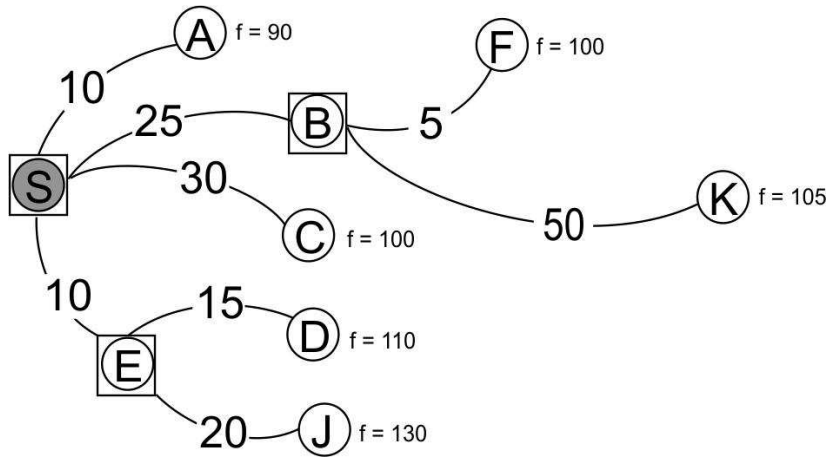
dapat dilihat pada gambar 2.2, gambar 2.3, gambar 2.4, gambar 2.5, gambar 2.6 dan gambar 2.7[6] :



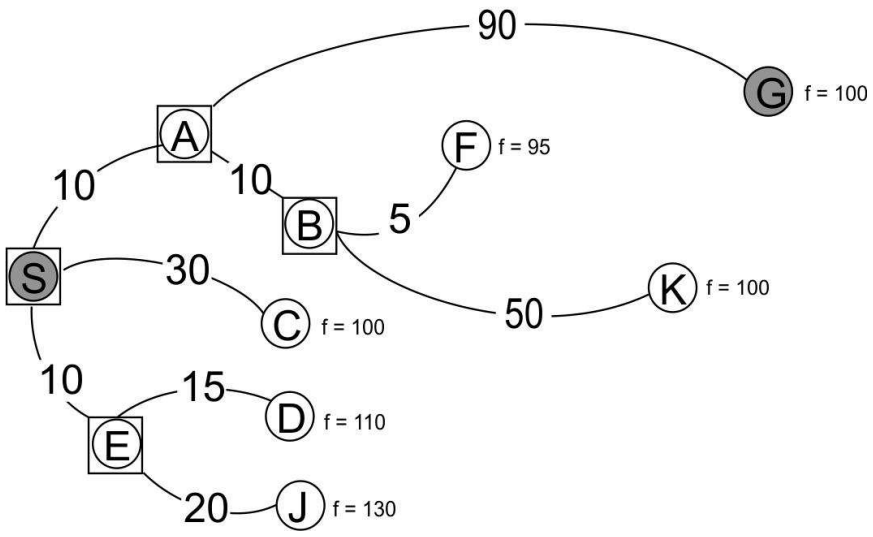

**Gambar 2.1 Langkah 1 pencarian rute algoritma A***



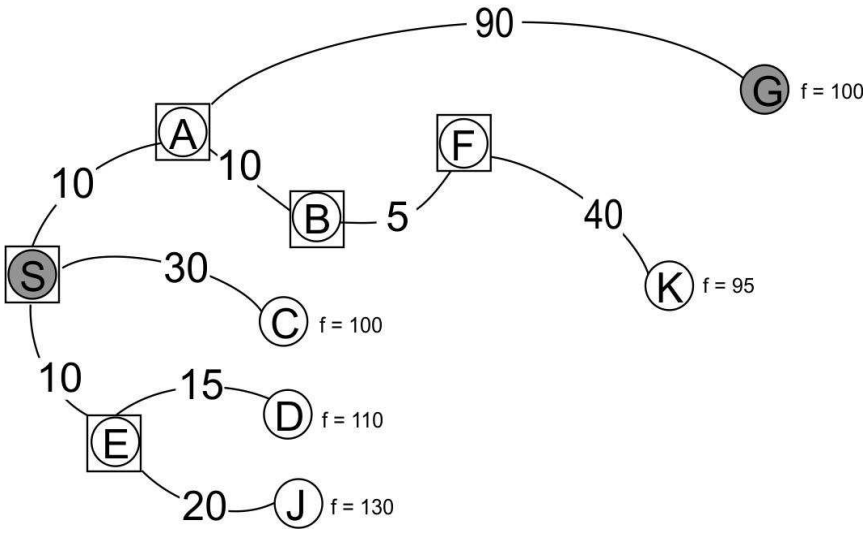

**Gambar 2.2 Langkah 2 pencarian rute algoritma A***



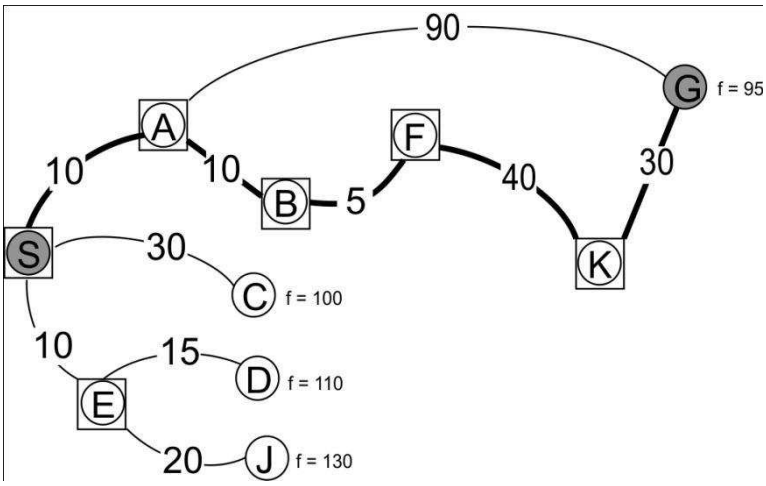

**Gambar 2.3 Langkah 3 pencarian rute algoritma A***




**Gambar 2.4 Langkah 4 pencarian rute algoritma A***




**Gambar 2.5 Langkah 5 pencarian rute algoritma A***




**Gambar 2.6 Langkah 6 pencarian rute algoritma A***

Simpul dengan tanda kotak menyatakan bahwa simpul tersebut berada di *CLOSED* (pernah terpilih sebagai *BestNode*) sedangkan simpul tanpa tanda kotak menyatakan simpul yang ada berada di *OPEN*.

Untuk langkah pertama, karena di *OPEN* hanya terdapat satu simpul yaitu S, maka S langsung terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*.Kemudian dibangkitkan semua suksesor S, yaitu : A, B, C, D dan E.

Karena kelima suksesor tidak ada di *OPEN* dan *CLOSED*, maka kelimanya dimasukkan ke *OPEN*. Langkah pertama ini menghasilkan *OPEN* = [A, B, C, D, E] dan *CLOSED* = [S].

Langkah kedua, E dengan biaya terkecil (yaitu 84) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Lalu semua suksesor E dibangkitkan, yaitu : D dan J. Karena belum pernah ada di *OPEN* maupun *CLOSED* sebelumnya maka J dimasukkan ke *OPEN*. Sedangkan simpul D sudah ada di *OPEN*, maka harus dicek apakah *parent* dari D perlu diganti atau tidak. Ternyata biaya dari S ke D melalui E (yaitu 10 + 15 = 25) lebih kecil daripada biaya S ke D (yaitu 35). Oleh karena itu, *parent* dari D harus dirubah, yang semula S menjadi E. Dengan perubahan *parent* ini maka nilai g dan f pada D juga diperbaharui (nilai g yang semula 35 menjadi 25, dan nilai f dari 120 menjadi 110). Langkah kedua ini menghasilkan *OPEN* = [A, B, C, D, J] dan *CLOSED* = [S, E].

Langkah ketiga, B dengan biaya terkecil (yaitu 85) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Lalu semua suksesor B dibangkitkan, yaitu : A, F dan K. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka F dan K dimasukkan ke *OPEN*. Sedangkan simpul A sudah ada di *OPEN*, maka harus dicek apakah *parent* dari A perlu diganti atau tidak. Ternyata, biaya dari S ke A melalui B (yaitu 25 + 10 = 35) lebih besar daripada biaya S ke A (yaitu 10). Oleh karena itu, *parent* dari A tidak perlu diubah (tetap S). Akhir dari langkah ke tiga ini menghasilkan *OPEN* = [A, C, D, F, J, K] dan *CLOSED* = [S, E, B].

Langkah keempat, A dengan biaya terkecil (yaitu 90) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Lalu semua suksesor A dibangkitkan,

yaitu : B dan G. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka G dimasukkan ke *OPEN*. Sedangkan simpul B sudah ada di *CLOSED*, maka harus dicek apakah *parent* dari B perlu diganti atau tidak. Ternyata biaya dari S ke B melalui A (yaitu 10 + 10 = 20) lebih kecil daripada biaya dari S ke B (yaitu 25). Oleh karena itu parent dari B harus diubah, yang semula S menjadi A. Nilai g dan f pada B juga harus diperbarui (nilai g yang semula 25 menjadi 20, dan nilai f dari 85 menjadi 80). Nilai g dan f pada suksesor-suksesor B (sampai anak cucu) juga harus diperbarui menggunakan penulusuran DFS *(Depth First Search).* Dalam kasus ini, B hanya mempunyai dua anak (tidak punya cucu), yaitu F dan K. Nilai g (F) yang semula 30 diubah menjadi 25, dan nilai f (F) dari 100 menjadi 95. Nilai g (K) yang semula 75 diubah menjadi 70, dan nilai f (K) dari 105 menjadi 100). Akhirnya, *OPEN* = [C, D, F, G, J, K] dan *CLOSED* = [S, E, B, A].

Langkah kelima, F dengan biaya terkecil (yaitu 95) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Lalu semua suksesor F dibangkitkan, yaitu : K. Karena K sudah ada di *OPEN*, maka harus dicek apakah parent dari K perlu diganti atau tidak. Biaya dari S ke K melalui F ternyata lebih kecil daripada biaya dari S ke K melalui *parent* lama (B). Oleh karena itu *parent* dari K harus diubah, yang semula B menjadi F. Selanjutnya nilai g (K) yang semula 70 diubah menjadi 65, dan nilai f (K) dari 100 menjadi 95). Akhirnya, *OPEN* = [C, D, F, G, J, K] dan *CLOSED* = [S, E, B, A, F].

Langkah keenam, K dengan biaya terkecil (yaitu 95) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Lalu semua suksesor K dibangkitkan, yaitu G. Karena G sudah ada di *OPEN*, maka harus dicek apakah *parent* dari G

perlu diganti atau tidak. Biaya dari S ke G melalui K ternyata lebih kecil daripada biaya dari S ke G melalui *parent* lama (yaitu A). Oleh karena itu *parent* dari G harus diubah, yang semula A menjadi K. Selanjutnya nilai g (G) yang semula 100 diubah menjadi 95, dan nilai f (G) dari 100 menjadi 95). Pada akhir langkah keenam ini, *OPEN* = [C, D, G, J] dan *CLOSED* = [S, E, B, A, F, K].

Selanjutnya G dengan biaya terkecil (yaitu 95) terpilih sebagai *BestNode*. Karena *BestNode* sama dengan goal, berarti solusi sudah ditemukan. Rute dantotal biaya bisa ditelusuri balik dari G menuju S karena setiap simpul hanya memiliki satu *parent* dan setiap simpul memiliki informasi biaya sebenarnya (g). Penelusuran balik menghasilkan rute S-A-B-F-K-G dengan total jarak sama dengan 95 kilometer. Rute ini merupakan rute terpendek yang ada di graphtersebut. Jadi, algoritma A* (A STAR) adalah optimal. Tanpa ada batasan waktu dan memory, A* (A STAR) adalah *complete* (selalu menemukan solusi jika solusinya ada). Pada kasus tersebut A* (A STAR) membangkitkan dan menyimpan 10 simpul (dari 13 simpul yang ada pada graph). Untuk masalah yang lebih kompleks, misalkan pencarian rute terpendek pada *graph* yang terdiri dari 100 juta simpul, A* (A STAR) akan menghadapi masalah waktu proses dan memori yang dibutuhkan. Untuk menyelesaikan kedua masalah tersebut, maka telah diusulkan berbagai variasi A* (A STAR) dengan karakteristik yang sesuai untuk permasalahan tertentu yang muncul [1].

### *2.4 Game Engine*

Teknologi *game* dewasa ini memang sangat populer, bukan hanya dari segi permainan, grafik, serta efek-efek yang sangat bagus. Untuk menghasilkan game

yang layak untuk dimainkan dan juga *'eye canding'*, tentunya membutuhkan sebuah perangkat bantu seperti *Game engine*.

*Game engine* merupakan sebuah *software* yang memang sengaja dibuat untuk keperluan pembuatan dan pengembangan sebuah *game*. Dengan adanya *Game engine* ini, memudahkan seorang pengembang untuk merancang *game* mulai dari *rendering*, *visual*, arsitektur, AI, dan bahkan sistem jaringan *game* itu sendiri. Jika *game* adalah jiwa para *gamer*, maka *Game engine* adalah jiwa dari sebuah *game.*

Dalam *Game engine* sendiri terdapat tools atau alat pendukung yang berguna untuk rancang bangun sebuah *game*, antara lain : mesin *render* 2D dan 3D, *script*, *collision detection*, suara, grafik, animasi, kecerdasan buatan, manajemen memori, jaringan streaming.

Pada perkembangannya, *Game engine* ini ada yang berbayar dan ada juga yang gratis alias *opensource*. Berikut adalah contoh-contoh *Game engine* yang *opensource* dan berbayar .

Berikut beberapa *Game engine* yang beredar sekarang ini :

1. *OGRE*
2. *Panda 3D*
3. *crystal space*
4. *blender Game engine*
5. *OPEN SceneGraph, dll.*
6. *C4 Engine*
7. *Game Maker*

8. *torque Game engine*
9. *3DGamestudio*
10. *DXStudio*
11. *neoExis engine*
12. *esenthel engine, dll*

### 2.4.1 *Game Maker*

*Game maker* atau biasa disingkat ”GM” adalah sebuah *software* komputer yang bisa memberikan akses agar seseorang bisa membuat *game* hanya dengan *metode* mudah yaitu *Drag & Drop*.

*Game Maker* merupakan suatu program yang bisa dikatakan cukup simple dan mudah untuk digunakan. Dengan program ini seseorang dapat menciptakan *game* jenis apa saja yang inginkan seperti: *Racing, Action, Fighting, RPG, Adventure, Puzzle, Tactic, Simulation*. Program ini lebih *focus* pada *game* berjenis 2D tetapi program ini juga menyediakan *feature* yang memungkinkan anda untuk membuat *game* jenis 3D[8].

*Game Maker* dirancang untuk memungkinkan pengguna untuk dengan mudah mengembangkan permainan komputer tanpa harus belajar yang kompleks bahasa pemrograman seperti *C + +* atau *Java* . Namun, adalah mungkin untuk membuat *add-ons* untuk *game* dibuat dengan *Game Maker*. Ekstensi tersebut, sebagaimana mereka dikenal untuk pengguna *Game Maker*, memberikan pengguna seperti 3D maju melalui *Port Rendering* Mesin Ogre, suara yang lebih baik, antara lain. Ekstensi untuk IDE utama adalah terbatas pada penciptaan *drag dan drop* naskah baru tindakan.

Untuk pengguna yang berpengalaman atau mereka dengan pengalaman pemrograman komputer, *Game Maker* berisi built-in bahasa pemrograman *scripting* yang disebut *GML (Game Maker Language)*, yang memungkinkan membangun permainan yang lebih kompleks..

*Game Maker* awalnya dikembangkan oleh Prof Mark Overmars pada tahun 1999. *Game Maker* adalah sebuah program yang digunakan untuk membuat sebuah *game* yang khusus untuk komputer. *Game Maker* menggunakan bahasa *GML* atau *Game Maker Language*. Namun, juga bisa menggunakan LIB atau *Libraries*. Libraries adalah *action-action* yang memang telah tersedia di *Game Maker*.

### *2.4.2 Game Maker Language*

*Game Maker Language* atau disingkat *GML* adalah bahasa pemrograman yang *built-in* dalam IDE *Game Maker*. Dengan sedikit saja kita menguasai bahasa *GML* (*Game maker Language*) maka kita akan mampu membuat *Game* yang *professional[8]*. Ada beberapa perbedaan dalam tempat mengetikan script dalam *GML*. Pertama, ketika mendefinisikan script. Sebuah script adalah sebuah program dalam GML. Kedua, ketika akan menambahkan kode aksi kepada sebuah *event.*

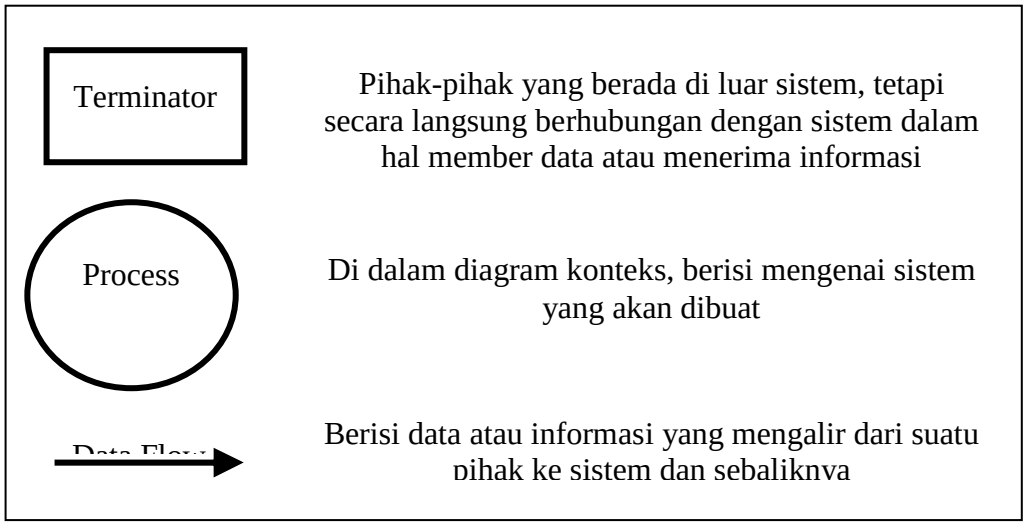
## *2.5 Diagram Konteks*

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau *output* dari sistem[14].

Diagram tersebut akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks.

Diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi "siapa saja yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem."

Jadi, yang dibutuhkan dalam pembangunan diagram ini adalah Siapa saja pihak yang akan memberikan data ke sistem, Data apa saja yang diberikannya ke sistem, kepada siapa sistem harus memberi informasi atau laporan, dan apa saja isi/ jenis laporan yang harus dihasilkan sistem. Kata "Siapa" di atas dilambangkan dengan kotak persegi (disebut dengan *terminator*), dan kata "apa" di atas dilambangkan dengan aliran data (disebut dengan *data flow*), dan kata "sistem" dilambangkan dengan lingkaran (disebut dengan *process*). Elemen-elemen dari Diagram Konteks tersebut dapat dilihat pada Gambar 2.6.




**Gambar 2.7 Elemen diagram konteks**

### *2.6 Data Flow Diagram*

DFD atau Diagram Arus Data adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut[14].

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

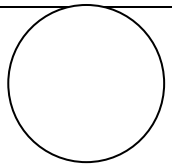
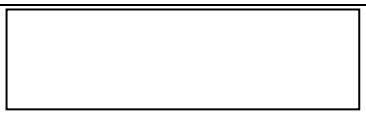
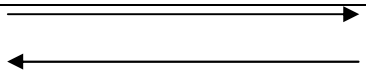
1. Kebebasan dari menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu :

1. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.
2. Membedakan sistem dari lingkungannya dengan menempatkan batas-batasnya.
3. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
4. Memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

Simbol-simbol DFD dapat dilihat pada Tabel 2.1.

**Tabel 2.1 Simbol-simbol data flow diagram**

| No | Simbol | Fungsi |
|---|---|---|
| 1. |  | Proses, digunakan untuk menggambarkan sebuah sistem atau proses. |
| 2. |  | Entitas Luar, digunakan untuk menggambarkan sebuah objek. |
| 3. |  | Penyimpanan, digunakan untuk menggambarkan/ menunjukkan sebuah file data |
| 4. |  | Arus data, digunakan untuk menggambarkan sebuah arus data. |

## *2.7 CorelDraw*

*Coreldraw* merupakan salah satu aplikasi pengolah gambar berbasis vector yang banyak dipakai oleh pengguna PC. Karena berbagai kemudahan dan keunggulan yang dimiliki oleh *coreldraw*, maka *coreldraw* sering dimanfaatkan untuk desktop publishing, percetakan, dan bidang lain yang memerlukan pemrosesan visual. Keunggulan mengolah gambar berbasis vektor adalah ukuran hasil akhir yang dapat ditekan seminimal mungkin namun dengan kualitas yang

tidak kalah dengan gambar berbasis raster atau bitmap. Karena itulah desain grafis dan olah gambar berbasis vektor sangat banyak digunakan untuk desktop publishing, percetakan, dan bidang lain yang memerlukan pemrosesan visual[15].

Sebenarnya ada banyak sekali software desain grafis dan pengolah gambar berbasis vektor yang dapat digunakan, misalnya saja diantaranya adalah *Coreldraw, Photoshop, Freehand, adobe ilustrator dll*. Namun *coreldraw* merupakan program paling populer yang sering dimanfaatkan. Selain karena tampilannya yang *user friendly* dan mudah dipelajari, *coreldraw* juga mempunyai berbagai keunggulan – keunggulan lain yang sangat bermanfaat.

Beberapa keunggulan *coreldraw* antara lain *coreldraw* sangat bagus dalam kolaborasi teks dan gambar. Meskipun photoshop juga bisa namun lebih mudah dengan *coreldraw*. Selain itu kelebihan *coreldraw* yang jarang dimiliki oleh aplikasi desain grafis dan pengolah gambar yang lain adalah dukungan forum dan komunitas *coreldraw* yang begitu banyak dan beragam, sehingga bisa dengan mudah menemukan *tutorial*, tips dan bisa berbagi ilmu dengan pengguna *coreldraw* lainnya di seluruh dunia.

### *2.8 Adobe Photoshop*

*Adobe Photoshop*, atau biasa disebut Photoshop, adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek[16]. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar/foto, dan, bersama Adobe Acrobat, dianggap sebagai produk terbaik yang pernah diproduksi

oleh Adobe Systems. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (Creative Suite), versi sembilan disebut *Adobe Photoshop* CS2, versi sepuluh disebut *Adobe Photoshop* CS3 , versi kesebelas adalah *Adobe Photoshop* CS4 , versi keduabelas adalah *Adobe Photoshop* CS5 , dan versi yang terakhir (ketigabelas) adalah *Adobe Photoshop* CS6.

Photoshop tersedia untuk Microsoft Windows, Mac OS X, dan Mac OS; versi 9 ke atas juga dapat digunakan oleh sistem operasi lain seperti Linux dengan bantuan perangkat lunak tertentu seperti *CrossOver*.

Beberapa tool - tool yang ada di Photoshop antara lain :

1. *History Brush Tool*
2. *Eraser Tool*
3. *Path Selection Tool*
4. *Direct Selection tool*
5. *Pen Tool*
6. *Shape Tool*
7. *Brush Tool*
8. *Audio Annotation Tool*
9. *Eyedropper Tool*
10. *Measure Tool*

### *2.9 Skala Likert*

Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian,

fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai *variabel* penelitian[17].

Dengan *skala Likert*, maka *variabel* yang akan diukur dijabarkan menjadi indikator *variabe*l. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun *item-item instrumen* yang dapat berupa pernyataan atau pertanyaan.

Jawaban setiap item instrumen yang menggunakan skala Likert mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapa berupa kata-kata antara lain :

a. Sangat Setuju
b. Setuju
c. Kurang Setuju
d. Tidak Setuju

a. Selalu
b. Sering
c. Kadang-kadang
d. Tidak Pernah

a. Sangat Positif
b. Positif
c. Negatif
d. Sangat Negatif