BAB 2
LANDASAN TEORI

2.1 Game

Game merupakan salah satu media hiburan yang paling popular untuk semua kalangan usia. Sejak pertama kali ditemukan sampai saat sekarang, teknologi *game* telah mengalami kemajuan yang terbilang pesat. Hal ini ditandai dengan berkembangnya jenis *game*, produk, alat dan jenis interaksi *game* dengan penggunaan yang semakin beragam bentuknya.

2.1.1 Pengertian Game

Game merupakan sebuah bentuk seni dimana penggunanya disebut dengan pemain (*player*), diharuskan membuat keputusan-keputusan dengan tujuan mengelola sumber daya yang diperoleh dari kesempatan-kesempatan bermain (*token*) miliknya untuk mencapai tujuan tertentu. Video *game* adalah bentuk *game* yang interaksi umumnya melibatkan media video dan audio.

Menurut Andang terdapat dua pengertian *game*. Pertama, *game* (permainan) adalah sebuah aktivitas bermain yang murni mencari kesenangan tanpa mencari menang atau kalah. Kedua, permainan diartikan sebagai aktifitas bermain yang dilakukan dalam rangka mencari kesenangan dan kepuasan, namun ditandai pencarian menang - kalah.

Berdasarkan pengertian - pengertian di atas, *game* merupakan sebuah aktivitas dalam rangka mencari kesenangan dimana penggunanya diharuskan membuat keputusan-keputusan untuk mencapai tujuan tertentu dengan ditandai menang - kalah [5].

2.1.2 Jenis – jenis *Game* Berdasarkan Platform

Berikut ini beberapa jenis *game* berdasarkan cara pembuatannya, cara pemasarannya dan mesin yang menjalankannya [4]. Jenis-jenis *game* tersebut adalah:

1. *Game* PC

   Game PC adalah *game* yang dimainkan pada PC (*Personal Computer*) yang memiliki kelebihan yaitu tampilan antarmuka yang baik untuk *input* maupun *output*.*Output* visual berkualitas tinggi karena layar komputer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televisi biasa.Kekurangannya adalah spesifikasi komputer yang sangat bervariasi antar satu komputer dengan komputer yang lainnya menyebabkan beberapa *game* dapat ditampilkan dengan baik pada satu komputer tetapi tidak berjalan dengan baik pada komputer yang lainnya.

2. *Game Console*

   Game console adalah *game* yang dijalankan pada suatu mesin spesifik yang biasanya tersedia di rumah seperti Xbox,Nintendo Wii dan lain-lain.

3. *Game Arcade*

   Game arcade adalah *game* yang dijalankan pada mesin dengan *input* dan *output* audio visual yang telah terintegrasi dan tersedia ditempat-tempat umum.

4. *Game Online*

   Game online adalah *game* yang hanya dapat dumainkan secara *online* melaui LAN atau internet.

2.1.3 *Genre Game*

Berdasarkan *genre* permainannya*, game* dapat dibagi menjadi beberapa *genre* yaitu [4]:

1. *Action*

   Action game adalah *game* dimana kebanyakan dari tantangan yang disajikan merupakan dari tes *physical skill* dan koordinasi pemain. Salah satu *sub-genre action game* adalah *shooter game*, baik yang 2D ataupun 3D seperti *First Person Shooter (FPS)*.

3. *Strategy*

   Strategy game menantang pemain untuk mencapai kemenangan dengan perencanaan, khususnya melalui perencanaan serangkaian tindakan yang dilakukan melawan satu lawan atau lebih. Kemenangan diraih dengan perencanaan matang dan pengambilan keputusan optimal.

4. *Role Playing Game* (RPG)

   RPG adalah *game* dimana pemain mengontrol satu atau lebih karakter yang biasanya di desain oleh pemain itu sendiri, dan memandu mereka melewati berbagai rintangan yang diatur oleh komputer. Perkembangan karakter dalam hal kekuatan dan kemampuannya adalah kunci dari *game* jenis ini.

5. *Sports*

   Sports game mensimulasikan berbagai aspek dari olahraga atletik nyata ataupun imajiner, apakah itu memainkan pertandingan, me-*manage* tim dan karir, atau keduanya. Salah satu contoh *game* jenis ini adalah *Pro Evolution Soccer 2012* (PES 2012), dimana pemain bisa memainkan pertandingan, menjadi manajer tim, maupun menjadi pemain dan mengembangkan karirnya sendiri.

6. *Vehicle Simulation*

   Vehicle simulation membuat *feeling* mengendarai kendaraan, baik kendaraan nyata maupun kendaraan imajiner. Performa dan karakteristik kasar mesin harus menyerupai kenyataan, kecuali jika yang didesain adalah kendaraan imajiner.

7. *Construction and Management Simulation*

   CMS game adalah *game* tentang proses. Tujuan pemain bukan untuk mengalahkan musuh, tetapi membangun sesuatu dengan konteks proses yang sedang berjalan. Semakin pemain mengerti dan mengontrol proses, semakin sukses sesuatu yang ia bangun. *Game* seperti ini biasanya menyediakan dua jenis permainan, yaitu mode bebas dimana pemain bebas membangun sesuatu, dan mode misi dimana terdapat skenario hal apa yang harus dibangun oleh pemain.

8. *Adventures*

   Adventure game adalah cerita interaktif tentang karakter protagonis yang dimainkan oleh pemain. Penyampaian cerita dan eksplorasi adalah elemen inti dari *game* ini. Penyelesaian teka-teki dan tantangan konseptual adalah bagian besar dari permainan.

9. *Artificial Life and Puzzle Game*

   Artificial Life game adalah *game* yang membuat tiruan dari kehidupan sebenarnya. Biasanya ada dua jenis *game* ini, tiruan kehidupan manusia, contohnya The SIMS, dan tiruan kehidupan binatang, contohnya Tamagochi.

10. *Online Game*

    Istilah *online game* disini mengacu kepada *multiplayer game* dimana mesin dari para pemain terhubung dengan jaringan.

2.2 *Game Adventure*

Game petualangan adalah sebuah cerita interaktif tentang karakter protagonist yang dimainkan oleh pemain. Cerita dan eksplorasi sangat penting dalam unsur permainan. Memecahkan teka-teki dan tantangan konseptual membentuk sebagian besar gameplay seperti combat, manajemen ekonomi, dan konflik tantangan yang kurang atau tidak ada. Definisi ini tidak berarti bahwa tidak ada konflik di game petualangan (walaupun banyak game petualangan tidak punya) seperti hanya bertempur yang bukan merupakan aktivitas utama. Game petualangan jarang memiliki ekonomi internal. Semua hubungan di dalam permainan bersifat simbolis daripada angka. Memanipulasi atau mengoptimalkan suatu ekonomi bentuk sistem tidak ada bagian dari game petualangan hal ini yang membedakan mereka dari role-playing *game*.

Fitur dalam game petualangan, avatar pemain mengunjungi daerah *explorable* yang mengandung bermacam-macam teka-teki atau masalah untuk memecahkan. Pemecahan masalah ini membuka daerah baru untuk eksplorasi atau kemajuan alur cerita, memberikan informasi baru pemain dan untuk memecahkan masalah baru. Menjelajahi lingkungan dan memanipulasi *item* di dalamnya merupakan elemen penting dari sebuah *game* petualangan. Banyak pemain juga

menikmati berinteraksi dengan berbagai karakter. Jenis lebih berbeda dari permainan game petualangan mengandung kaya akan kualitas yang berbagi dengan Peran games. Game petualangan biasanya hanya menawarkan beberapa mode gameplay. Tidak seperti olahraga permainan, dengan semua fungsi yang terkait tim manajemen, atau latihan perang, dengan terkait mode pertempuran-perencanaan, game petualangan tidak perlu banyak khusus layar. Terlepas dari kebutuhan untuk melihat peta atau persediaan avatar atau untuk memeriksa benda dekat, pemain selalu melihat dan berinteraksi dengan dunia dengan cara sama, dan yang tidak berubah dari satu ujung pertandingan yang lain. Mayoritas tantangan dalam sebuah game petualangan yang konseptual seperti teka-teki yang bisa hanya diselesaikan dengan berpikir lateral.

Percakapan dengan *non-player character* didefinisikan oleh peran karakter seperti pandai besi, penyembuh, kedai kiper, dan sebagainya. Pemain tidak mengharapkan untuk dapat mendiskusikan senjata dan baju besi dengan penjaga kedai (Meskipun permainan mungkin akan lebih menarik dan tentunya kurang baku jika ia bisa). Tetapi karena game petualangan cerita interaktif, pemain mengharapkan karakter di dalamnya untuk menjadi lebih seperti manusia dan kurang mekanis. Beberapa game berbasis teks awal mencoba menerapkan *parser* yang bisa memahami kalimat terbatas. Pada akhirnya, kebanyakan desainer game petualangan menyerah berusaha untuk menciptakan kesan bahwa pemain bisa berbicara dengan siapa pun tentang apa saja dan merancang scripted percakapan, mekanisme yang menjadi standar de facto untuk kedua game petualangan dan CRPG [4].

2.3 Teuku Umar

Teuku Umar dilahirkan di Meulaboh Aceh Barat pada tahun 1854. Ayahnya bernama Achmad Mahmud yang berasal dan keturunan Uleebalang Meulaboh. Ibunya merupakan adik dari raja Meulaboh. Pada Permulaan Perang pada tahun 1871 Inggris dan Belanda bertemu dalam Traktat Sumatera. Dalam Traktat tersebut disebutkan bahwa Belanda bebas bergerak dan mengadakan perluasan wilayah di Aceh. Rakyat Aceh marah mengetahui perjanjian tersebut. Kemarahan itu sebenarnya sudah lama terasa setelah melihat gelagat dan gerak-

gerik Belanda di Sumatera yang merugikan Aceh telah terbaca sejak tahun 1857. Pada tahun itu Belanda menduduki Siak yang merupakan daerah taklukan Aceh. Setelah lahir Traktat Sumatra tahun 1871, rakyat Aceh semakin meluap-luap marahnya.

Pada tanggal 5 April 1873, Belanda dengan kekuatan 3000 orang tentara menyerang Kerajaan Aceh Darussalam dan berhasil menduduki Mesjid Raya Baiturrahman. Namun dapat direbut kembali oleh pejuang Aceh setelah Panglima tentara Belanda Mayor Jenderal JHR. Kohler ditembak mati oleh pejuang Aceh pada tanggal 14 April 1873. Dengan tewasnya JHR. Kohler, penyerbuan tidak diteruskan. Seluruh pasukan Belanda yang ada di Aceh akhirnya ditarik kembali. Pada bulan Nopember tahun 1873 dikirimlah ekspedisi kedua yang dipimpin oleh Van Swieten dengan tentara sebanyak 13.000 orang. Serbuan kali ini berhasil menduduki Mesjid Raya Baiturrahman. Sebelum Istana Raja oleh pasukan diserbu Belanda, Sultan dan seluruh penghuninya telah diungsikan. Dalam pengungsiannya, Sultan terserang penyakit kolera dan akhirnya meninggal. Para pengikutnya memindahkan tempat pengungsiannya sampai jauh ke pedalaman Aceh Besar. Dalam perangnya melawan gerilyawan Aceh, Belanda menggunakan strategi menunggu dan menjalankan sistem pasifikasi. Hal ini terjadi juga dengan adanya raja-raja daerah pantai yang menyatakan tunduk pada Pemerintah Kolonial Belanda.

Waktu Jenderal Van Der Heiden menggantikan pendahulunya, mulailah diadakan ofensif dengan mengirim ekspedisi. Perang akan dikobarkan di daerah 6 Mukim dalam tahun 1873. Teuku Umar yang waktu itu baru berumur 19 tahun, sebelum berangkat ke medan perang terlebih dahulu berpamitan pada kedua orang tuanya. Kedua orang tuanya meneteskan air mata haru bercampur bangga bahwa Teuku Umar telah berikrar bersama rakyat mengusir penjajah Belanda di Aceh dan berjuang sampai titik darah penghabisan. Pada waktu perang berkobar, prajurit-prajurit Aceh sangat bersemangat, walaupun persenjataanya sangat sederhana dibandingkan dengan prajurit Belanda, namun mereka dapat mendatangkan korban yang besar di pihak lawan. Kesemua itu dapat terjadi

karena antara lain disebabkan tentara Aceh mahir bertempur secara gerilya dan hidup di hutan belantara yang sudah mereka kuasai medannya.

Teuku Umar mulai memimpin perlawanan dengan berbagai siasat pada tahun 1883. Peristiwa itu sangat menggemparkan sehingga berita mengenai Teuku Umar menyerahan diri dan memihak kepada Belanda yang pada saat itu dipimpin Jendral Deykerhoof membuat rakyat Aceh marah dan banyak yang mengutuk sebagai pengkhianat diantara mereka ada pula yang menghendaki agar Teuku Umar dibunuh oleh rakyat sendiri. Sementara itu, Belanda sangat gembira menerima penyerahan diri Teuku Umar. Dengan menyerahnya Teuku Umar, Belanda berharap dapat dengan mudah menaklukkan seluruh rakyat Aceh. Setelah menyerahkan diri, maka Umar mendapat kepercayaan dari Belanda. Ia diserahi tugas yang penting-penting untuk melaksanakan keinginan Belanda menumpas perlawanan rakyat Aceh.

Pada mulanya tugas yang diberikan kepada Teuku Umar adalah melatih tentara Belanda bertempur di hutan belantara dan mengajarkan teknik perang gerilya. Teuku Umar melaksanakan tugas dengan sebaik-baiknya, tetapi di dalam hatinya ia memegang teguh siasat perang yang telah ditetapkan bersama dengan para pemimpin pejuang Aceh beberapa waktu sebelumnya. Selesai melatih perang gerilya di hutan belantara, Teuku Umar ditugaskan memimpin penumpasan perlawanan rakyat Aceh. Dalam pertempuran itu memang banyak korban jatuh di kedua belah pihak, tetapi tentara Belanda banyak yang mati dan senjatanya banyak yang berhasil dirampas tentara Aceh. Tentara Aceh hanya sebentar saja mampu melawan serangan tentara Belanda dan kemudian mereka mundur meninggalkan benteng pertahanannya. Apalagi tentara Aceh hanya berpura-pura saja berperang melawan tentara Umar. Demikian juga sebaliknya, Umar juga berpura-pura menyerang Aceh. Karena tidak tahu siasat Umar, Belanda gembira menyaksikan mundurnya tentara Aceh itu.

Belanda menganggap dengan bantuan Umar, mereka dapat mematahkan seluruh perlawanan Aceh. Untuk itu, Umar mendapat hadiah besar berupa uang dan materi lainnya yang berguna untuk menambah modal perang tentara Aceh yang dikirim secara rahasia. Setelah mendapatkan segala persenjataan dan

perlengkapan perang terpenuhi Teuku Umar kembali memihak rakyat Aceh. Mengetahui Teuku Umar hanya bersiasat untuk menyerah dan berpihak kepada Belanda. Belanda pun marah besar dan mengganti Jendral Deykerhoof dengan mendatangkan bantuan dari Batavia dan dipimpin langsung Jendral van Heutsz yang merupakan musuh bebuyutan Teuku Umar.

Pada bulan Februari 1899 Jenderal Van Heutsz berada di Meulaboh dengan tanpa pengawalan yang ketat sebagaimana biasanya. Keadaan ini diketahui oleh Teuku Umar dari mata-matanya yang bertugas di sana. Untuk menangkap dan mencegat Jenderal Belanda tersebut, Teuku Umar bersama sejumlah pasukannya datang ke Meulaboh. Tetapi malang bagi Umar karena sebelum rencananya berhasil dilaksanakan, gerak-gerik Umar justru telah diketahui oleh Belanda Setelah mendengar laporan dari mata-matanya mengenai kedatangan Teuku Umar di Meulaboh, Jenderal Van Heutsz segera menempatkan sejumlah pasukan yang cukup kuat diperbatasan kota Meulaboh untuk mencegat Teuku Umar. Pada malam menjelang tanggal 11 Februari 1899 Teuku Umar bersama pasukannya telah berada di pinggiran kota Meulaboh. Pasukan Aceh terkejut ketika mengetahui pasukan Van Heutsz telah mencegatnya. Posisi pasukannya sudah tidak menguntungkan dan tidak mungkin lagi untuk mundur. Satu-satunya jalan untuk menyelamatkan pasukannya adalah bertempur. Dalam pertempuran itu Teuku Umar gugur terkena peluru musuh yang menembus dadanya [3].

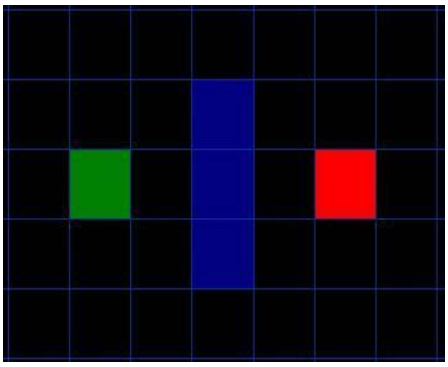
2.4 Algoritma A* (*A Star*)

Algoritma ini merupakan algoritma *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best First Search*. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. dengan perhitungan biaya seperti ini, algoritma A* adalah *complete* dan *optimal* [14]. Dalam notasi matematika dituliskan sebagai berikut:

$$f(n) = g(n) + h(n).$$

Dengan perhitungan biaya seperti diatas, algoritma A* adalah complete dan optimal. Untuk lebih jelasnya berikut merupakan gambaran dari penggunaan algoritma A* dalam pencarian rute terpendek dari titik A menuju titik B [10].

Asumsikan ada seseorang yang pergi dari point A menuju titik B dan rute yang akan dilewati dihalangi sebuah dinding. Pada ilustrasi ini kotak berwarna hijau merupakan titik A, kotak berwarna merah merupakan titik B, dan kotak berwarna biru merupakan dinding yang menghalangi rute titik A menuju titik B, untuk lebih jelasnya terdapat pada gambar 2.1.



Gambar 2.1 Ilustrasi awal algoritma [10]

Hal pertama yang harus dilakukan adalah membuat sebuah grid layout, karena dalam pencarian rute pada algoritma A* akan lebih mudah apabila peta yang ada berbentuk kotak-kotak kecil, metode ini memudahkan pencarian dalam area yang didefinisikan sebagai array dua dimensi, setiap kotak memiliki status walkable dan unwalkable.

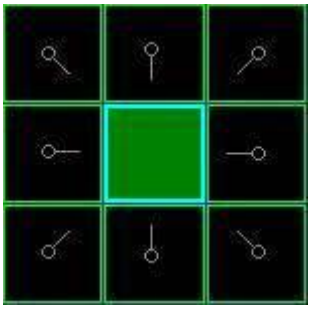
Pusat point pada algoritma A* dinamakan node, hal ini dikarenakan area yang dapat digunakan algoritma A* tidak hanya berbentuk kotak, akan tetapi dapat berbentuk yang lainnya, namun terkadang untuk memudahkan dalam perhitungan maka digunakanlah bentuk kotak.

Langkah-langkah yang dilakukan pada algoritma A* dalam melakukan pencarian rute terpendek dibagi menjadi beberapa tahap, yaitu :

a. Dimulai dari point A awal dan masukkan ke dalam kotak "*OPEN LIST*". "*OPEN LIST*" merupakan sebuah kotak yang berfungsi untuk menyimpan setiap node, baik yang akan kita lewati maupun yang tidak kita lewati, pada dasarnya setiap kotak yang ada pada "*OPEN LIST*" akan dilakukan perhitungan jarak untuk menentukan arah mana yang akan diambil menuju point B.

b. Lihat semua kotak yang memungkinkan untuk dilewati dari point awal, abaikan kotak yang terdapat penghalang seperti tembok, air, dan lain-lain. Masukan kotak-kotak tersebut ke dalam *OPEN LIST*, dan untuk setiap kotak ini, simpan point A sebagai induk kotak.
c. Keluarkan kotak point awal dari *OPEN LIST* kemudian masukkan ke dalam *CLOSED LIST*. *CLOSED LIST* berfungsi untuk menyimpan kotak yang tidak akan pernah dipakai ataupun dilihat lagi.

Kotak berwarna hijau merupakan pusat dari point awal, pada sisinya dikelilingi oleh garis berwarna biru yang berarti bahwa kotak tersebut telah dimasukkan ke dalam *CLOSED LIST*, dan untuk setiap garis disekeliling kotak awal menunjukkan setiap jalur yang dapat ditempuh dimulai dari kotak pada point awal, perhatikan pada gambar 2.2.



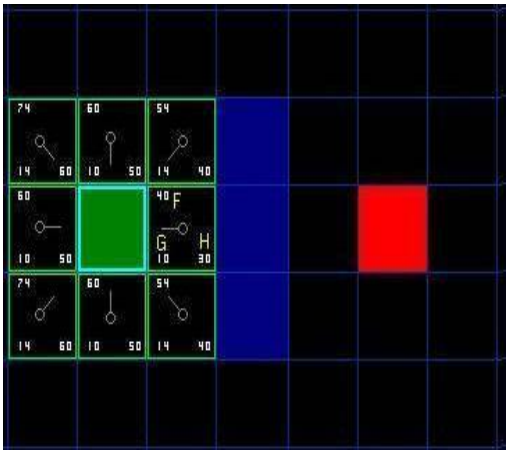
Gambar 2.2 Ilustrasi kotak awal sebagai closed list [10]

Perhitungan yang terdapat pada algoritma A* untuk melakukan pencarian rute terpendek adalah dengan menggunakan rumus F = G + H, dimana penjelasan dari rumus tersebut adalah sebagai berikut :

G : nilai yang dibutuhkan untuk menempuh jarak dari titik awal menuju kotak selanjutnya.
H : estimasi nilai yang dibutuhkan untuk menempuh jarak dari kotak selanjutnya untuk menempuh point tujuan.

Dalam proses perhitungannya dapat dimisalkan sebuah nilai untuk G baik yang terdapat pada kotak horizontal, vertikal, dan diagonal. Pada kasus ini dimisalkan nilai horizontal dan vertikal dari point awal untuk menuju kotak selanjutnya ialah 10, dan untuk nilai diagonalnya ialah 14.

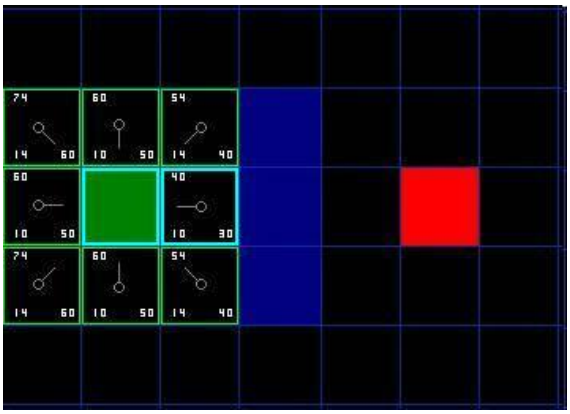
Setiap kotak memiliki nilai F, G, dan H masing-masing, jadi perhitungan tersebut berlaku bagi setiap kotak yang berada di sekitar kotak induk, untuk nilai F ditulis di atas sebelah kiri, nilai G ditulis di bawah sebelah kiri, dan nilai H ditulis di bawah sebelah kanan, seperti pada gambar 2.3.



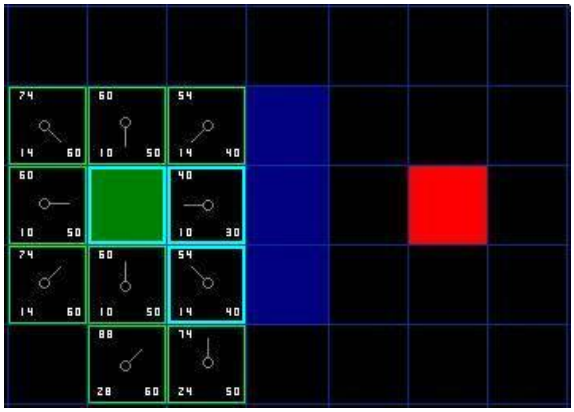
Gambar 2.3 Ilustrasi perhitungan F pada open list [10]

Langkah selanjutnya yang harus dilakukan adalah sebagai berikut :

d. Pilih nilai F yang terkecil dari setiap kotak yang telah dihitung nilai F nya, kemudian keluarkan dari *OPEN LIST* dan masukkan ke dalam *CLOSED LIST*.
e. Cek semua kotak yang berada di sekitar, abaikan kotak yang unwalkable dan kotak yang terdapat pada *CLOSED LIST*, masukkan kotak yang belum ada di dalam *OPEN LIST* dan jadikan kotak selanjutnya menjadi kotak induk untuk melakukan perhitungan ulang untuk setiap kotak disekitarnya.
f. Ketika sebuah kotak sekitar telah berada di dalam *OPEN LIST*, cek apakah jalur kotak tersebut adalah yang terbaik dibanding yang jalur kotak yang lainnya. Di lain pihak, ketika nilai G untuk jalur yang baru itu paling kecil, ganti induk kotak pada kotak sekitarnya menjadi kotak yang telah terpilih, dan terakhir hitung ulang nilai F, G, dan H untuk kotak tersebut, seperti pada gambar 2.4 dan perhitungan selanjutnya pada gambar 2.5.

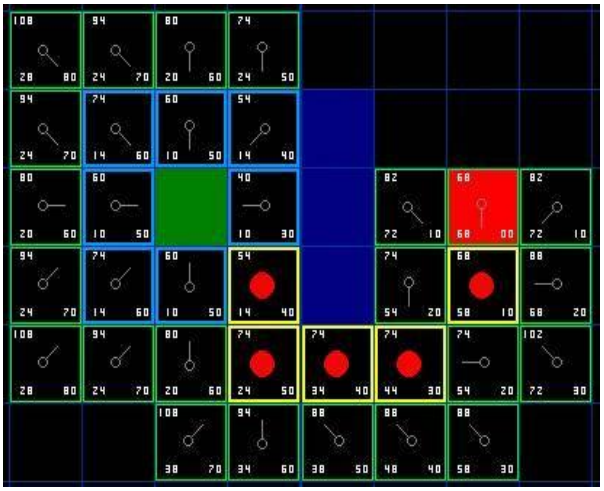


Gambar 2.4 Ilustrasi perhitungan selanjutnya [10]



Gambar 2.5 Ilustrasi perhitungan terhadap kotak unwalkable [10]

Setelah mengulangi setiap langkah yang ada di atas, perhitungan dapat dihentikan ketika point akhir atau yang disebut kotak target telah dimasukkan ke dalam *CLOSED LIST*, perhatikan gambar 2.6.



Gambar 2.6 Ilustrasi hasil akhir pencarian rute terpendek [10]

2.5 UML (*Unified Modeling Language*)

UML adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. Definisi ini merupakan definisi sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang bangun perangkat lunak efektif.

UML merupakan standar yang relatif terbuka yang dikontrol oleh O*bject Management Group (OMG),* sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar *COBRA (Common Object Request Broker Architecture).*

UML lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak kehadirannya pada tahun 1997, UML menggantikan menara Babel yang telah menjadi sejarah. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

Bagian-bagian utama dari UML adalah *view*, *diagram*, *model element*, dan *general mechanism* [6]. Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya. Berikut ini merupakan beberapa diagram yang terdapat dalam UML [8] :

1. *Use Case Diagram*

   Use Case mendeskripsikan interaksi tipikal antara para pengguna dengan sistem itu sendiri dan menggambarkan prosesnya.
2. *Activity Diagram*

Alur untuk menggambarkan rangkaian aliran dari aktivitas yang digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi.

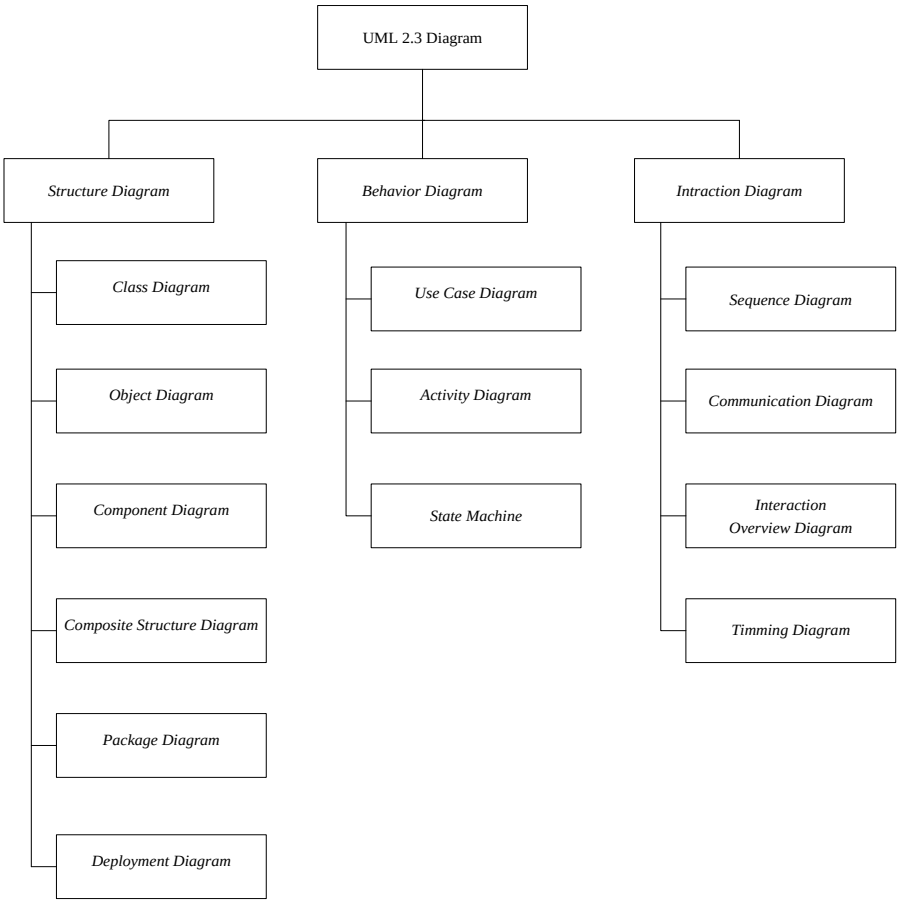
3. *Sequence Diagram*

   Menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

4. *Class Diagram*

   Menggambarkan struktur dan deskripsi class *package* (paket) pada objek beserta penghubungnya satu sama lain. Menunjukan properti dan operasi sebuah class pada batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah *class*.

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori [6]. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.5.



Gambar 2.7 *Diagram UML* [6]

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1 *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2 *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkain perubahan yang terjadi pada sebuah sistem.

3 *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

2.6 C# (CSharp)

C# (CSharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka .NET *Framework.* C# merupakan Java versi Microsoft, sebuah bahasa

multi flatform yang didesain untuk bisa berjalan diberbagai mesin. C# adalah pemrograman berorientasi Object (OOP). C# memiliki kekuatan bahasa C++ dan portabilitas seperti Java. Fitur-fitur yang diambilnya dari bahasa C++ dan Java adalah desain berorientasi objek, seperti *garbage collection, reflection,* akar kelas *(root class)*, dan juga penyederhanaan terhadap pewarisan jamak *(multiple inheritance)*. Bahasa pemrograman C# dibuat sebagai bahasa pemrograman yang bersifat *general-purpose* (untuk tujuan jamak), berorientasi objek, modern, dan sederhana [7].

2.7 Unity

Unity Game Engine adalah *software* atau *game engine* yang digunakan untuk membuat *video game* berbasis dua atau tiga dimensi dan dapat digunakan secara gratis. Selain untuk membuat *game*, *unity* juga dapat digunakan untuk membuat konten yang interaktif lainnya seperti, visual arsitektur dan *real-time* 3D animasi.

Unity adalah sebuah sebuah *tool* yang terintegrasi untuk membuat *game*, arsitektur bangunan dan simulasi. *Unity* bisa digunakan untuk *game*s PC dan *games online*. Untuk *games online* diperlukan sebuah plugin, yaitu *Unity Web Player*, yang sama halnya dengan *flash player* pada *browser*. Bahasa pemrograman yang digunakan bermacam-macam, mulai dari *javascript*, C#, dan *boo*.

Unity tidak bisa melakukan desain atau *modelling*, dikarenakan *unity* bukan merupakan *tools* untuk mendesain. Banyak hal yang bisa di lakukan di unity, ada fitur *audio reverb zone , particle effect , sky box* untuk menambahkan langit, dan masih banyak lagi, dan juga bisa langsung *edit texture* dari *editor* seperti *photoshop* dll.

Features (Scripting) didalam *unity* adalah sebagai berikut (Unity Technologies, 2013):

1. Mendukung 3 bahasa pemrograman, *JavaScript*, C#, dan *Boo*.
2. *Flexible and EasyMoving*, *rotating,* dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *duplicating, removing*, dan *changing properties.*

3. *Multi Platform Game* bisa di PC, Mac, Wii, iPhone, iPad dan *android*.
4. *Visual Properties Variables* yang di definisikan dengan *scripts* ditampilkan pada editor. Bisa digeser, di-*drag* and *drop*, bisa memilih warna dengan *color picker.*
5. Berbasis .NET, penjalanan program dilakukan dengan *Open Source* .NET *platform, Mono* [9].



Gambar 2.8 Logo Unity

2.8 Adobe Photoshop

Adobe Photoshop adalah software yang dibuat oleh perusahaan Adobe System, yang di khususkan untuk pengeditan foto atau gambar dan pembuatan effect. Perangkat lunak ini banyak digunakan oleh Fotografer Digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (market leader) untuk perangkat lunak pengolah gambar. Meskipun pada awalnya Photoshop dirancang untuk menyunting gambar untuk cetakan berbasis-kertas, Photoshop yang ada saat ini juga dapat digunakan untuk memproduksi gambar untuk *World Wide Web*. Beberapa versi terakhir juga menyertakan aplikasi tambahan, *Adobe ImageReady*, untuk keperluan tersebut.

Photoshop juga memiliki hubungan erat dengan beberapa perangkat lunak penyunting media, animasi, dan *authoring* buatan-*Adobe* lainnya. File format asli *Photoshop*, PSD, dapat diekspor ke dan dari *Adobe ImageReady*. *Adobe Illustrator, Adobe Premiere Pro, After Effects dan Adobe Encore DVD* untuk membuat DVD profesional, menyediakan penyuntingan gambar non-linear dan layanan special *effect* seperti *background*, tekstur, dan lain-lain untuk keperluan televisi, film, dan situs web [11].



Gambar 2.9 Logo Adobe Photoshop

2.9 Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses menelusuri dan mempelajari sebuah program dalam rangka menemukan kesalahan pada perangkat lunak sebelum diserahkan kepada *end user*.

2.9.1 Pengujian *Black Box*

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

Pengujian *black box* akan menemukan kesalahan dengan kategori sebagai berikut:

1. Fungsi tidak benar atau hilang
2. Kesalahan antar muka
3. Kesalahan pada struktur data (pengaksesan basis data)
4. Kesalahan inisialisasi dan akhir program
5. Kesalahan performasi

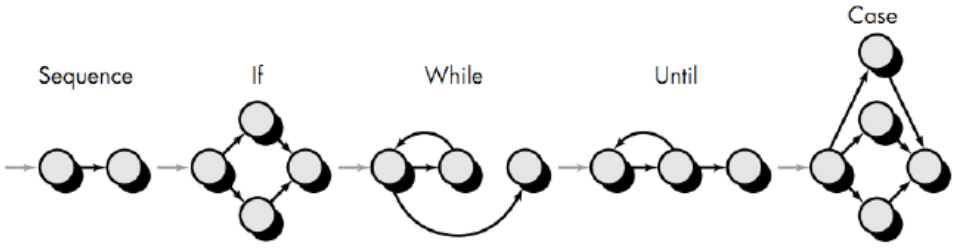
2.9.2 Pengujian *White Box*

Pengujian *white box*, yang kadang – kadang disebut pengujian *glass box*, adalah metode desain *test case* yang menggunakan struktur kontrol desain procedural untuk memperoleh *test case*. Dengan demikian metode pengujian

white box, perekayasa sistem dapat melakukan *test case* sebagai berikut:

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

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



Gambar 2.10 Gambar Notasi Diagram Alir

2.10 Skala Guttman

Skala Guttman dikembangkan oleh Louis Guttman. Skala ini mempunyai ciri penting, yaitu merupakan skala kumulatif dan mengukur satu dimensi saja dari satu variabel yang multi dimensi, sehingga skala ini termasuk mempunyai sifat undimensional. Skala Guttman yang disebut juga metode *scalogram* atau analisa skala (*scale analysis*) sangat baik untuk meyakinkan peneliti tentang kesatuan dimensi dari sikap atau sifat yang diteliti, yang sering disebut isi universal (*universe of content*) atau atribut universal (*universe attribute*).

Skala Guttman menginginkan jawaban tegas, seperti jawaban benar – salah, ya – tidak, pernah – tidak pernah, positif – negative, tinggi – rendah, baik – buruk, dan seterusnya. Pada skala guttman hanya ada dua interval, yaitu setuju dan tidak setuju. Skala Guttman dapat dibuat dalam bentuk pilihan ganda maupun daftar *checklist*. Untuk jawaban positif seperti benar, ya, tinggi, baik, dan semacamnya diberi skor 1, sedangkan untuk jawaban negative seperti salah, tidak, rendah, buruk, dan semacamnya diberi skor 0 [12].

Kelemahan pokok dari Skala Guttman, yaitu:

1. Skala ini bisa jadi tidak mungkin menjadi dasar yang efektif baik untuk mengukur sikap terhadap objek yang kompleks ataupun untuk membuat prediksi tentang perilaku objek tersebut.
2. Satu skala bisa saja mempunyai dimensi tunggal untuk satu kelompok tetapi ganda untuk kelompok lain, ataupun berdimensi satu untuk satu waktu dan mempunyai dimensi ganda untuk waktu yang lain.