BAB II
LANDASAN TEORI

II.1. Banjir

Banjir adalah tergenangnya suatu wilayah daratan yang normalnya kering dan diakibatkan oleh sejumlah hal antara lain air yang meluap yang disebabkan curah hujan yang tinggi dan semacamnya.

II.1.1. Macam-macam Banjir

Terdapat berbagai macam banjir yang disebabkan oleh beberapa hal, diantaranya:

1. Banjir Air

Banjir yang satu ini adalah banjir yang sudah umum. Penyebab banjir ini adalah meluapnya air sungai, danau, atau selokan sehingga air akan meluber lalu menggenangi daratan. Umumnya banjir seperti ini disebabkan oleh hujan yang turun terus-menerus sehingga sungai atau danau tidak mampu lagi menampung air.

2. Banjir Cileunang

Jenis banjir yang satu ini hampir sama dengan banjir air. Namun banjir cileunang ini disebakan oleh hujan yang sangat deras dengan debit air yang sangat banyak. Banjir akhirnya terjadi karena air-air hujan yang melimpah ini tidak bisa segera mengalir melalui saluran atau selokan di sekitar rumah warga. Jika banjir air dapat terjadi dalam waktu yang cukup lama, maka banjir cileunang adalah banjir dadakan (langsung terjadi saat hujan tiba).

3. Banjir Bandang

Tidak hanya banjir dengan materi air, tetapi banjir yang satu ini juga mengangkut material air berupa lumpur. Banjir seperti ini jelas lebih berbahaya daripada banjir air karena seseorang tidak akan mampu berenang ditengah-tengah banjir seperti ini untuk menyelamatkan diri. Banjir bandang mampu menghanyutkan apapun, karena itu daya rusaknya sangat tinggi. Banjir ini biasa

terjadi di area dekat pegunungan, dimana tanah pegunungan seolah longsor karena air hujan lalu ikut terbawa air ke daratan yang lebih rendah. Biasanya banjir bandang ini akan menghanyutkan sejumlah pohon-pohon hutan atau batu-batu berukuran besar. Material-material ini tentu dapat merusak pemukiman warga yang berada di wilayah sekitar pegunungan.

4. Banjir Rob (Laut Pasang)

Banjir rob adalah banjir yang disebabkan oleh pasangnya air laut. Banjir seperti ini kerap melanda kota Muara Baru di Jakarta. Air laut yang pasang ini umumnya akan menahan air sungai yang sudah menumpuk, akhirnya mampu menjebol tanggul dan menggenangi daratan.

5. Banjir Lahar Dingin

Salah satu dari macam-macam banjir adalah banjir lahar dingin. Banjir jenis ini biasanya hanya terjadi ketika erupsi gunung berapi. Erupsi ini kemudian mengeluarkan lahar dingin dari puncak gunung dan mengalir ke daratan yang ada di bawahnya. Lahar dingin ini mengakibatkan pendangkalan sungai, sehingga air sungai akan mudah meluap dan dapat meluber ke pemukiman warga.

6. Banjir Lumpur

Banjir lumpur ini identik dengan peristiwa banjir Lapindo di daerah Sidoarjo. Banjir ini mirip banjir bandang, tetapi lebih disebabkan oleh keluarnya lumpur dari dalam bumi dan menggenangi daratan. Lumpur yang keluar dari dalam bumi bukan merupakan lumpur biasa, tetapi juga mengandung bahan dan gas kimia tertentu yang berbahaya. Sampai saat ini, peristiwa banjir lumpur panas di Sidoarjo belum dapat diatasi dengan baik, malah semakin banyak titik-titik semburan baru di sekitar titik semburan lumpur utama.

II.1.2. Penyebab Terjadinya Banjir

Terdapat beberapa penyebab terjadinya banjir, diantaranya :

1. Sungai
   a. Lama

      Endapan dari hujan atau pencairan salju cepat melebihi kapasitas saluran sungai. Diakibatkan hujan deras monsun, hurikan dan depresi

tropis, angin luar dan hujan panas yang mempengaruhi salju. Rintangan drainase tidak terduga seperti tanah longsor, es, atau puing-puing dapat mengakibatkan banjir perlahan di sebelah hulu rintangan.

b. Cepat

Termasuk banjir bandang akibat curah hujan konvektif (badai petir besar) atau pelepasan mendadak endapan hulu yang terbentuk di belakang bendungan, tanah longsor, atau gletser.

Sungai-sungai yang membelah Jakarta sudah tidak lagi berfungsi maksimal dalam menampung air. Selain karena pendangkalan dan rumah-rumah penduduk yang menyemut di sepanjang pinggirannya, juga karena sungai-sungai ini penuh dengan sampah. Berbagai jenis sampah dapat ditemukan di badan sungai. Di beberapa tempat, tumpukan sampah itu begitu banyak sehingga menjadi sebuah daratan yang dapat diinjak manusia.

2. Muara

Biasanya diakibatkan oleh penggabungan pasang laut yang diakibatkan angin badai. Banjir badai akibat siklon tropis atau siklon ekstra tropis masuk dalam kategori ini.

3. Pantai

Diakibatkan badai laut besar atau bencana lain seperti tsunami atau hurikan. Banjir badai akibat siklon tropis atau siklon ekstrat ropis masuk dalam kategori ini.

4. Peristiwa Alam

Diakibatkan oleh peristiwa mendadak seperti jebolnya bendungan atau bencana lain seperti gempa bumi dan letusan gunung berapi.

5. Manusia

Kerusakan akibat aktivitas manusia, baik disengaja atau tidak merusak keseimbangan alam.

6. Lumpur

Banjir lumpur terjadi melalui penumpukan endapan di tanah pertanian. Sedimen kemudian terpisah dari endapan dan terangkut sebagai materi tetap atau penumpukan dasar sungai. Endapan lumpur mudah diketahui ketika

mulai mencapai daerah berpenghuni. Banjir lumpur adalah proses lembah bukit, dan tidak sama dengan aliran lumpur yang diakibatkan pergerakan massal.

7. Lainnya
    a. Banjir dapat terjadi ketika air meluap di permukaan kedap air (misalnya akibat hujan) dan tidak dapat terserap dengan cepat (orientasi lemah atau penguapan rendah).
    b. Rangkaian badai yang bergerak ke daerah yang sama.
    Berang-berang pembangun bendungan dapat membanjiri wilayah perkotaan dan pedesaan rendah, umumnya mengakibatkan kerusakan besar.

II.1.3. Dampak yang ditimbulkan oleh Banjir

1. Primer

Kerusakan fisik: Mampu merusak berbagai jenis struktur, termasuk jembatan, mobil, bangunan, sistem selokan bawah tanah, jalan raya, dan kanal.

2. Sekunder
    a. Persediaan air: Kontaminasi air. Air minum bersih mulai langka.
    b. Penyakit: Kondisi tidak higienis. Penyebaran penyakit bawaan air.
    c. Pertanian dan persediaan makanan: Kelangkaan hasil tani disebabkan oleh kegagalan panen. Namun, dataran rendah dekat sungai bergantung kepada endapan sungai akibat banjir demi menambah mineral tanah setempat.
    d. Pepohonan: Spesies yang tidak sanggup akan mati karena tidak bisa bernapas.
    e. Transportasi: Jalur transportasi rusak, sulit mengirimkan bantuan darurat kepada orang-orang yang membutuhkan.
3. Dampak Tersier/Jangka Panjang
    a. Ekonomi: Kesulitan ekonomi karena kerusakan pemukiman yang terjadi akibat banjir, dalam sector pariwisata menurunnya minat wiasatawan, biaya pembangunan kembali, kelangkaan makanan yang mendorong kenaikan harga, dll.

Dari berbagai dampak negatif yang ditimbulkan, ternyata banjir (banjir air skala kecil) juga dapat membawa banyak keuntungan, seperti mengisi kembali air tanah, menyuburkan serta memberikan nutrisi kepada tanah. Air banjir menyediakan air yang cukup di kawasan kering dan semi-kering yang curah hujannya tidak menentu sepanjang tahun. Air banjir tawar memainkan peran penting dalam menyeimbangkan ekosistem di koridor sungai dan merupakan faktor utama dalam penyeimbangan keragaman makhluk hidup di dataran. Banjir menambahkan banyak nutrisi untuk danau dan sungai yang semakin memajukan industri perikanan pada tahun-tahun mendatang, selain itu juga karena kecocokan dataran banjir untuk pengembangbiakan ikan (sedikit predasi dan banyak nutrisi).

II.2. *Game*

Game merupakan suatu permainan yang dapat dilakukan baik secara individu maupun berkelompok untuk dapat menyelesaikn misi yang diberikan pada *game* tersebut. *Game* biasanya digunakan sebagai media untuk mengisi waktu luang, kesenangan dan pendidian.

Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana *game* itu menarik untuk dimainkan secara maksimal. Pada awalnya, *game* indentik dengan permainan anak-anak. Kita 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 sebagi *game* [5].

II.2.1. Klasifikasi *Game*

Berikut ini adalah klasifikasi *game* berdasarkan jenis *platform* yang digunakan dan juga berdasarkan *genre* permainannya.

II.2.1.1. Berdasarkan Jenis *Platform* yang digunakan

1. *Arcade games*, yaitu yang sering disebut *ding-dong* di Indonesia, biasanya berada di daerah/tempat khusus dan memiliki box atau mesin yang

memang khusus di design untuk jenis video games tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati , seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya)

2. *PC Games*, yaitu video *game* yang dimainkan menggunakan *Personal Computer*.
3. *Console games*, yaitu *video games* yang dimainkan menggunakan *console* tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
4. *Handheld games*, yaitu yang dimainkan di *console* khusus *video game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
5. *Mobile games*, yaitu yang dapat dimainkan atau khusus untuk *mobile phone* atau PDA.

II.2.1.2. Berdasarkan *Genre* permainannya

1. Aksi *Shooting*, *video game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari *game* jenis ini adalah tembak, tembak dan tembak. Termasuk didalam-nya:
   a. *First person shooting (FPS)* seperti *Counter Strike* dan *Call of Duty.*
   b. *Drive n shoot,* menggunakan unsur simulasi kendaraan tetapi tetap dengan tujuan utama menembak dan menghancurkan lawan, contoh : *Spy Hunter, Rock and Roll Racing, Road Rash.*
   c. *Shoot em up,* seperti Raiden, 1942, dan gradius.
   d. *Beat em up* (tonjok hajar) seperti *Double Dragon* dan *Final Fight*, lalu *hack and slash* (tusuk tebas) seperti *Shinobi dan Legend of Kage*.
   Light gun shooting, yang menggunakan alat yang umumnya berbentuk seperti senjata, seperti *Virtua Cop* dan *Time Crisis*.
2. *Fighting* (pertarungan) Ada yang mengelompokan *video game* fighting dibagian Aksi, namun penulis berpendapat berbeda, jenis ini memang memerlukan kecepatan refleks dan koordinasi mata-tangan, tetapi inti dari game ini adalah penguasaan jurus (hafal caranya dan lancar

mengeksekusinya), pengenalan karakter dan timing sangatlah penting, iya, combo-pun menjadi esensial untuk mengalahkan lawan secepat mungkin. Dan berbeda seperti *game* Aksi pada umumnya yang umumnya hanya melawan *Artificial Intellegence* atau istilah umumnya melawan komputer saja, pemain jenis *fighting game* ini baru teruji kemampuan sesungguhnya dengan melawan pemain lainnya. Seri *Street Fighter, Tekken, Mortal Kombat, Soul Calibur* dan *King of Fighter* adalah contohnya [2].

3. Aksi Petualangan. Memasuki gua bawah tanah, melompati bebatuan diantara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris, atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam *video game* jenis ini. Menurut penulis, *game* jenis ini sudah berkembang jauh hingga menjadi *genre* campuran *action beat-em up* juga, dan sekarang, di tahun 2000 an, jenis ini cenderung untuk memilik visual 3D dan sudut pandang orang ke-tiga. *Tomb Rider, Grand Theft Auto* dan *Prince of Persia* termasuk didalamnya [2].
4. Petualangan. Bedanya dengan jenis video game aksi-petualangan, refleks dan kelihaian pemain dalam bergerak, berlari, melompat hingga memecut atau menembak tidak diperlukan di sini. *Video Game* murni petualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat. Termasuk didalamnya:
    a. Petualangan dengan teks atau sistem tunjuk dan klik, contoh: *Kings Quest, Space Quest, Heroes Quest, Monkey Island*, *Sam and Max*.
    b. Novel atau film interaktif, seperti game dating yang banyak beredar di *jepang, Dragons Lair* dan *Night Trap*.
5. Simulasi, Konstruksi dan manajemen. *Video Game* jenis ini sering kali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata

dan memperhatikan dengan detil berbagai faktor. *Video Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas. Contoh: *Sim City, The Sims, Tamagotchi* [2].

6. *Role Playing*. *Video game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya level, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun mahluk peliharaan.Secara kebudayaan, pengembang game Jepang biasanya membuat *Role Playing Game (RPG)* ke arah cerita linear yang diarahkan seolah karakter kita adalah tokoh dalam cerita itu, seperti *Final Fantasy*, *Dragon Quest* dan *Xenogears*. Sedangkan pengembang *game RPG* Eropa, cenderung membuat karakter kita bebas memilih jalan cerita sendiri secara non-linear, seperti *Ultima, Never Winter Nights, baldurs gate, Elder Scroll*, dan *Fallout* [2].
7. Strategi. Kebalikan dari *video game* jenis *action* yang berjalan cepat dan perlu refleks secepat kilat, *video game* jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. *Video game* strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusal pelatihan tempur, tergantung dari tema ceritanya. Pemain *game* strategi melihat dari sudut pandang lebih meluas dan lebih kedepan dengan waktu permainan yang biasanya lebih lama dan santai dibandingkan *game action*. Unsur-unsur permainannya biasanya berkisar sekitar, prioritas pembangunan, peletakan pasukan, mencari dan

memanfaatkan sumberdaya (uang, besi, kayu, minyak,dll), hingga ke pembelian dan pengembangan pasukan atau teknologi. *Game* jenis ini terbagi atas:

a. *Real Time Strategy*

   Game berjalan dalam waktu sebenarnya dan serentak antara semua pihak dan pemain harus memutuskan setiap langkah yang diambil saat itu juga berbarengan mungkin saat itu pihak lawan juga sedang mengeksekusi strateginya. Contoh: *Starcraft, Warcraft*, dan *Command and Conquer*.

b. *Turn Based Strategy*

   Game yang berjalan secara bergiliran, saat kita mengambil keputusan dan menggerakan pasukan, saat itu pihak lawan menunggu, begitu pula sebaliknya, layaknya catur. Contoh: *Front Mission, Super robot wars, Final Fantasy tactics, Heroes of might and magic, Master of Orion*. Sebenarnya ada yang memilah lagi menjadi jenis *tactical* dan strategi, namun penulis cenderung untuk menggabungkannya karena perbedaannya hanya ada di masalah skala dan kerumitan dalam manajemen sumber daya saja [2].

8. *Puzzle. Video game* jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, itu semua termasuk dalam jenis ini. Sering pula permainan jenis ini adalah juga unsur permainan dalam *video game* petualangan maupun *game* edukasi. *Tetris, Minesweeper, Bejeweled, Sokoban* dan *Bomberman*.
9. Simulasi Kendaraan. *Video Game* jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, muskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut. Terbagi atas beberapa jenis:

   a. Perang

Video game simulasi kendaraan yang sempat tenar di tahun 90-an ini mengajak pemain untuk menaiki kendaraan dan berperang melawan kendaraan lainnya. Dan kebanyakan diantaranya memiliki judul sama dengan nama kendaraannya. Contoh : Apache *64, Comanche, Abrams, YF-23, F-16 fighting eagle* [2].

b. Balapan

Dari namanya sudah jelas, siapa sampai duluan di garis finish dialah pemenangnya! Terkadang malah pemain dapat memilih kendaraan, mendandani, *upgrade* mesin bahkan mengecatnya. Contoh: *Top Gear, Test Drive, Sega Rally Championship, Daytona, Grand Turismo, Need For Speed, Mario Cart, ManXTT.*

c. Luar Angkasa

Walau masih dapat dikategorikan simulasi kendaraan perang, tetapi segala unsur fiksi ilmiah dan banyaknya judul yang beredar membuat subgenre ini pantas dikategorikan diluar simulasi kendaraan perang. Jenis ini memungkinkan pemain untuk menjelajah luar angkasa, berperang dengan mahluk alien, mendarat di planet antah berantah atau sekedar ingin merasakan bagaimana menjadi kapten di film fiksi ilmiah kesayangan kamu. Contoh : *Wing Commander, Freelancer , Star Wars X-Wing, Star Wars Tie Fighter,* dl [2].

d. *Mecha*

Pendapat bahwa hampir tidak ada orang yang terekspos oleh film robot jepang saat kecilnya tidak memimpikan ingin mengendalikan robot, memang sulit dibantah. Dipopulerkan oleh serial *Mechwarrior* oleh Activision, subgenre Simulasi *Mecha* ini memungkinkan pemainnya untuk mengendalikan robot dan menggunakannya untuk menghancurkan gedung, helikopter dan tentu saja robot lainnya. Contoh : *Mechwarrior, Gundam Last war Chronicles,* dan *Armored Core.*

10. Olahraga. Singkat, padat, jelas, bermain sport di PC atau konsol anda. Biasanya permainannya diusahakan serealistik mungkin walau kadang ada

yang menambah unsur fiksi seperti *NBA JAM*. Contohnya pun jelas, Seri *Winning Eleven, seri NBA,* seri *FIFA, John Madden NFL, Lakers vs Celtics, Tony hawk pro skater*, dll [2].

II.2.2. Sudut Pandang *Game*

Sebuah *game* biasanya mempunyai sudut pandang permainan tersendiri disesuaikan berdasarkan genre game yang diambil. Berikut beberapa macam sudut pandang permainan yang biasa digunakan :

1. *Side Scrolling*

   Side Scrolling adalah sudut pandang permainan yang terlihat dari samping dan memungkinkan karakter utama untuk bergerak dari kiri ke kanan serta memungkinkan background pada game seolah-olah bergeser mengikuti pergerakan karakter utama.

2. *Top Down*

   Top Down adalah sudut pandang permainan yang memungkinkan karakter utama bermanuver ke empat arah namun cara permainannya sendiri bergeser dari bawah ke atas, dan biasanya game yang menggunakan sudut pandang permainan jenis ini adalah shooter game.

3. *Isometric*

   Isometric adalah sudut pandang permainan yang memungkinkan permainan terlihat diantara sisi *Side Scrolling* dan juga *Top Down*, dan biasanya diterapkan pada game dengan genre RTS.

II.3. Kecerdasan Buatan

Kecerdasan buatan merupakan sub-bidang ilmu komputer (*computer science*) yang khusus ditujukan untuk membuat perangkat lunak dan perangkat keras yang sepenuhnya bisa menirukan beberapa fungsi otak manusia, atau cabang ilmu komputer yang mempelajari otomatisasi tingkah laku cerdas (*intelegent*) [9].

Kecerdasan harus didasarkan pada prinsip-prinsip teoritikal dan terapan yang menyangkut [9]:

1. Struktur data yang digunakan dalam representasi pengetahuan (*knowledge representation*).
2. Algoritrma yang diperlukan dalam penerapan pengetahuan.
3. Teknik-teknik bahasa dan pemrograman yang dipakai dalam implementasinya.

Kecerdasan buatan menawarkan baik media maupun uji teori kecerdasan. Teori-teori semacam ini dapat dinyatakan dalam bahasa program komputer dan dibuktikan melalui eksekusinya pada komputer.

Bidang-bidang teknik kecerdasan buatan diantaranya adalah [9]:

1. Sistem pakar (*expert system*).
2. Robot (*robotic*).
3. Logika samar (*fuzzy logic*).
4. Jaringan saraf (*neural networks*) tiruan.
5. Pengolahan bahasa alami (*natural language processing*).
6. Pengolahan citra.

II.3.1. Agen Cerdas

Agen (*agent*) adalah sesuatu yang dapat melihat, merasakan (perceiving), memahami, dan mengerti lingkungannya melalui detektor (*detektor*) / sensor (*censor*) dan bertindak (*acting*) atas lingkungan itu pula melalui media sarananya (*effector*).

Manusia mempunyai agen berupa mata, dan telinga yang berupa detektor/sensor, sedangkan tangan, kaki, dan mulut merupakan efektornya. Robot mempunyai kamera dan detektor infra merah sebagai sensor, dan motor-motor listrik, lengan mekanik sebagai efektornya.

Tugas dari kecerdasan buatan adalah merancang program agen (*agent program*), yaitu suatu fungsi yang mengimplementasikan peta agen dari persepsi hingga aksi. Sedangkan program yang dijalankan tersebut memerlukan sebuah peralatan tertentu yang dapat berupa komputer atau perangkat keras khusus untuk menangani tugas tertentu yang dikenal sebagai arsitektur (*architecture*).

II.3.2. Moving Target Search

a. Pengertian *Moving Target Search*

Algoritma *Moving Target Search* didefinisikan sebagai salah satu algoritma pemecahan masalah yang berbasis pencarian terhadap target dan juga memiliki jangkauan deteksi atau daerah dimana dapat mendeteksi target [1].

b. Pergerakan *Moving Target Search*

Ada dua peristiwa berbeda yang terjadi pada algoritma *MTS* yang pertama adalah pergerakan dari tim penyelamat dan pergerakan dari target yang masing-masing bisa disertai dengan memperbarui nilai heuristik. Tim penyelamat dan target bergerak secara bergantian, sampai posisi dari tim penyelamat dan target menduduki posisi yang sama. Algoritma *MTS* ini dijelaskan lebih detail seperti di bawah ini [6]:

1. Pada saat target diam, maka tim penyelamat bergerak dari titik awal ke tujuan selanjutnya dengan menggunakan perhitungan sebagai berikut :

$$F(x',y) = H(x',y) + 1 \quad \text{… Persamaan 2.1}$$

Dimana :

a. F(x’,y) adalah fungsi evaluasi (jumlah Heuristik + 1). F(x’,y) merupakan merupakan sebuah nama variable satu kesatuan bukan terpisah, x’ dan y terbungkus atau tergabung dalam nilai F.
b. H(x’,y) adalah perkiraan nilai pergerakan simpul awal menuju tujuan selanjutnya. Fungsi ini seringkali disebut dengan fungsi heuristik, dinamakan heuristik karena perhitungan tersebut berdasarkan perkiraan.
c. X adalah nama variabel dari tim penyelamat. Posisi awal dari X sudah ditentukan sedangkan untuk X’ adalah perpindahan posisi dari posisi awal ke tejuan selanjutnya.
d. Y adalah nama varibel dari korban. Posisi awal dari Y sudah ditentukan sedangkan untuk Y’ adalah perpindahan posisi dari posisi awal ke tujuan selanjutnya.

2. Ketika target mulai bergerak dari titik awal ke tetangganya, maka tim penyelamat memperhatikan pergerakannya, kemudian akan selalu memperbaharui nilai perpindahan dari korban, dengan menggunakan perhitungan sebagai berikut :

$$F(x,y') = H(x,y') - 1 \quad \text{… Persamaan 2.2}$$

Dimana :

a. F(x,y') adalah fungsi evaluasi (jumlah Heuristik + 1). F(x,y') merupakan merupakan sebuah nama variable satu kesatuan bukan terpisah, x dan y' terbungkus atau tergabung dalam nilai F.
b. H(x,y') adalah perkiraan nilai pergerakan simpul awal menuju tujuan selanjutnya. Fungsi ini seringkali disebut dengan fungsi heuristik, dinamakan heuristik karena perhitungan tersebut berdasarkan perkiraan.
c. X adalah nama variabel dari tim penyelamat. Posisi awal dari X sudah ditentukan sedangkan untuk X' adalah perpindahan posisi dari posisi awal ke tejuan selanjutnya.
d. Y adalah nama varibel dari korban. Posisi awal dari Y sudah ditentukan sedangkan untuk Y' adalah perpindahan posisi dari posisi awal ke tujuan selanjutnya.

c. Fungsi Heuristik

Sama seperti kebanyakan algoritma pencarian lainnya, kinerja algoritma *MTS* bergantung pada fungsi heuristik yang digunakan, game ini menggunakan fungsi heuristic Non-Manhattan Distance, dengan fungsi heuristic yang digunakan adalah sebagai berikut [7]:

$$h_diagonal(n) = \min(abs(n.x - goal.x) + abs(n.y - goal.y))$$

$$h_orthogonal(n) = (abs(n.x - goal.x) + abs(n.y - goal.y))$$

$$h(n) = h_diagonal(n) + (h_orthogonal(n) - (2 * h_diagonal(n)))$$

… Persamaan 2.3

Dimana :

a. h_diagonal(n) adalah banyaknya langkah diagonal yang bisa diambil untuk mencapai goal dari *node* n.
b. h_orthogonal(n) adalah banyaknya langkah lurus yang bisa diambil untuk mencapai goal dari *node* n.
c. n.X adalah koordinat X tetangga dari tim penyelamat.
d. n.Y adalah koordinat Y tetangga dari tim penyelamat.
e. goal.X adalah koordinat X dari target.
f. goal.Y adalah koordinat Y dari target.

II.4. Pemrograman Berorientasi Objek

OOP (*Object Oriented Programing*) adalah sebuah pendekatan untuk pengembangan suatu *software*, dimana dalam struktur *software* tersebut didasarkan kepada interaksi objek dalam penyelesaian suatu proses atau tugas.

Interaksi tersebut mengambil *form* dari pesan-pesan dan mengirimkannya kembali antar objek tersebut. *Object* akan merespon pesan tersebut menjadi sebuah tindakan (*action*) atau metode.

Konsep *OOP* dimulai pada pertengahan 1960-an dengan bahasa program SIMULA, kemudian dilanjutkan pada era 70-an dengan SMALLTALK. Meskipun *developer software* tidak secara intensif mengembangkan *OOP*, tetapi metodologi *Object Oriented* tetap digunakan.

Konsep dasar umum yang biasanya terdapat pada semua bahasa pemrograman yang mendukung semua bahasa pemrograman yang mendukung *OOP* [4]:

1. Objek

   Objek merupakan segala sesuatu yang dapat dibedakan satu sama lainnya. Segala sesuatu yang ada dialam semesta ini adalah objek.

   Contohnya manusia, mobil, hewan, tumbuhan, tempat, atau bahkan yang tidak bersifat fisik seperti kejadian atau konsep-konsep. Sehingga bisa disimpulkan bahwa objek tidak harus bersifat fisik, karena jika dikaitkan dengan *OOP*, objek akan menjadi bentuk logis.

Dalam bahasa teoritis *OOP*, objek berfungsi untuk membungkus data dan fungsi bersama menjadi satu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer yang berorientasi pada objek. Setiap objek memiliki dua ciri khas, yaitu Atribut atau Properti (data merupakan cirri-ciri objek) dan Method atau behavior.

2. *Class*

   Definisi *Class* yaitu template untuk membuat objek. *Class* merupakan prototipe atau *blue print* yang mendefinisikan variabel-variabel dan *method-method* secara umum. Objek merupakan hasil dari *instances* dari suatu *class.* Proses pembentukan objek dari suatu *class* disebut sebagai *instances.* Objek disebut juga sebagai *instances*. Dalam bahasa *OOP*, *class* merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.

3. Properti

   Properti atau atribut adalah data yang membedakan antara objek satu dengan yang lainnya. Atribut dibedakan menjadi dua jenis, yaitu *Instance Variable* dan *Class Variable.*

   Instance Variable adalah atribut untuk setiap objek dari *class* yang sama. Setiap objek mempunyai dan menyimpan nilai atributnya sendiri. Jadi, setiap objek dari *class* yang sama boleh mempunyai nilai yang sama atau berbeda.

   Class Variable adalah atribut untuk semua objek yang dibuat dari *class* yang sama. Semua objek mempunyai nilai atribut yang sama. Jadi, semua objek dari *class* yang sama mempunyai hanya satu nilai yang isinya sama.

4. *Method*

   Behavior atau tingkah laku adalah hal-hal yang bisa dilakukan oleh objek dari suatu *class.* Behavior dapat digunakan untuk mengubah nilai atribut suatu objek, menerima informasi dari objek lain, dan mengirim informasi ke objek lain untuk melakukan suatu tugas (*task*). Dalam *class*, behavior disebut juga sebagai *method. Method* sendiri adalah serangkaian pernyataan dalam

suatu *class* yang menghandle suatu *task* tertentu. Cara objek berkomunikasi dengan objek yang lain dengan menggunakan *method.*

5. *Encapsulation* (Pembungkusan)

   Salah satu cirri penting *OOP* adalah *encapsulaption.* Definisi *encapsulaption* secara teoritis adalah pembungkusan variabel dan *method* dalam sebuah objek yang terlindungi serta menyediakan *interface* untuk mengakses variabel tersebut. Variabel dan *method* yang dimiliki oleh suatu objek, bisa ditentukan hak aksesnya.

   Dengan kata lain, enkapsulasi berfungsi untuk memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam atau dari sebuah objek dengan cara yang tidak layak hanya *method* dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

6. *Inheritance* (Pewarisan)

   Inheritance merupakan pewarisan atribut dan *method* dari sebuah *class* ke *class* lainnya. *Class* yang mewarisi disebut *superclass* dan *class* yang diwarisi disebut *subclass. Subclass* bisa berlaku sebagai *superclass* bagi *class* lainnya, disebut sebagai *multilevel inheritance*.

   Prinsip dasar *inheritance* yaitu persamaan-persamaan yang dimiliki oleh beberapa *class* dapat digabungkan dalam sebuah *class* induk sehingga setiap kelas diturunkannya memuat hal-hal yang spesifik untuk *class* yang bersangkutan. Secara singkat bisa diartikan teknik yang menyatakan bahwa anak dari objek akan mewarisi data atau atribut dan metode dari induknya langsung.

7. Polimorfisme

   Polimorfisem adalah kemampuan suatu objek untuk mempunyai lebih dari satu bentuk. *Polimorfisme* tidak bergantung kepada pemanggilan subrutin. Metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu dimana pesan tersebut dikirim.

Polimorfisme bisa juga diartikan sebagai aksi yang sama yang dapat dilakukan terhadap beberapa objek. *Polimorfisme* berarti bahwa operasi yang sama mungkin mempunyai perbedaan dalam *class* yang berbeda.

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

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek. Berikut merupakan macam-macam diagram UML [8]:

1. *Use Case Diagram*
   a. Menggambarkan bagaimana sistem digunakan
   b. Mendeskripsikan apa yang sistem lakukan tanpa mendeskripsikan bagaimana sistem menyelesaikannya.
   c. Dibuat berdasarkan interaksi dan relasi dari individual use case.
   d. Berisi aktor, event, dan use case.
   e. Titik awal untuk pemodelan UML.
2. *Use Case Scenario*

   Sebuah artikulasi verbal pengecualian terhadap perilaku utama yang dijelaskan oleh kasus penggunaan utama. Use case skenario merupakan hasil instansiasi dari setiap use case. Terbagi menjadi tiga bagian, yaitu:
   a. Identifikasi dan inisiasi.
   b. Step perfomed.
   c. Kondisi, asumsi dan pertanyaan.
3. *Activity Diagram*

   Menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.
4. *Sequence Diagrams*

   Menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainnya) berupa *message* yang

digambarkan terhadap waktu. *Sequence diagrams* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

5. *Class Diagram*

   Menggambarkan keadaan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem.

II.6. Adobe Flash CS5

Flash merupakan program atau software yang ditujukan untuk membuat animasi, desain, web, aplikasi, game, dan sebagaianya. Flash lebih dikenal khalayak ramai sebagai software pembuat animasi. Seiring perkembangan zaman yang semakin maju dan pesat dalam bidang teknologi khususnya dalam bidang teknologi informasi, flash sampai sekarang telah mencapai versi 11 atau lebih dikenal dengan Adobe Flash CS5 Profesional yang dirilis oleh perusahaan software ternama di Amerika Serikat, yaitu Adobe System Incorporated.

Flash CS5 yang sekarang banyak dikenal dengan kemampuan yang andal dalam membuat animasi-animasi, tetapi tidak jarang yang mengenal sisi lain dari kelebihan Flash ini. Salah satunya adalah mempunyai bahasa pemrograman yang sangat ampuh dalam segala hal, yaitu Action Script yang sekarang sudah dirilis samapai ke versi 3.0 yang dimiliki oleh Flash CS3,CS4, dan CS5.

Adobe Flash CS5 menawarkan beberapa pembaruan yang komplet dengan penawaran penggunaan lebih menyenangkan. Fasilitas seperti 3D Effect atau transformations yang sangat menarik dapat digunakan untuk membuat efek-efek animasi 3 dimensi yang menarik. Berbagai fasilitas dan fitur terbaru telah disediakan untuk kemudahan dalam pengolahan para penggunanya. Adobe Flash CS5 merupakan penyempurnaan dari versi sebelumnya (CS4) [5].

Beberapa komponen kerja program Adobe Flash CS5 telah ditampilkan sebagai tampilan standart. Masih banyak lagi komponen yang masih tersembunyi sehingga memerlupan perintah tertentu untuk menampilkannya. Berikut ini adalah beberapa komponen kerja dari Adobe Flash CS5 [5]:

1. *Toolbox.*
2. *Timeline*

Timeline berguna untuk menentukan durasi animasi, jumlah layer, frame, menempatkan script dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang Anda buat akan diatur dan ditempatkan pada layer dalam timeline.

3. *Output.*
4. *Compiler errors.*
5. *Motion Editor*

   Motion editor berguna untuk melakukan control animasi yang telah dibuat seperti : mengatur motion, transformasi, pewarnaan, filter dan parameter animasi lainnya.
6. *Stage*

   Stage adalah lembar kerja yang digunakan untuk membuat atau mendesain objek yang akan dianimasikan. Objek yang dibuat dalam lembar kerja dapat berupa objek vector, movie clip, text, button dan lain-lain.
7. *Properties, Library.*
8. *Color, Swatches.*
9. *Align, Info, Transform.*