BAB 2
TINJAUAN PUSTAKA

Tinjauan pustaka memberikan gambaran dari teori yang terkait dengan pembangunan sistem. Teori pokok pada tinjauan pustaka yang terdapat di dalam skripsi ini adalah sebagai berikut :

2.1. Jaringan Komputer

Jaringan komputer adalah sekumpulan peralatan komputer *(hardware dan software)* yang dihubungkan agar dapat saling berkomunikasi dengan tujuan komunikasi dan berbagi sumber daya.

2.1.1. Sejarah Jaringan Komputer

Sejarah jaringan komputer berawal dari time-sharing networks, yaitu ‘rangkaian’ terminal yang terhubung dengan komputer sentral yang disebut mainframe. Contoh time-sharing networks adalah IBM’s System Network Architecture (SNA) dan Digital Network Architecture [1].

Tujuan dari jaringan komputer adalah untuk melakukan komunikasi data, sharing data maupun pemakaian resource bersama seperti printer dan media penyimpanan sekunder. Komunikasi data sendiri memiliki tujuan yang lebih khusus [12], yaitu:

1. Memungkinkan pengiriman data dalam jumlah besar, efisien, ekonomis, dan tanpa kesalahan dari suatu tempat ke tempat yang lain.
2. Memungkinkan penggunaan sistem komputer dan peralatan pendukung dari jarak jauh (remote).
3. Memungkinkan penggunaan komputer secara terpusat maupun secara tersebar sehingga mendukung manajemen dalam hal kontrol, baik desentralisasi ataupun sentralisasi.
4. Mempermudah kemungkinan pengelolaan dan pengaturan data yang ada dalam berbagai macam sistem komputer.
5. Mengurangi waktu untuk pengelolaan data.
6. Mendapatkan data langsung dari sumbernya.
7. Mempercepat penyebarluasan informasi.

8. Komunikasi data berkaitan dengan pertukaran data diantara dua perangkat yang terhubung secara langsung yang memungkinkan adanya pertukaran data antar kedua pihak.

2.1.2. Topologi Jaringan

Topologi jaringan merupakan suatu aturan atau cara untuk menghubungkan komputer yang satu dengan komputer yang lainnya sehingga membentuk suatu jaringan. Topologi jaringan dapat didefinisikan juga sebagai gambaran secara fisik dari pola hubungan antar komponen jaringan. [12]

Ada beberapa jenis topologi jaringan yang dikenal secara luas yang memiliki kelebihan dan kekurangan tersendiri. Topologi jaringan juga sangat berpengaruh terhadap performasi maupun kompleksitas jaringan yang dibangun, oleh karena itu dalam pemilihan topologi jaringan harus disesuaikan dengan kebutuhan terhadap kesesuaian infrastruktur jaringan yang akan dibangun.

2.1.3. Konsep Topologi Jaringan

Dalam membangun sebuah topologi jaringan tidak selalu perangkat-perangkat yang terhubung mewakili cara akses dari setiap perangkat jaringan yang terhubung itu. Cara akses dan keterhubungan antar komponen jaringan dapat dipandang sebagai dua jenis topologi yang berbeda yang dilihat secara fisik dan logic.

Topologi fisik jaringan adalah cara yang digunakan untuk menghubungkan workstation-workstation di dalam LAN [13] atau dapat disebut sebagai suatu desain posisi dan letak node yang dilihat secara nyata untuk menentukan keterhubungan antar komponen jaringan melalui kabel atau media lainnya.

Sementara topologi logic jaringan adalah topologi yang dilihat dari metode accessnya [13] atau dapat disebut sebagai desain jaringan komputer untuk mentransmisikan data antar komputer yang dilihat dari cara kerjanya.

Sehingga dapat disimpulkan bahwa topologi jaringan jika dilihat dari keterhubungan antar komponen jaringan dan cara transmisi data yang berlangsung antar komputer dapat dibedakan sebagai topologi fisik jaringan dan topologi logic jaringan.

2.2. Internet

Internet adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut. Internet awalnya merupakan rencana dari

Departemen Pertahanan Amerika Serikat *(US Departement Of Defense)* pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau *Advanced Research Project Agency Network.* Beberapa universitas di Amerika Serikat diantaranya UCLA, Stanford, UC Santa Barbara dan University of Utah, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat komputer di lokasi universitas yang berbeda tersebut. Perkembangan ARPANET ini cukup pesat jika dilihat dari perkembangan komputer pada saat itu. Karena perkembanganya sangat pesat, jaringan komputer ini tidak dapat lagi disebut sebagai APRANET karena semakin banyak komputer dan jaringan-jaringan regional yang terhubung. Konsep ini kemudian berkembang dan dikenal sebagai konsep Internetworking (Jaringan antar jaringan). Oleh karena itu istilah internet menjadi semakin popular, dan orang menyebutnya jaringan besar komputer tersebut dengan istilah internet.

2.2.1. Transmision Control Protocol / Internet Protocol (TCP/IP)

TCP/IP adalah sekumpulan protokol yang terdapat di dalam jaringan komputer (*network*) yang digunakan untuk berkomunikasi atau bertukar data antar komputer. TCP/IP merupakan protokol standar pada jaringan internet yang menghubungkan banyak komputer yang berbeda jenis mesin maupun sistem operasi agar dapat berinteraksi satu sama lain [14].

Konsep TCP/IP berawal dari kebutuhan DoD (*Departement of Defense*) USA akan suatu komunikasi di antara berbagai variasi komputer yang telah ada. Komputer-komputer DoD ini seringkali harus menghubungkan antara satu organisasi peneliti dengan organisasi peneliti lainnya. Komputer tersebut harus tetap berhubungan karena terkait dengan pertahanan negara dan sumber informasi harus tetap berjalan meskipun terjadi bencana alam besar, seperti ledakan nuklir. Oleh karenanya pada tahun 1969 dimulailah penelitian terhadap serangkaian protokol TCP/IP [14].

Adapun tujuan penelitian tersebut adalah sebagai berikut:

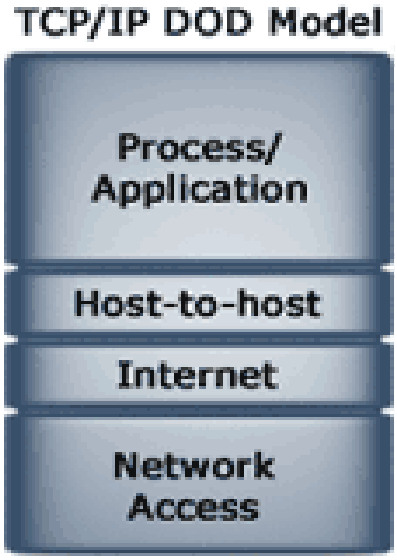
1. Terciptanya protokol-protokol umum, (DoD memerlukan suatu protokol yang dapat dipergunakan untuk semua jenis jaringan).
2. Meningkatkan efisiensi komunikasi data.
3. Dapat dipadukan dengan teknologi WAN (Wide Area Network) yang telah ada.

4. Mudah dikonfigurasi.

Tahun 1968 DoD ARPAnet (Advanced Research Project Agency) memulai penelitian yang kemudian menjadi cikal bakal packet switching. Packet switching inilah yang memungkinkan komunikasi antara lapisan network, dimana data dijalankan dan disalurkan melalui jaringan dalam bentuk unit-unit kecil yang disebut packet. Tiap-tiap paket ini membawa informasi alamat masing-masing yang ditangani dengan khusus oleh jaringan tersebut dan tidak tergantung dengan paket-paket lain. Jaringan yang dikembangkan ini adalah awal mula jaringan yang memiliki cakupan yang luas dan menjadi terkenal sebagi internet [14].

2.2.2. Arsitektur TCP/IP

Arsitektur rangkaian protokol TCP/IP didefinisikan dengan berbagai cara agar fungsi protokol-protokol TCP/IP tersebut dapat saling menyesuaikan. Protokol TCP/IP merupakan protokol standar yang terdapat pada Referensi Model DoD atau TCP/IP maupun Refernesi Model OSI. Jika model OSI memiliki 7 lapisan layer maka pada model TCP/IP memiliki 4 lapisan layer yang diperlihatkan sebagai berikut:



Gambar 2. 1 TCP/IP Model

Fungsi dari setiap layer pada TCP/IP model adalah sebagai berikut:

1. Layer Application

Layer Application merupakan layer teratar dalam model TCP/IP layer ini bertanggung jawab untuk menyediakan akses kepada apliaksi terhadap layanan jaringan TCP/IP.

2. Layer Host-to-Host atau layer Transport

Bertanggung jawab untuk komunikasi antar aplikasi dengan melakukan sesi koneksi yang bersifat connection-oriented (TCP) dan connectionless (UDP)

3. Layer Internet

Layer internet ini berfungsi untuk melakukan pemetaan (routing) dan enkapsulasi packet. Salah satu protokol yang terlibat didalamnya adalah Internet Protocol (IP)

4. Layer Network Access

Berfungsi untuk meletekkan frame-frame jaringan di atas media jaringan yang digunakan.

2.2.3. TCP Dan IP

TCP dan IP merupakan protokol yang bekerja pada suatu layer dan menjadi penghubung antara satu komputer dengan komputer yang lain di dalam network, meskipun kedua komputer tersebut memiliki OS yang berbeda. Dalam melakukan komunikasi didalam jaringan, TCP dan IP memiliki tugas masing-masing yaitu sebagai berikut:

1. TCP, Transmission Control Protocol merupakan connection-oriented, yang berarti bahwa kedua komputer yang ikut serta dalam pertukaran data harus melakukan hubungan terlebih dulu sebelum pertukaran data berlangsung. Selain itu TCP juga bertagnggungjawab untuk meyakinkan bahwa data yang dikirimkan sampai ke tujuan, memeriksa kesalahan dan mengirimkan error ke lapisan atas jika TCP tidak berhasil melakukan hubungan. Jika ukuran data terlalu besar untuk satu datagram, TCP akan membaginya ke dalam beberapa datagram.

2. IP, Internet Protocol bertanggung jawab setelah hubungan berlangsung. Tugasnya adalah untuk me-rute-kan paket data di dalam network. IP hanya bertugas sebagai pengirim dari TCP dan mencari jalur yang terbaik dalam penyampaian datagram. IP tidak bertanggung jawab jika data tersebut tidak sampai dengan utuh, karena IP tidak memiliki informasi mengenai isi data yang dikirimkan, namun IP akan mengirimkan pesan kesalahan (error message) melalui ICMP (Internet Control Message Protocol) jika hal ini terjadi dan kemudian kembali ke sumber data berasal. Karena IP hanya mengirimkan data tanpa mengetahui urutan data mana yang akan disusun berikutnya, maka hal ini menyebabkan IP mudah untuk dimodifikasi di daerah sumber dan tujuan datagram. Hal inilah yang menyebabkan adanya paket data yang hilang sebelum sampai ke tujuan [14].

2.2.4. Internet Protocol Version (Ipv4)

IP address merupakan pengenal atau identifier yang digunakan untuk memberi alamat pada tiap-tiap komputer dalam jaringan. Format IPv4 address adalah 32 bit yang tiap 8 bit-nya dipisahkan oleh tanda titik, format bakunya adalah seperti demikian xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx, dengan x merupakan bilangan biner 0 atau 1. Atau dengan bentuk empat bilangan dengan titik sebagai pemisahnya seperti berikut xxx.xxx.xxx.xxx, di mana xxx merupakan nilai desimal dari 8 bit biner.

2.2.5. *Network* ID Dan *Host* ID

Dalam 1 alamat IPv4 dibagi menjadi 2 bagain yaitu network ID dan host ID yang ditandai oleh subnet mask. Untuk menandai suatu network ID subnet mask di set ke angka 1 dan untuk menandai suatu host ID subnet mask di set ke angka 0, misalkan alamat 192.168.10.100 dengan subnet masuk 255.255.255.0 menandakan bahwa 192.168.10.0 merupakan alamat network dan .100 merupakan alamat host.

Alamat host ID dalam 1 network ID harus unik, karena host ID harus merupakan satu-satunya alamat yang dimiliki oleh suatu host dalam satu jaringan, jika terdapat 2 host ID dalam 1 network ID maka akan terjadi konflik dalam jaringan tersebut.

Alamat IPv4 dibagi menjadi 3 jenis yaitu :

1. Alamat Unycast

   Alamat ini digunakan dalam komunikasi point to point.

2. Alamat Broadcast

   Alamat ini digunakan dalam komunikasi one to everyone. Biasanya dipakai dalam satu segmen jaringan yang sama.

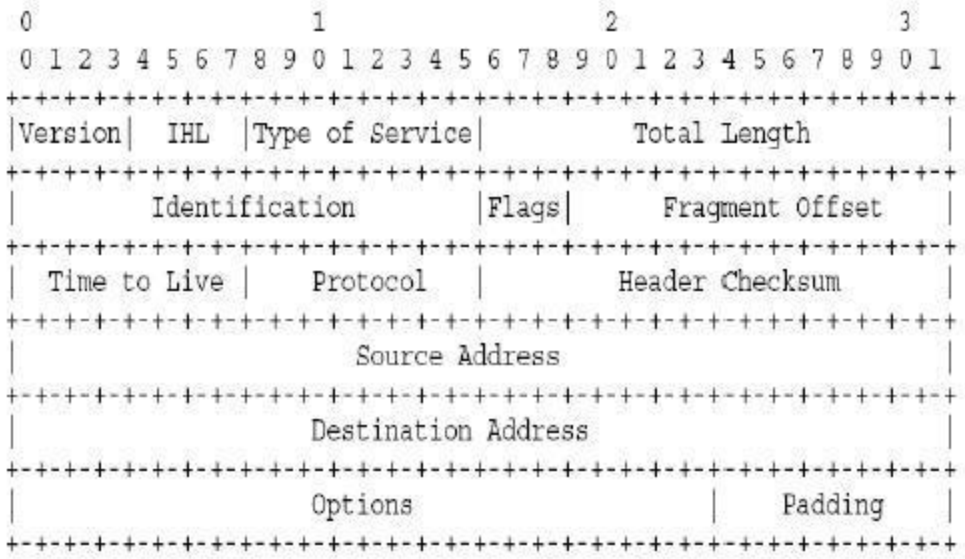
3. Alamat Multicast

   Alamat ini digunakan dalam komunikasi one to many.

2.2.6. Struktur Header Internet Protocol Version 4

IPv4 dibentuk oleh struktur header yang memiliki 14 field dan panjang 32 bit. Setiap field memiliki fungsi dan tujuan masing-masing. Pada header IPv4 masih ditemukan kekurangan-kekurangan yang dapat dijadikan celah bagi penyusup atau orang-orang yang tidak bertanggungjawab untuk melakukan aksi pencurian data pada packet IPv4.

Berikut merupakan struktur header dari IPv4:



Gambar 2. 2 Format *Header* IPv4

Header IPv4 terdiri atas beberapa field, dengan penjelasan sebagai berikut:

1. Version

   Digunakan sebagai tanda version dari header IP yang digunakan, yaitu IPv4.

2. Internet Header Length

   Digunakan untuk mengindikasikan ukuran header IP.

3. Type of Service

   Field ini digunakan untuk menentukan kualitas transmisi dari sebuah datagram IP.

4. Total Length

   Merupakan panjang total dari datagram IP, yang mencakup header IP dan muatannya.

5. Identification

   Digunakan untuk mengidentifikasi sebuah packet IP tertentu yang akan di fragmentasi.

6. Flags

   Berisi dua buah flag yang berisi apakah sebuah datagram IP mengalami fragmentasi atau tidak.

7. Fragment Offset

   Digunakan untuk mengidentifikasikan offset dimana fragmen yang bersangkutan dimulai, dihitung dari permulaan muatan IP yang belum pecah.

8. Time to Live

   Digunakan untuk mengidentifikasikan berapa banyak saluran jaringan dimana sebuah datagram IP dapat berjalan-jalan sebelum sebuah router mengabaikan datagram tersebut.

9. Protocol

   Digunakan untuk mengidentifikasikan jenis protokol lapisan yang lebih tinggi yang dikandung oleh muatan IP.

10. Header Checksum

    Field ini berguna hanya untuk melakukan pengecekan integritas terhadap header IP.

11. Source IP Address

Field ini mengandung alamat IP dari sumber Host yang mengirimkan datagram IP tersebut.

12. Destination IP Address

    Field ini mengandung alamat IP tujuan kemana datagram IP tersebut akan disampaikan.

13. Option

    Fitur pendukung IP yang dapat dipilih, biasanya fitur kemanan.

14. Padding

    Bit-bit 0 tambahan yang ditambahkan ke dalam field ini untuk memastikan header IPv4 tetap berukuran 32 bit.

2.2.7. Internet Protocol Version 6 (IPv6)

Internet Protocol version 6 atau IPv6 merupakan internet protocol yang dirancang untuk menggantikan IPv4. Jumlah 4.3 miliyar alamat yang dimiliki oleh IPv4 dirasa akan mengalami kekurangan untuk memenuhi jumlah kebutuhan alamat IP. IPv6 mulai dikembangkan pada 14 Juli 1999 [11]. Selain untuk memenuhi jumlah alamat yang diperlukan, IPv6 juga didesain untuk memperbaiki kekurangan-kekurangan yang ada pada IPv4.

Kekurangan pada sisi jumlah alamat IP disebabkan oleh kebutuhan manusia yang kian hari kian meningkat serta alih fungsi yang terjadi pada jaringan yang sudah menjadi kebutuhan disebagian aspek kegiatan manusia. Kebutuhan akan hiburan (entertainment), channel untuk televisi maupun video on demand dan sebagainya yang sudah berbasis internet memunculkan suatu kebutuhan akan hadirnya internet protokol yang mampu menangani kebutuhan-kebutuhan tersebut dengan baik.

2.2.8. Fitur IPv6

IPv6 memiliki beberapa fitur yang mampu mengantisipasi perkembangan aplikasi masa depan dan mengatasi kekurangan yang dimiliki IPv4. Fitur-fitur tersebut adalah [14]:

1. Jumlah IP address yang sangat banyak

Sesuai dengan salah satu tujuannya, IPv6 dibuat untuk memenuhi jumlah kebutuhan alamat IP yang sangat banyak. Alamat IPv6 memiliki panjang 128 bit atau setara dengan 3.4 x 10 ^38.

2. Autoconfiguration

   IPv6 dirancang agar penggunanya tidak dipusingkan dengan konfigurasi IP address. Komputer pengguna yang terhubung dengan jaringan IPv6 akan mendapatkan IP address langsung dari router sehingga akan sangat berguna untuk peralatan mobile internet karena pengguna tidak direpotkan dengan konfigurasi sewaktu berpindah tempat dan jaringan.

3. Security

   Kebutuhan security yang hanya menjadi optional pada IPv4 telah menjadi suatu kebutuhan pada IPv6. IPv6 telah dilengkapi dengan protokol IPSec, sehingga semua aplikasi telah memiliki sekuriti yang optimal bagi berbagai aplikasi yang membutuhkan keamanan.

4. Quality of Services

   IPv6 memiliki protokol QoS yang terintegrasi dengan baik sehingga semua aplikasi yang berjalan di atas IPv6 memiliki jaminan QoS.

2.2.9. Perubahan IPv4 Ke IPv6

Fungsi-fungsi yang bekerja pada IPv4 juga ada pada IPv6, sedangkan fungsi-fungsi yang tidak bekerja pada IPv4 dihilangkan pada IPv6. Perubahan dari IPv4 ke IPv6 dapat dibagi dalam beberapa kategori, yaitu :

1. Kapabalitas routing dan pengalamatan yang semakin besar.

   Pengalamatan yang digunakan meningkat dari 32 bit pada IPv4 menjadi 128 bit pada IPv6 untuk mendukung hierarki pengalamatan dan jumlah pengalamatan node-node yang lebih banyak, dan konfigurasi alamat-alamat secara otomatis yang lebih sederhana.

2. Tipe alamat yang baru yang disebut anycast address.

Didefinisikan untuk mengidentifikasi kumpulan node-node dimana paket yang dikirimkan ke sebuah anycast address dikirimkan ke salah satu node. Penggunaan anycast address pada rute sumber memungkinkan node-node mengontrol path tempat aliran traffic.

3. Penyederhanaan format header.

   Beberapa field-field pada header IPv4 telah dihilangkan atau lebih optional untuk mengurangi pemrosesan dari penanganan paket dan untuk menjaga bandwidth dari IPv6 sekecil mungkin walaupun ada peningkatan ukuran alamat.

4. Peningkatan dukungan untuk option-option.

   Perubahan pada penyedian option-option pada header IPv6 memungkinkan proses pelewatan paket yang lebih efisien, batasan-batasan panjang option yang lebih besar, dan fleksibilitas untuk option-option yang mungkin ada di masa depan.

5. Kapabilitas Quality of Services (QoS).

   Sebuah kapabilitas baru ditambahkan untuk memungkinkan pemberian label pada paket-paket dari aliran traffic tertentu dimana pengirim membutuhkan penanganan khusus, seperti quality of service yang bukan default dan service yang bersifat real-time.

6. Kapabilitas authentikasi dan privasi.

   IPv6 juga mendukung authentikasi, kesatuan data, dan kerahasiaan.

2.2.10. Jenis Alamat IPv6

Pada alamat IPv6, jenis alamat broadcast pada IPv4 ditiadakan, namun pada IPv6 muncul sebuah pengalamatan baru dengan nama multicast address. Sehingga jenis alamat yang terdapat pada IPv6 sama dengan IPv4 yaitu sebanyak 3 jenis dengan penjelasan sebagai berikut [15]:

1. Unycast address

   Unicast address merupakan jenis IP address yang digunakan untuk identifikasi sebuah interface saja. Paket yang dikirimkan ke unicast address hanya akan diterima oleh sebuah interface yang menggunakan alamat tersebut.

2. Anycast address

   Anycast address merupakaan jenis IP address yang digunakan untuk identifikasi sekumpulan interface. Paket yang dikirim ke anycast address akan diterima oleh interface terdekat (salah satu interface) dari sekumpulan interface yang menggunakan alamat tersebut. Penentuan interface terdekat adalah berdasarkan pengukuran jarak dari protokol routing. Anycast address tidak lain adalah unicast address yang diberikan pada sekumpulan interface dengan persyaratan khusus, yaitu:

   a. Anycast address hanya digunakan pada router IPv6 saja. Tidak boleh digunakan untuk alamat host.
   b. Anycast address tidak boleh digunakan sebagai alamat asal dari paket IPv6

3. Multicast address

   Multicast address digunakan untuk identifikasi sekumpulan interface. Paket yang dikirim ke multicast address akan diterima oleh semua interface yang menggunakan alamat tersebut. Secara umum, multicast address pada IPv6 berfungsi sama dengan multicast address pada IPv4.

2.2.11. Format Penulisan IPv6

Secara umum format penulisan IPv6 adalah x:x:x:x:x:x:x:x yang dipisahkan oleh double-colon (titik dua ‘:’) dan x yang merepresentasikan panjang 16 bit atau 2 octec kemudian diikuti dengan tanda slash (/) untuk menandakan prefix jaringan. Untuk menuliskan alamat IPv6, dapat dilakukan dengan cara sebagai berikut yaitu:

Alamat IPv6 : 2001:0716:0250:0000:0000:0000:0000:4/64

Dapat ditulis juga dengan bentuk:

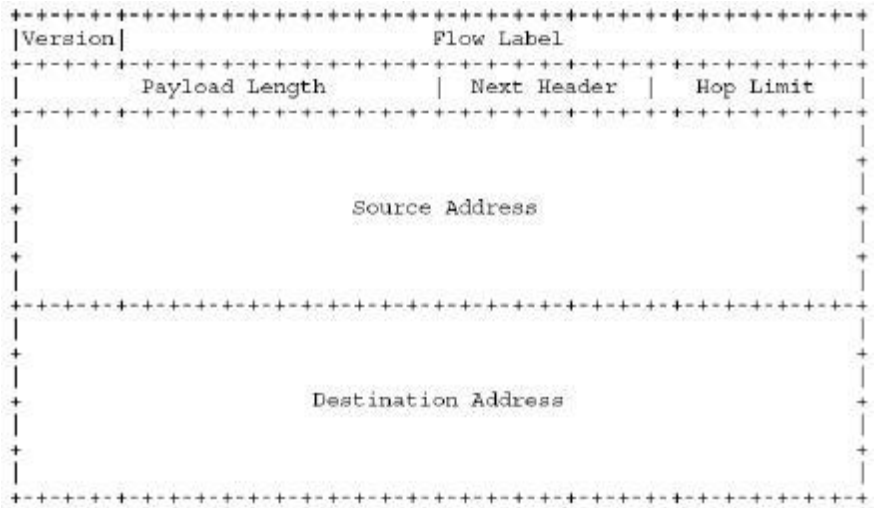
2001:716:250:0:0:0:0:4/64 atau 2001:716:250::4/64

Tanda /64 menunjukan bahwa 64 bit pertama merupakan alamat jaringan yang digunakan. Penulisan 2 double-colon menunjukan bahwa bit yang ada didalamnya adalah bit 0 dan penggunaan 2 double-colon tersebut tidak dapat diberikan sebanyak 2 kali pada 1 alamat IPv6 karena dapat menyebabkan alamat IPv6 menjadi tidak valid.

2.2.12. Struktur *Header Internet Protocol Version* 6

IPv6 memiliki beberapa fitur yang mampu mengantisipasi perkembangan aplikasi masa depan dan mengatasi kekurangan yang dimiliki pendahulunya, yaitu IPv4. IPv6 dirancang sebagai perbaikan dari IPv4. Header pada IPv6 terdiri dari dua jenis, yang pertama, yaitu field yang dibutuhkan oleh setiap paket disebut header dasar, sedangkan yang kedua yaitu field yang tidak selalu diperlukan pada packet disebut header ekstensi, dan header ini didifinisikan terpisah dari header dasar. Header dasar selalu ada pada setiap paket, sedangkan header tambahan hanya jika diperlukan diselipkan antara header dasar dengan data. Header tambahan, saat ini didefinisikan selain bagi penggunaan ketika paket dipecah, juga didefinisikan bagi fungsi sekuriti dan lain-lain. Header tambahan ini, diletakkan setelah header dasar, jika dibutuhkan beberapa header maka header ini akan disambungkan berantai dimulai dari header dasar dan berakhir pada data. Router hanya perlu memproses header yang terkecil yang diperlukan saja, sehingga waktu pemrosesan menjadi lebih cepat. Hasil dari perbaikan ini, meskipun ukuran header dasar membesar dari 20 bytes menjadi 40 bytes namun jumlah field berkurang dari 12 menjadi 8 buah saja.

Berikut ini merupakan format header dari IPv6.



Gambar 2. 3 Format Header IPv6

Field-field pada header IPv6 dapat dijelaskan secara singkat sebagai berikut:

1. Version

Field 4 bit yang menunjukkan versi Internet Protocol, yaitu 6.

2. Prior

   Field 4 bit yang menunjukkan nilai prioritas. Field ini memungkinkan pengirim paket mengidentifikasi prioritas yang diinginkan untuk paket yang dikirimkan, relatif terhadap paket-paket lain dari pengirim yang sama.

3. Flow Label

   Field 24 bit yang digunakan oleh pengirim untuk memberi label pada paket-paket yang membutuhkan penanganan khusus dari router IPv6, seperti quality of service yang bukan default, misalnya service-service yang bersifat real-time.

4. Payload Length

   Field berisi 16 bit yang menunjukkan panjang payload, yaitu sisa paket yang mengikuti header IPv6, dalam oktet.

5. Next Header

   Field 8 bit yang berfungsi mengidentifikasi *header* berikut ini merupakan yang mengikuti header IPv6 utama.

6. Hop Limit

   Field berisi 8 bit unsigned integer. Menunjukkan jumlah link maksimum yang akan dilewati paket sebelum dibuang. Paket akan dibuang bila Hop Limit berharga nol.

7. Source Address

   Field 128 bit, menunjukkan alamat pengirim paket.

8. Destination Address

   Field 128 bit, menunjukkan alamat penerima paket.

2.3. Game

Menurut Erick Zimmerman, dan Katie Salen [8] *Game* merupakan suatu sistem yang memiliki aturan-aturan tertentu, dimana pemain akan terlibat di dalam suatu permasalahan sehingga dapat menghasilkan suatu hasil yang dapat diukur yaitu menang atau kalah. *Game* merupakan sesuatu hal yang dimainkan dengan suatu aturan tertentu

yang biasa digunakan untuk tujuan kesenangan dan dapat juga digunakan sebagai sarana edukasi.

Game umumnya melibatkan stimulasi mental, fisik, atau keduanya. Banyak *game* yang dapat membantu mengembangkan keterampilan praktis yang berfungsi sebagai latihan, atau melakukan peran pendidikan, simulasional, atau psikologis.

2.3.1. Sejarah Dan Perkembangan Game

Game kini sudah tidak asing lagi bagi kita apalagi seperti pada jaman modern sekarang ini. Seiring dengan perkembangan jaman, *game* juga semakin berkembang dari *Atari, computer space, nintendo, sega, playstatiton,* dan sekarang *game* komputer sendiri.

Atari merupakan sebuah fenomena besar dalam dunia video game. Pada tahun 1972, *atari* telah memperkenalkan permainan 'pong' yang sangat terkenal sehingga menjadi best selling item pada tahun 1975. *Atari* seterusnya mengeluarkan *console video game* yang dikenal sebagai *atari* 2600 yang telah menjadi pelopor *video game* sekarang.

Walaupun *atari* dianggap sebagai pelopor kejayaan *video game,* sebaliknya *video game* yang benar-benar dibangun untuk umum adalah *computer space* oleh *Nutting Associetes* pada tahun 1971. Namun *atari* dan *Nutting Associates* mempunyai kaitan erat karena *atari* dikeluarkan oleh orang yang membuat *computer space* yaitu Nolan Bushnell.

Nintendo, satu lagi nama besar dalam dunia *video game* telah menggariskan beberapa konsep yang juga menjadi dasar infrastruktur *video game* pada masa sekarang misalnya penggunaan *pad* serta membuat video game yang sesuai dengan *home theater system* (bisa dihubungkan pada TV atau radio). *Nintendo* merupakan sistem permainan yang paling berjaya di dunia dengan lebih dari 100 juta unit terjual di seluruh dunia.

Konsol pertama yang membuat *user* bisa bermain secara online adalah *dreamcast* yang dikeluarkan oleh sega. Pada 1999, majalah populer *Science* mengumumkan *dreamcast* sebagai produk yang paling inovatif. Saat itu, sega mengutamakan isi dari *game* dimana *game* dibuat serasi dengan kebanyakan platform *video game* seperti *PS2, Xbox, dan GameCube.*

Playstation adalah *game* yang cukup populer pada saat sekarang ini. *Playstation* terbaru yaitu *Playstation 3* (PS3) merupakan *vdeo game* pertama yang menggunakan

DVD. Grafik yang menarik serfta jumlah *game* yang banyak terdapat dipasaran menjadikan *PS3* cukup polpuler.

Game pada komputer diperkenalkan oleh IBM PC pada tahun 1981 dimana komputer jenis ini lebih meningkatkan kreativitas pembuat program *game* di komputer, karena semakin lama komputer IBM *compatible* ini semakin cepat, maka grafiknya semakin baik, dan tempat penyimpanannya semakin besar.

Pada komputer yang berdiri sendiri atau *stand alone* kita hanya bisa bermain melawan komputer atau maksimum dua orang dengan menggunakan *joystick*, *joystick* merupakan alat bantu untuk menghubungkan anatara user dengan komputer untuk mengendalikan sebuah permainan dalam komputer. Sedangkan pada komputer yang menggunakan sistem jaringan bisa lebih dari tiga orang pemain untuk memainkannya, dan bahkan dengan kemajuan jaringan internet memungkinkan ribuan orang yang dapat bermain dalam satu server komputer.

2.3.2. Game Online Multiplayer

Trend perkembangan aplikasi *game* di masa lalu adalah merancang *single user game* dimana *game* jenis ini hanya melibatkan salah satu orang saja. Seiring dengan perkembangan teknologi *internet* dan jaringan, *trend* tersebut mulai berubah. *Multi user game* pun menjadi perhatian para pengembang aplikasi *game* sekarang ini. Meskipun lebih sulit dalam pembuatannya dibandingkan dengan *single user game, game* jenis ini memiliki unsur-unsur yang menarik. *Multi user game* melibatkan dua atau lebih *user* dalam melakukan interaksi. *Multi user game* berkembang menjadi *game* online.

Perkembangan teknologi *game* memungkinkan pemain bermain dengan beberapa orang sekaligus, kini dengan adanya perkembangan pada dunia *game online*, para pemain *game* online dapat saling berinteraksi dengan ribuan orang dari penjuru dunia bukanlah hal yang mustahil. Kemampuan *game* jaringan yang memungkinkan puluhan orang bermain sekaligus mulai mendapatkan pesaing ketika *game* online muncul. Dengan adanya *game online* memungkinkan pemain bermain bersama ribuan orang sekaligus dari berbagai lokasi. Infrastruktur yang dibutuhkan dalam *game* online hanya diperlukan sambungan internet untuk melakukan koneksi pada *game* yang akan dimainkan.

2.4. Tools Yang Digunakan

Tools yang digunakan meliputi perancangan berbasis objek UML,Desktop,dan bahasa pemrograman C#

2.4.1. Perancangan Berbasis Objek

Perancangan berorientasi objek adalah strategi perancangan di mana perancangan sistem memikirkan ‘benda’ dan bukan operasi atau fungsi. Proses perancangan berorientasi objek berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilaku. Beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek secara umum mencakup empat hal, yaitu identifikasi, klasifikasi polimorfisme, dan pewarisan. Keuntungan menggunakan perancangan berorientasi objek diantaranya yaitu alami (Natural), dapat diandalkan (*Reliable*), dapat digunakan kembali (*Reusable*), mudah untuk dalam perawatan (*Maintainable*), dapat diperluas (*Extendable*), dan efisiensi waktu. [9]

2.4.2. Unified Modeling Language UML

Unified Modeling Language (UML) merupakan bahasa standar yang bekerja dalam object-oriented untuk menentukan, memvisualisasikan, merancang, dan mendokumentasikan elemen-elemen informasi yang terdapat dalam sistem software. [4] UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh grady Booch, Jim Rumbaugh dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.



Gambar 2. 4 Logo UML

2.4.3. Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, [4] yaitu :

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

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu :

1. Use Case diagram, untuk memodelkan proses bisnis dan merepresentasikan sebuah interaksi antara aktor dengan sistem
2. Sequence diagram, untuk menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok yakni nama (dan stereotype) , atribut , dan metoda .
3. Statechart diagram , untuk memodelkan perilaku objects di dalam sistem atau menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya)

suatu objek pada sistem. Pada umumnya statechart diagram menggambarkan class tertentu .

4. Activity diagram , untuk memodelkan perilaku Use Cases dan objects di dalam system.
5. Sequence diagram , untuk memodelkan pengiriman pesan (message) antar objects dan juga digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.
6. Collaboration diagram , untuk memodelkan interaksi antar objects seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message
7. Component diagram , untuk memodelkan komponen object atau menggambarkan struktur dan hubungan antar modul yang berisi code, termasuk ketergantungan (dependency) di antaranya.
8. Deployment diagram, untuk memodelkan distribusi aplikasi.

2.4.4. Unity

Unity adalah sebuah aplikasi yang berintegrasi dengan banyak *tools* dan *rapid work flows* yang digunakan untuk membuat konten dua dimensi maupun tiga dimensi yang interaktif dan bersifat *multiplatform*. Unity juga memungkinkan pengembang untuk membuat *object*, meng-import *asset* yang telah disediakan dari luar dan menggabungkan semuanya secara cepat dan efisien. Pengembang dapat menggabungkan beberapa *script* dan waktu untuk *compile* relatif cepat.

Adapun fitur-fitur yang dimilik oleh Unity 3D antara lain sebagai berikut.

a. Integrated development environment (IDE) atau lingkungan pengembangan terpadu

b. Penyebaran hasil aplikasi pada banyak platform:

c. Engine grafis menggunakan Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iOS), and proprietary API (Wii).

d. Game Scripting melalui Mono. Scripting yang dibangun pada Mono, implementasi open source dari NET Framework. Selain itu Pemrogram dapat menggunakan UnityScript (bahasa kustom dengan sintaks Java Script inspired), bahasa C # atau Boo (yang memiliki sintaks Python-inspired).

2.4.5. C# (C Sharp)

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka *.NET Framework*. C# adalah *Java* versi *Microsoft*, sebuah bahasa multi *flatform* yang didesain untuk bisa berjalan di berbagai 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. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem *klien-server* (*hosted system*) maupun sistem *embedded* (*embedded system*), mulai dari program aplikasi yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada program aplikasi yang sangat kecil.

Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran program aplikasi yang dibuat dengan menggunakan bahasa pemrograman C.

2.4.6. 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. 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*.

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.

2.5. Kecerdasan Buatan

Kecerdasan Buatan atau *Artificial Intelligence* merupakan salah satu bagian dari ilmu komputer yang membuat agar mesin komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Adanya irisan penggunaan kecerdasan buatan di berbagai disiplin ilmu tersebut menyebabkan cukup rumitnya untuk mengklasifikasikan kecerdasan buatan menurut disiplin ilmu yang menggunakannya, untuk memudahkan hal tersebut maka untuk pengklasifikasian lingkup kecerdasan buatan didasarkan pada output yang diberikan yaitu [6]:

1. Sistem pakar *(expert system)* disini computer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.

2. Pengolahan Bahasa Alami *(Natural Language Processing)* dengan pengolahan bahasa alami ini diharapkan user dapat berkomubikasi dengan komputer dengan menggunakan bahasa sehari-hari.

3. Pengenalan ucapan *(Speech Recognition)* melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan computer dengan menggunakan suara.

4. Robotika dan sistem sensor *(Robotics & Sencory Systems)*

5. Computer Vision, mencoba untuk dapat mengintrepertasikan gambar atau objek-objek tampak melalui komputer.

6. Intelligent Computer-aided Instruction Computer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

7. Game playing.

2.5.1. Algoritma Pencarian

Permasalahan pencarian dapat diselesaikan dengan beberapa metode yaitu [6]:

1) Metode pencarian yang pertama adalah metode sederhana yang hanya berusaha mencari kemungkinan penyelesaian yang disebut juga pencarian buta.

2) Metode yang lebih kompleks yang akan mencari jarak terpendek.Metode ini adalah British *Museum Procedure*, *Branch and Bound*, *Dynamic Programming*, *Best First Search*, *Greedy Search*,A* *(A Star)* dan *Hill Climbing Search*. Metode-metode ini digunakan pada saat perjalanan untuk mencari kemungkinan menjadi perhitungan.

Metode pencarian sangat penting untuk menyelesaikan permasalahan karena setiap *state* atau keadaan menggambarkan langkah-langkah untuk menyelesaikan permasalahan. Dalam sebuah permainan, metode pencarian akan menentukan apa yang harus dilakukan dimana setiap state menggambarkan kemungkinan posisi pada suatu saat. Metode pencarian adalah bagian dari kesimpulan dimana setiap state menggambarkan hipotesis dalam sebuah rangkaian deduktif.

Menurut cara algoritma mengembangkan node dalam proses pencarian, gambar bagan metode penulusuran dibagi menjadi dua golongan, yakni pencarian buta *(blind search)* dan pencarian terbimbing *(heuristic search).*

Beberapa contoh algoritma pencarian yang menggunakan metode *heuristic search* adalah : *Best First Search, Greedy Search,* A* *(A Star) Search*, dan *Hill Climbing Search*.

2.5.2. Metode Pencarian *Heuristik*

Kata *heuristik* berasal dari kata kerja bahasa Yunani, *heuriskein*, yang berarti ‘mencari’ atau ‘menemukan. Dalam dunia pemograman, sebagaian orang menggunakan kata *heuristik* sebagai lawan kata dari algoritmik, dimana kata heuristik ini diartikan sebagai „suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditemukan“. Di dalam mempelajari metode – metode pencarian ini, kata heuristik diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi [6].

2.5.3. Pencarian Buta

Pencarian buta *(Blind Search)* adalah pencarian solusi tanpa adanya informasi yang dapat mengarahkan pencarian untuk mencapai *goal state* dari *current state*. Informasi yang ada hanyalah definisi *goal state* itu sendiri sehingga algoritma dapat mengenali *goal state*. Apabila tidak ada informasi maka pencarian buta dalam kerjanya akan memeriksa *node-node* secara tidak terarah dan kurang efisien untuk kebanyakan kasus karena banyanknya node yang dikembangkan. Beberapa contoh algoritma yang termasuk blind search adalah *Breadth First Search*, *Uniform Cost Search*, *Depth First Search*, *Depth Limited Search*, *Interative Deepening Search* dan *Bidirectional Search* [6].

2.5.4. Pencarian Terbimbing

Pencarian terbimbing *(Heuristic Search)* mempunyai informasi tentang biaya untuk mencapai *goal state* dari *current state*. Pencarian terbimbing dapat melakukan pertimbangan untuk mengembangkan atau memeriksa *node-node* yang mengarah ke *goal state*. Pencarian terbimbing untuk menghitung *cost* ke *goal state* digunakan fungsi *heuristic*. Fungsi *heuristic* berbeda dari pada algoritma dimana *heuristic* lebih merupakan perkiraan untuk membantu algoritma dan tidak harus valid setiap waktu. Beberapa contoh algoritma pencarian yang mengguanakan metode *heuristic search* adalah *Best First Search*, *Greedy Search*, A* *(A Star)* dan *Hill Climbing Search* [6].

2.6. AI Yang Digunakan

AI yang digunakan pada pembangunan *game Fire Shooter 2D Mutiplayer* adalah

2.6.1. Algoritma A* (*A Star*)

Algoritma A* *(A Star)* merupakan perbaikan dari metode BFS dengan memodifikasi fungsi heuristicnya. A* *(A Star)* akan meminimumkan total biaya lintasan. Pada kondisi yang tepat, A* akan memberikan solusi yang terbaik dalam waktu yang optimal. Pencarian menggunakan algoritma A* mempunyai prinsip yang sama dengan algoritma BFS hanya saja dengan dua faktor tambahan yaitu :

1. Setiap sisi mempunyai cost yang berbeda-beda, seberapa *cost* untuk pergi dari satu simpul ke simpul lain.

2. Cost dari setiap simpul ke simpul tujuan bisa diperkirakan. Ini membantu pencarian sehingga lebih kecil kemungkinan kita mencari ke arah yang salah.

Cost untuk setiap simpul tidak harus berupa jarak. *Cost* bisa saja berupa waktu bila ingin mencari jalan dengan waktu tercepat untuk dilalui. Algoritma A* bekerja dengan prinsip yang hampir sama kecuali dengan dua perbedaan yaitu :

1. Simpul-simpul di list terbuka diurutkan oleh *cost* keseluruhan dari simpul awal ke simpul tujuan, dari *cost* terkecil sampai *cost* terbesar. *Cost* keseluruhan dihitung dari *cost* dari simpul awal ke simpul sekarang *(current node)* ditambah *cost* perkiraan menuju simpul tujuan.
2. Simpul di list tertutup bisa dimasukkan ke list terbuka bila jalan terpendek menuju simpul tersebut ditemukan.

Cost antara simpul adalah jaraknya dan perkiraan *cost* dari suatu simpul ke simpul tujuan adalah penjumlahan jarak dari simpul tersebut ke simpul tujuan. Untuk lebih mudah dimengerti, berikut rumusnya:

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

keterangan :

f(n) = fungsi evaluasi

g(n) = biaya yang sudah dikeluarkan dari keadaan awal sampai keadaan n

h(n) = estimasi biaya untuk sampai pada suatu tujuan mulai dari n

Node dengan nilai terendah merupakan solusi terbaik untuk diperiksa pertama kali pada g(n) + h(n). Dengan fungsi *heuristic* yang memenuhi kondisi tersebut maka pencarian dengan algoritma A* dapat optimal. Keoptimalan dari A* cukup langsung dinilai optimal jika h(n) adalah *admissible heuristic* yaitu nilai h(n) tidak akan memberikan penilaian lebih pada *cost* untuk mencapai tujuan. Salah satu contoh dari admissible *heuristic* adalah jarak dengan menarik garis lurus karena jarak terdekat dari dua titik adalah dengan menarik garis lurus [6].

Beberapa terminologi dasar yang terdapat pada algoritma ini adalah *starting point*, *current node*, simpul, *neighbor node*, *open set*, *closed set*, *came from*, harga *(cost)*, *walkability*, *target point.*

1. Simpul awal adalah sebuah terminologi untuk posisi awal sebuah benda.

2. Current node adalah simpul yang sedang dijalankan dalam algoritma pencarian jalan terpendek.

3. Simpul adalah petak-petak kecil sebagai representasi dari area *pathfinding*. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.

4. *Neighbor node* adalah simpul-simpul yang bertetangga dengan *current node.*

5. *Open List* adalah tempat menyimpan data simpul yang mungkin diakses dari *starting point* maupun simpul yang sedang dijalankan.

6. *Closed List* adalah tempat menyimpan data simpul sebelum *current node* yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan.

7. *Parent* adalah tempat menyimpan data ketetanggaan dari suatu simpul, misalnya y *parent* x artinya *neighbor node* y dari *current node* x.

8. Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari starting point ke current node, dan H, jumlah nilai perkiraan dari sebuah simpul ke target point.

9. Target point yaitu simpul yang dituju.

10. *Walkability* adalah sebuah atribut yang menyatakan apakah sebuah simpul dapat atau tidak dapat dilalui oleh *current node.*

Algoritma A* secara ringkas langkah demi langkahnya adalah sebagai berikut:

1. Tambahkan simpul awal ke dalam open list.

2. Ulangi langkah berikut sampai pencarian berakhir:

a. Carilah simpul n dengan biaya F(n) paling rendah, dalam *open list*. Simpul dengan biaya F terendah kemudian disebut *current node.*

b. Keluarkan current node dari open list dan masukkan ke dalam *closed list.*

c. Untuk setiap 8 simpul *(neighbor node)* dari *current node* lakukan langkah berikut:

1). Jika sudah terdapat dalam *closed list* atau tidak *walkable*, maka abaikan, jika tidak lanjutkan.

2). Jika belum ada pada *open list*, tambahkan ke *open list.* Simpan *current node* sebagai parent dari *neighbor node* ini. Simpan harga F masing-masing simpul.

3). Jika sudah ada dalam *open list*, periksa apakah ini jalan dari simpul ini ke *current node* yang lebih baik dengan menggunakan biaya G sebagai ukurannya. Simpul dengan biaya G yang lebih rendah berarti bahwa ini adalah jalan yang lebih baik. Jika demikian, buatlah simpul ini *(neighbor node)* sebagai *parent* dari *current node*, dan menghitung ulang nilai G dan F dari simpul ini.

d. Berhenti ketika:

1. Menambahkan target *point* ke dalam *closed list,* dalam hal ini jalan telah ditemukan, atau,

2. Gagal untuk menemukan target *point*, dan *open list* kosong. Dalam kasus ini, tidak ada jalan.

e. Walaupun telah mencapai target *point*, jika masih ada *neighbor node* yang memiliki nilai yang lebih kecil, maka simpul tersebut akan terus dipilih sampai bobotnya jauh lebih besar atau mencapai target point dengan bobot yang lebih kecil dibanding dengan simpul sebelumnya yang telah mencapai target *point*.

f. Pada saat pemilihan simpul berikutnya, nilai F(n) akan dievaluasi, dan jika terdapat nilai F(n) yang sama maka akan dipilih berdasarkan nilai G(n) terbesar.

3. Simpan jalan. Bekerja mundur dari target *point*, pergi dari masing-masing simpul ke simpul *parent* sampai mencapai *starting point.*