

BAB II

TEORI PENUNJANG

1.1 Permainan Komputer (Computer Game)

Games adalah sebuah aktifitas yang dilakukan secara sukarela, dimana setiap satu atau lebih pemainnya mengikuti suatu aturan yang membatasi apa yang bisa mereka lakukan dan melakukan konflik buatan yang dimana menghasilkan hasil yang terukur[6]. Namun dalam penelitian ini bentuk games yang akan dibicarakan adalah video games.

Video games merupakan sebuah permainan atau game elektronik yang membutuhkan sebuah interaksi dari manusia dengan tampilan daripada video game itu sendiri. Dalam interaksi ini video game pun harus dapat memberikan sebuah respon dalam bentuk tampilan atau pun audio dan akan ditampilkan oleh media seperti berupa tampilan atau audio. Respon ini berguna untuk memberikan interaktivitas dengan manusia yang berperan sebagai player dalam video game[7].

Video game pada umumnya dimainkan dengan sebuah perangkat elektronik yang bisa disebut dengan *platform*. Ada banyak macam *platform* dikhususkan untuk bermain video game hingga *platform* yang memperbolehkan untuk bermain video games, contohnya seperti Xbox, Nitendo3Ds, Playstation hingga Personal Computer dan Laptop[7].

Menurut Andrew Rollings dan Earnest Adams bahwasanya games dibagi kebeberapa kategori yaitu [8]:

1. Action Games

Kategori ini adalah salah satu kategori tertua. Pada kategori ini game mengetes kemampuan pemain dalam penggunaan relfleks dan kordinasi mata dan tangan.

2. *Strategy Games*

Game ini bersal daripada kategori sebelumnya yaitu board games yang dimana memiliki jumlah aturan yang banyak. Dalam strategi terdapat dua subkategori yaitu Real Time Strategy (RTS) yang permainan strategi bersifat realtime dan dibutuhkan kecakapan pemain, lalu ada Turn Based Strategy (TBS) dimana pemain dan lawannya akan saling bergantian untuk melawan atau bergerak.

3. *Role Playing Game*

Game di kategori ini biasanya memiliki ciri khas dengan memiliki alur cerita yang kuat dan meningkatkan pengalaman pemain. Permainan peran (Role-playing games), merupakan permainan video yang menempatkan pemain sebagai tokoh dalam permainan tersebut untuk memecahkan suatu misteri dengan menyelesaikan berbagai macam puzzle dan quest. Jenis ini biasanya memiliki alur cerita yang kompleks

4. *Sport Game*

Pertandingan olahraga seperti di dunia nyata yang dibawa ke dalam sebuah game. Permainan olahraga (Sports games), yaitu jenis permainan video yang menuntut keterampilan pemain untuk melakukan pertandingan olahraga secara virtual, seperti pertandingan sepak bola, basket, dan sebagainya.

5. *Construction and Management Simulation Game*

Jenis game ini bertujuan untuk membuat sesuatu dalam konteks proses yang berkelanjutan. Semakin baik pula hasil yang diperoleh. Konstruksi dan simulasi manajemen (CMS) adalah jenis simulasi permainan di mana pemain membangun, memperluas atau mengelola komunitas fiksi atau proyek dengan sumber daya yang terbatas. Strategi video game kadang-kadang menggabungkan aspek CMS ke dalam perekonomian. Pemain harus mengelola sumber daya sekaligus

memperluas proyek mereka. Permainan CMS murni berbeda dari game strategi. Tujuan pemain adalah tidak untuk mengalahkan musuh, tetapi untuk membangun sesuatu dalam konteks yang berkelanjutan

6. *Vehicle Simulation*

Game ini mencoba menciptakan bagaimana rasa mengemudi atau menerbangkan suatu kendaraan sebagai media simulasi. Game simulasi kendaraan adalah genre video game yang mencoba untuk memberikan pengalaman pemain dengan interpretasi realistis operasi berbagai jenis kendaraan. Kendaraan yang dimaksud termasuk mobil, pesawat, perahu, pesawat ruang angkasa, kendaraan militer dan berbagai kendaraan lain. Tantangan utama game ini adalah untuk menguasai dan mengemudi kendaraan dari perspektif pilot atau driver.

7. *Adventure Games*

Game petualangan dengan sebuah cerita interaktif tentang karakter yang dikendalikan oleh pemain. Permainan petualangan aksi (Action-adventure games), yaitu jenis permainan video yang menempatkan pemain sebagai tokoh dalam permainan tersebut, mirip dengan permainan peran, tetapi ditambah dengan unsur-unsur aksi, misalnya perkelahian dan tembak- menembak.

8. *Artificial Life*

Jenis ini melibatkan proses pemodelan biologis dan seringkali untuk mensimulasikan siklus kehidupan makhluk hidup. Game simulasi kehidupan sekitar, memelihara dan menumbuhkan populasi. Pemain diberikan kekuatan untuk mengontrol kehidupan makhluk otonom atau orang-orang. Artificial Life berhubungan dengan penelitian ilmu komputer di kehidupan buatan.

1.2 Narrative Choice-Driven Game

Narrative choice-driven merupakan sebuah bentuk media entertainment yang memungkinkan user atau penikmat bisa berinteraksi dan memberikan sebuah perubahan daripada narrative tersebut. Perubahan narrative ini bisa dicapai dengan melakukan perubahan pada hal-hal seperti rangkaian event atau cerita atau bagaimana sikap dan aksi karakter yang disediakan pada narrative tersebut. Sebelum munculnya video game sebagai salah satu bentuk entertainment, bentuk paling awal dari pada penerapan Narrative *choice-driven* terdapat pada buku bernama “choose-your-own adventure” yang diterbitkan pada tahun 1970. Dalam buku ini menerapkan sebuah narrative yang bersifat non-linear dengan menggunakan pilihan atau choice sebagai penentu alur. Choice dalam buku ini muncul dalam bentuk pilihan halaman yang akan dipilih oleh penikmat dalam menentukan cerita seperti *turn to page 104 to follow him; turn to page 48 to go back home*, dan setiap pilihan memiliki akhir cerita yang berbeda-beda [9].

Setelah masuk pada era modern dan video game sudah mulai dikenal khalayak umum, muncul beberapa video game yang menerapkan choice driven narrative seperti *Baldur's Gate*, *Star Wars: Knights of the Old Republic*, *Zork* dan *Dragon Age Origins*. Salah satu developer game bernama “BIOWARE” kiat menyuarkan game berbasis narrative choice-driven hingga titik dimana mereka menolak untuk membuat game yang tak berbasis narrative choice-driven[9].

1.3 Petri Net

Petri Net merupakan sebuah *mathematical* dan *graphical* modeling tool yang bisa dipergunakan untuk berbagai macam sistem. Petri Net pun biasa dipergunakan untuk mendeskripsikan sebuah processing system yang bersifat concurrent, asynchronus, distributed hingga pararel. Sebagai sebuah tool yang bersifat graphical, Petri Net memiliki kegunaan layaknya flowchart atau block diagram dengan perbedaannya yaitu terdapatnya token. Token ini merupakan sebuah mekanisme agar dapat untuk mesimulasikan bagaimana dinamika dari pada pada suatu sistem [4].

1.3.1 Defnisi Petri Net

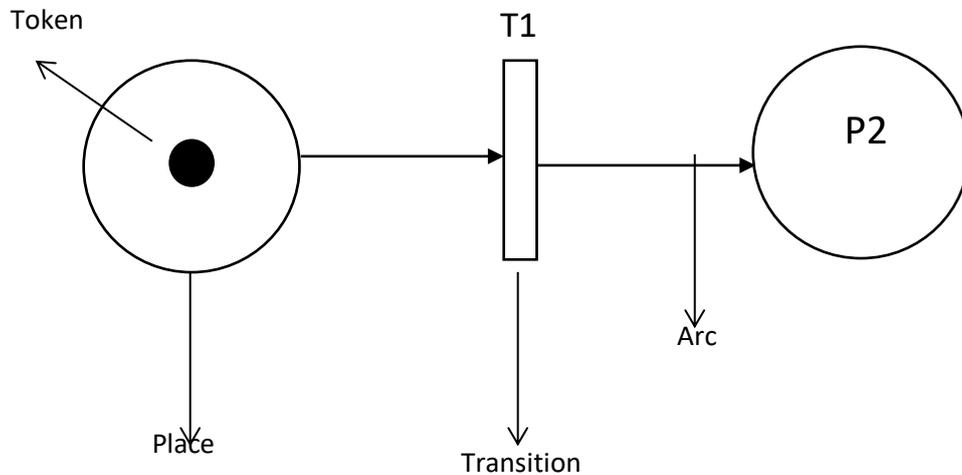
Petri Net terdiri dari beberapa komponen dasar yaitu *places* yang berbentuk lingkaran, *transitions* yang berbentuk persegi panjang atau garis lurus yang tebal lalu ada *arcs* yang berbentuk seperti panah dan berguna untuk menghubungkan antara satu *places* dengan *places* lainnya melalui sebuah *transitions* dan *places* [4].

Berikut ini adalah simbol daripada Petri Net secara lengkap[10]:

- P adalah sekumpulan *Places* yang berjumlah finite.
- T adalah *Transitions* yang berjumlah finite.
- $A \subseteq (P \times T) \cup (T \times P)$ adalah sekumpulan *Arc* yang menghubungkan dari *Places* ke *Transitions* dan ke *Transitions* kembali.
- $W : A \rightarrow \{1, 2, 3, \dots\}$ adalah fungsi *weight* bagi *arc*.

Sekumpulan *places* akan dituliskan dengan $P = \{p_1, p_2, \dots, p_n\}$; dan *Transitions* akan dituliskan dengan $T = \{t_1, t_2, \dots, t_m\}$. Sedangkan *arc* akan dituliskan dengan (p_i, t_j) or (t_j, p_i) [10].

Gambar dibawah merupakan contoh struktur daripada petri pada umumnya :



Gambar II. 1 Diagram Petri Net

Dari gambar Petri Net diatas dapat kita difenisikan sebagai berikut :

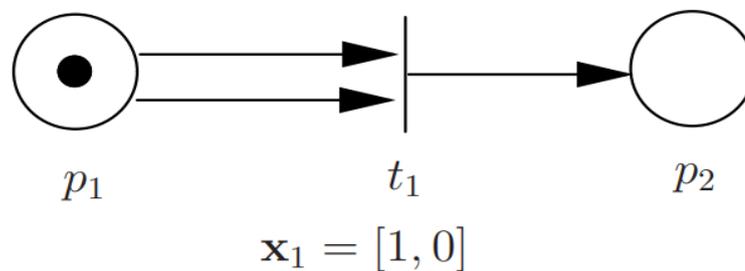
- $P = \{p1, p2\}$ memiliki arti bahwa dalam Petri Net terdapat dua buah *places*
- $T = \{t1\}$ memiliki arti bahwa terdapat satu *transition* dalam struktur Petri Net ini.
- $A = \{(p1, t1), (t1, p2)\}$ memiliki arti bahwa *places* satu atau $P1$ terkoneksi dengan *transition* satu atau $T1$ dan $T1$ terkoneksi dengan $P2$.
- $w(p1, t1)=1$ dan $w(t1, p2)=1$ memiliki arti bahwa masing-masing *arc* memiliki jumlah *weight* bernilai satu.

Selain places dan transition kita juga bisa melihat sebuah titik hitam atau *black dot*. Titik ini dinamakan *token*, berguna sebagai tanda terhadap kondisi tertentu pada sebuah places. *Token* ini disimpan pada places dan jumlah token sekurang-kurangnya sama dengan nilai daripada *arc weight*.

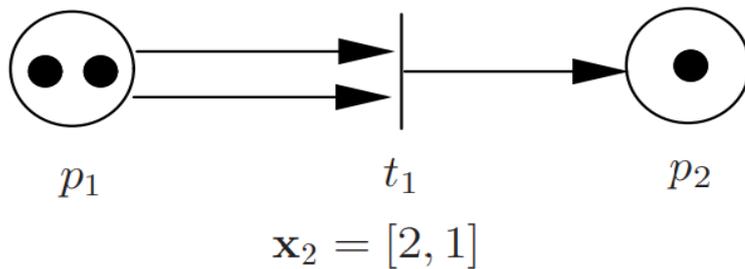
Menempatkan token pada Petri Net dinamakan dengan *marking* sedangkan Petri Net yang sudah ditempatkan dengan *token* disebut dengan *marked Petri Net* namun untuk mempermudah pembacaan maka *marked Petri Net* dapat juga di definisikan *Petri Net* [10].

1.3.2 Definisi Marked Petri Net

Seperti yang telah kita ketahui bahwa Petri Net yang memiliki token didalam placency disebut marked Petri Net namun bisa juga disebut dengan nama Petri Net saja diakrenakan *token* itu sendiri masih menjadi bagian utama daripada Petri Net itu sendiri. Pada Petri Net yang tidak di marked atau tanpa ada token terdiri 4-tuple (P,T,A,w) namun untuk marked Petri Net terdiri dari 5-tuple (P,T, A, w, x) .



Gambar II. 2 Petri Net Satu Marking



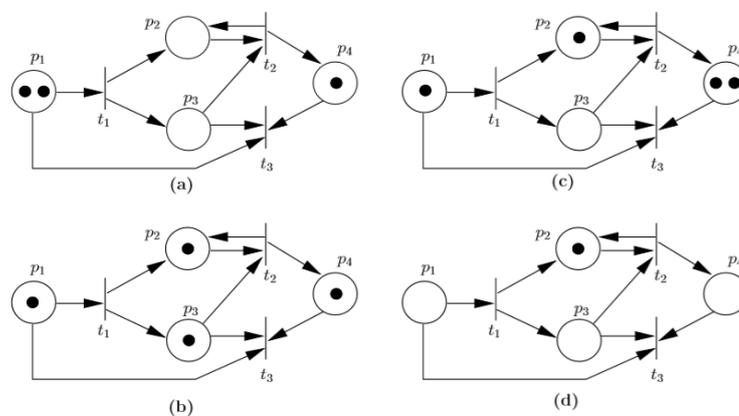
Gambar II. 3 Petri Net Tiga Marking

1.3.3 Definisi Enable Transitions

Sebuah *transitions* dapat disebut *enable* jika jumlah token pada place yang terkoneksi dengan sebuah *transitions* memiliki *weight* yang lebih atau sama dengan *weight* yang pada *arc* yang terkoneksi. Jika sebuah *place* memiliki jumlah token yang kurang dari jumlah *weight* dengan *arc* yang terkoneksi maka *transitions* pada Petri Net tersebut tidak *enable*. Alasan dari adanya hal ini dikarenakan token merupakan mekanisme yang digunakan untuk menentukan satisfikasi terhadap *places* yang selalu diasosiasikan dengan *condition* yang di *enable* melalui sebuah *transitions*[10].

1.3.4 Dinamika Petri Net

Mekanisme *transitions* dalam memenuhi suatu kondisi dalam petri-net ditentukan oleh pergerakan token melewati keseluruhan diagram Petri Net. Pergerakan token ini dapat merubaha kondisi di dalam sebuah diagram Petri Net. Ketika sebuah transisi sudah menjadi *enable* maka dalam istilah Petri Net transisi tersebut dapat melakukan *fire* namun juga biasa disebut *firing*. *Enable transions*. Untuk lebih jelas bias dilihat dari gambar dibawah ini [10]:



Gambar II. 4 Proses Firing

Dari gambar diatas bawah pada diagram (a) bawah diagram tersebut sudah *enabled* dana siap untuk di firing. Hasil dari firing pada diagram (a) bisa dilihat pada diagram (b) diman msaing token terisi pada places yang terkoneksi dengan *arc* dengan ukuran *weightnya*. Pada diagram (b) proses firing terjadi lagi pada t2 dan token berpindahke places p4 hsail berada pada diagram (c). lalu terjadi firing kembali pad t3 namun dalam hal ini berbeda dikarenakan tidak ada place untuk menampung *token* maka *token* otomatis hancur[10].

1.3.5 State Space Analysis

State Space Analysis merupakan metode yang digunakan untuk menghitung setiap marking dan perubahan yang terjadi pada element yang terdapat pada petri net. *State Space* ini melakukan kalkulasi pada model petri net untuk melihat setiap sejumlah kemungkinan eksekusi, firing hingga transisi yang bisa dilakukan oleh model petri net, metode ini sangat efektif untuk mendeteksi error dalam sistem. Metode *State Space Analysis* ini dilaksanakan menggunakan pilihat *state space report* yang terdapat pada CPN Tools. *State space report* sendiri menjalankan metode *State Space Analysis* beserta laporan terhadap property di dalam petri net. Property tersebut masing adalah *Behavioural Properties*, *Reachability Properties*, *Boundedness Properties*, *Home Properties*, *Liveness Properties*, *Fairness Properties*[11].

1.4 CPN Tools

Petri Net terbagi menjadi dua *level* yaitu *low-level Petri Net* dan *high-level Petri Net*. *Colored Petri Net* merupakan Petri Net yang berlevel tinggi. Dalam *colored Petri Net* terdapat karakteristik yang mengkombinasikan antara bahasa pemograman dan Petri Net itu sendiri dan juga terdapat sebuah fitur yang bernama hierarchical CP-net yang dimana memungkinkan memodelkan sebuah model Petri Net yang berukuran besar namun tergabung dari beberapa model kecil konsep

tersebut dinamakan *fusion places*. *colored Petri Net*-pun bisa memungkinkan untuk melakukan simulasi dan analisis terhadap model yang di bangun. Metode analisis tersebut dinamakan *state space analysis*. Tools yang tepat untuk menggunakan *colored-Petri Net* adalah aplikasi bernama CPN tools, aplikasi ini memiliki kemampuan untuk melakukan simulasi otomatis terhadap model Petri Net serta melakukan metode *state space analysis*[12].

1.5 Unity

Unity3D atau sering disebut dengan unity saja merupakan game engine yang dibuat untuk menciptakan sebuah aplikasi game dan didalamnya sudah terdapat sebuah IDE (*Integrated Development Environment*) yang dimana software tidak membutuhkan software lainnya untuk melakukan pembangunan aplikasi game, hal seperti code editor dan compiler sudah tersedia di dalam unity [13]. Selain itu juga unity pun terdapat komunitas serta sebuah *store* yang menyediakan berbagai macam asset yang dapat dipergunakan bagi pengguna unity dan juga terdapat komunitas yang besar untuk berbagi ilmu pengetahuan[14].



Gambar II. 5 Lambang