

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Game

##### 2.1.1 Pengertian *Game*

Menurut Tracy Fullerton Game adalah bagian integral dari semua budaya manusia yang dikenal. Permainan digital, dalam berbagai format dan genre , hanyalah sebuah ungkapan baru dari metode interaksi sosial kuno ini . (Tracy Fullerton,2014)

Sedangkan menurut Raph Koster definisi game yang baik adalah satu yang mengajarkan semua hal yang harus ditawarkan sebelum pemain berhenti bermain (Raph Koster,2013)

Sedangkan menurut Chris Crawford(Chris Crawford on Game Design,2003)

- Ekspresi kreatif adalah seni jika dibuat untuk keindahannya sendiri, dan *entertainment* jika dibuat untuk uang.
- Sebuah *entertainment* adalah sesuatu yang dapat dimainkan jika bersifat interaktif.
- Jika tidak ada *goal* yang ikut andil dalam permainan maka itu adalah sebuah *toy*.
- Jika tantangan tidak memiliki “agen aktif yang dapat ditemani bermain” maka itu adalah sebuah *puzzle*.
- Terakhir, apabila pemain dapat mengalahkan lawan, tetapi tidak menyerang atau mengganggu performa lawan, maka konfliknya adalah kompetisi. Tetapi apabila, penyerangan diperbolehkan maka konfliknya dapat dikualifikasi sebagai sebuah game.

Game sendiri telah ada sejak jaman dahulu kala dan masih bersifat fisik, sedangkan *video game* sendiri pertama kali ditemukan pada awal tahun 1950 saat akademis ilmuwan komputer pertama kali mendesain sebuah permainan sederhana yang digunakan dalam penelitiannya. Game saat ini telah berkembang dengan sangat pesat dan telah diterima oleh masyarakat luas. Seiring dengan waktu juga

genre dari Game semakin berkembang dan bukan hanya genre *Action*, *Survival* yang ada saat ini tetapi ada juga penggabungan dari beberapa genre seperti Horror dan Survival yang menciptakan *Survival Horror*.

### 2.1.2 Unsur-unsur dari *Gameplay*

Unsur-unsur penyusun game sendiri ada 4 menurut Chris Crawford dalam bukunya yang berjudul "*Chris Crawford on Game Design*" unsur ini kadang juga disebut sebagai *gameplay*.

- alat : *Game* terkadang diklasifikasikan berdasarkan alat yang digunakan untuk memainkan game tersebut (contoh : miniatur, bola, komputer dll). Selain itu banyak juga dari game *tool* yang dianggap sebagai *token*, yang digunakan untuk mewakili benda lain, token dapat berupa pion pada papan, uang mainan dll.
- aturan : Walaupun terkadang game diklasifikasi dari alat tetapi game juga dapat diklasifikasi dari aturan main game tersebut. Aturan sangat bervariasi dan berubah-ubah sehingga dapat menciptakan sebuah permainan baru. Contoh dari aturan adalah penentuan giliran pemain, hak dan kewajiban pemain, dan goal masing-masing pemain.
- *Skill*, strategi, and kesempatan: sebuah tools dan rule game akan menciptakan sebuah *skill*, strategi, keberuntungan. Contoh dari skill adalah dalam permainan menembak, contoh dari strategi adalah permainan catur dll, sedangkan contoh kesempatan adalah dalam permainan *blackjack* atau permainan dadu, dll.
- *Single-player games*: Kebanyakan game membutuhkan lebih dari satu pemain. Tetapi, *Single-player games* tetap membutuhkan sebuah tantangan dan tantangan ini dapat berasal dari banyak hal bisa dari tantangan terhadap skill, waktu, ataupun strategi.

### 2.1.3 Jenis-jenis genre Game

- *RPG (Role Playing Game)*

*RPG* Adalah game yang mengizinkan pemain untuk mengeksplorasi dunia dalam game yang biasanya mengandung unsur inti *experience* dan *leveling*. Contoh: Final Fantasy, Mass Effect .

- *FPS (First Person Shooting)*

*FPS* adalah game yang memungkinkan pemain untuk menembak musuh dengan sisi pandang orang pertama sehingga tampak seperti pemain yang memegang senjata. Contoh : Counter Strike, Battlefield .

- *TPS (Third Person Shooting)*

*TPS* adalah game yang hampir mirip dengan game *FPS* tetapi sudut pandang yang digunakan dalam game ini adalah sisi pandang orang ketiga. Contoh : Lost Planet. Dead Space.

- Strategi

Strategi adalah game yang membutuhkan pemikiran strategi dalam penyelesaian permainan ini. Dalam game ini ada banyak jenis kontrol baik itu mengontrol bangunan maupun mengontrol pasukan. Contoh : Warcraft, Stronghold Crusader.

- *Sports*

*Sports* adalah game yang bertemakan permainan olahraga. Biasanya game *Sports* akan mengikuti aturan dari pemain olahraga di dunia nyata . Contoh : FIFA , PES , NBA2K.

- *Racing*

*Racing* adalah game yang bertemakan balapan yang memungkinkan pemain untuk memilih kendaraan yang akan digunakan dan berlomba untuk sampai ke tujuan. Contoh : Need For Speed, MotoGp.

- *Action Adventure*

*Action adventure* adalah game yang biasanya berfokus pada satu karakter dan memiliki ciri khas cerita yang menarik. Contoh : Bioshock Infinite, Batman seri .

- *Survival Horror*

*Survival Horror* adalah game dimana game mencoba menakuti pemain dan biasanya pemain hanya akan diberi sedikit nyawa dan amunisi dalam permainan game ini, dan tujuan dari game ini adalah pemain bertahan hidup hingga akhir dari game. Contoh : The Last of Us , Resident Evil.

## **2.2. AI (Artificial Intelligence)**

### **2.2.1 Pengertian**

*Artificial Intelligence* sendiri diartikan oleh Margaret A. Boden adalah sesuatu yang Berusaha membuat komputer melakukan hal-hal yang bisa dilakukan oleh pikiran(Margaret A. Boden,2016).

Sedangkan menurut Neil Kirby AI adalah kemampuan untuk bertindak cerdas dalam menghadapi masalah yang berubah-ubah.(Neil Kirby,2014). Ada beberapa pengspesialisasian AI diantaranya adalah sebagai berikut:

- *AI games*

*AI games* adalah suatu pemrograman komputer yang mampu membuat AI mampu melawan pemain maupun membantu pemain selama jalan permainan.

- *Expert System*

Expert system Adalah sebuah pemrograman komputer di mana AI mampu membuat keputusan berdasarkan situasi di kehidupan nyata sebagai contoh *expert system* yang mampu membantu dokter dalam pengambilan keputusan untuk mendiagnosa penyakit berdasarkan gejala-gejala yang muncul

- *Natural Language*

*Natural Language* adalah sebuah AI yang diharapkan dapat mempelajari dan memahami bahasa natural manusia. Misalnya saja *Natural Language* yang dapat mendengarkan dan mengucapkan bahasa inggris.

- *Robotics*

*Robotics* AI adalah AI yang biasa diterapkan pada sebuah robot. Sehingga robot dapat mendengar, melihat atau bereaksi terhadap sensor-sensor lain.

### **2.3. Ellie: Buddy AI in The Last of Us**

Dalam pembuatan game The Last of Us terdapat sebuah Karakter Companion bernama Ellie. Ellie disini akan membantu Joel (karakter protagonist yang akan dimainkan oleh pemain). Dalam game ini pembuat berusaha membuat AI yang dapat membantu pemain bukannya membuat permainan pemain bertambah sulit.

Dalam game ini Ellie dibuat adalah seorang anak kecil yang sedang bertumbuh dan belum pernah melihat tindakan brutal maupun kekerasan, dan baru melihat kota yang hancur seperti pemain baru melihat kota yang hancur. Dan Joel sebagai protagonist melihat Ellie sebagai karakter yang rentan dalam dunia yang berbahaya, sehingga AI diharapkan dapat bertumbuh dan bertambah kuat seiring berjalannya permainan.

#### **2.3.1 Metode**

Adapun sebagai berikut merupakan metode-metode yang digunakan pada Ellie:Buddy AI:

- **Follow Position**

Untuk membuat AI dapat mengikuti pemain maka dibuatlah sistem yang dapat mengevaluasi posisi yang berdekatan dengan pemain, dan kemudian menggerakkan AI dekat dengan pemain. Inti dari sistem ini adalah area yang digunakan dalam mengikuti akan menggunakan sebuah torus di sekitar pemain yang akan dijelaskan menggunakan data yang telah dikumpulkan.

Pertama-tama beberapa kandidat posisi mengikuti akan dibuat di dalam daerah ini dan kemudian posisi ini akan dinilai lagi kualitasnya. Posisi kandidat akan dibuat dengan cara memancarkan 3 buah *navmesh rays*. Seperti yang ada pada Gambar 2.1

Sekumpulan *ray* pertama akan dikeluarkan oleh pemain ke dalam *follow region* agar dapat mengetahui apakah pemain dan AI memiliki ruang jalan yang

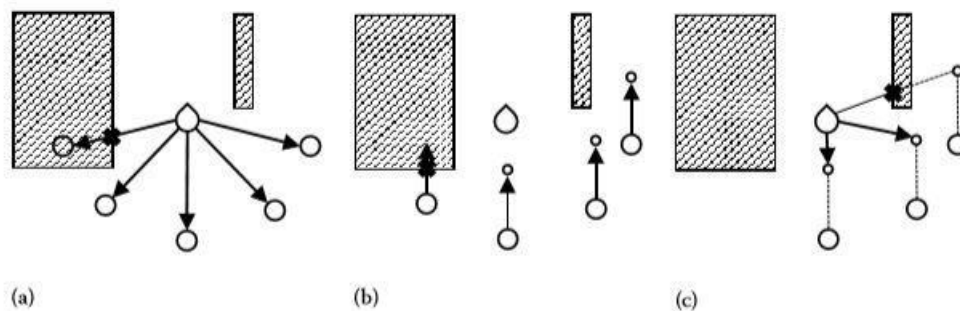
renggang sehingga tidak saling bertabrakan. kandidat posisi akan dibuat berdasarkan setiap ray yang mencapai *follow region*. Seperti pada Gambar 2.1a

Sekumpulan *ray* kedua lalu akan dipancarkan kedepan dari posisi AI untuk memastikan AI tidak menghadap ke dinding (Gambar 2.1b).

Dan yang terakhir adalah ray yang dipancarkan dari lokasi arah depan pemain, untuk memastikan bahwa arah kedepan pemain tidak akan membuat halangan untuk pemain maupun AI .seperti pada Gambar 1.1c .

Hasil dari *follow position* kemudian akan dinilai pada setiap frame. Dari metode ini dapat mendapatkan hasil berupa :

- Jarak antara AI dan pemain
- Tinggal di posisi yang sama dengan pemain
- Tingkat visibilitas terhadap target (tersembunyi atau kelihatan)
- Tidak di depan pemain
- Jarak dengan Companion lain



Gambar 2.1 Pembuatan *follow position* dengan cara membuat *pathfinding raycast*. (a) Memancarkan array untuk mebuat kandidat posisi. (b) Memancarkan array untuk mengecek arah di depan. (c) Memancar ray untuk mengecek posisi berikutnya. Sumber : Game AI Pro, Steve Rabin (2015)

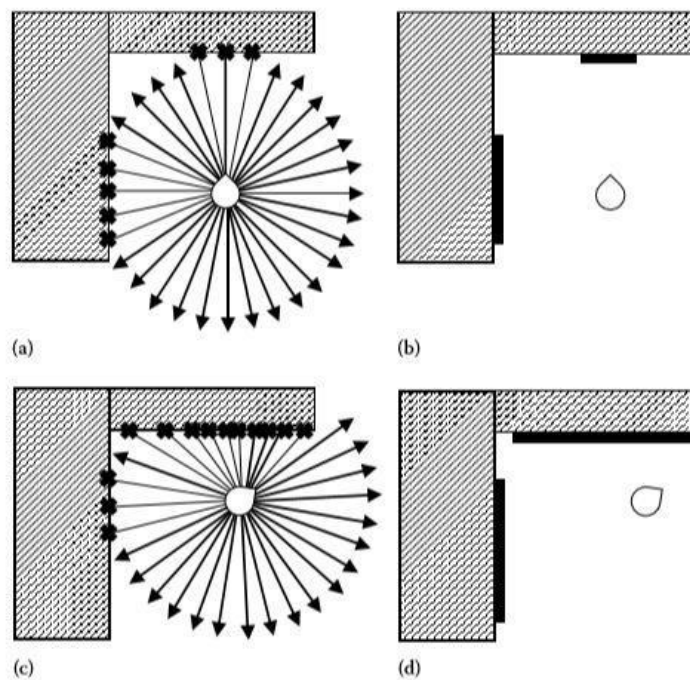
- **Berlindung secara *runtime***

Untuk membuat AI dapat berlindung secara *runtime* atau ketika program sedang berjalan, pertama-tama AI akan memindai lingkungan di sekitarnya dan mencari fitur *cover edge*. Kemudian memancarkan *raycast* dari posisi pemain

terhadap *collision geometry*(seperti pada Gambar 2.2a),membandingkan normal dari *collision plane* dan lokasi dari titik kena, dan kemudian menyatukan titik terdekat dengan fitur *cover edge* terdekat(Gambar 2.2b).

Kedua set fitur *cover edge* satunya bersifat statis dan satunya bersifat *procedural* akan disatukan sebagai bahan yang akan digunakan di masa depan. Setelah itu apabila pemain berubah posisi maka prosedur yang sama akan dilakukan lagi sehingga menciptakan *edges* yang lebih besar(Gambar 2.2c, d). Ada beberapa point yang harus diperhatikan dalam pembuatan fitur ini yaitu :

- Visibilitas dari musuh, dot-product check.
- Kedekatan dengan pemain.
- Memperkirakan visibilitas kedepan, dengan mengukur pergerakan musuh.



Gambar2.2Pembuatan procedural fitur *cover edge* menggunakan raycast. (a) Memancarkan ray untuk mencari kolisi terdekat. (b) Menggabungkan kolisi yang mirip untuk membuat fitur *cover edge*. (c) Menembakkan ray lagi, saat pemain bergerak. (d) Menyatukan titik kolisi dengan *cover edge* sebelumnya untuk memperluas fitur *cover edges*. Sumber : Game AI Pro,Steve Rabin (2015)

- **Menembak**

Dalam game ini bersembunyi adalah point yang penting sehingga apabila AI memberikan posisi kepada musuh akan merusak jalannya permainan. Pertama-tama logika menembak diubah dari sering menembak menjadi hanya menembak apabila diperlukan. AI dibuat hanya akan menembak hanya apabila pemain menembak secara terus-menerus, ataupun apabila AI sedang dalam bahaya. Hal ini dinilai dengan menghitung jarak antara perkiraan letak pemain oleh musuh dan letak sebenarnya apabila jarak sangat jauh, maka pemain dianggap sedang bersembunyi dan AI tidak diperbolehkan untuk menembak

- **Memberi**

Dalam game ini terdapat sebuah *drop system* di mana drop akan muncul sesuai dengan area maupun musuh yang mengdrop. Dalam game ini AI hanya dibatasi dapat memberikan amunisi dan obat kepada pemain, sehingga pemain harus mencari sendiri bahan-bahan yang akan digunakan untuk merakit sesuatu sehingga tidak membuat game terlalu mudah.

Dalam fitur memberi AI akan diberikan beberapa jeda waktu agar tidak membuat pemain terganggu. Dengan fitur ini apabila AI dapat memberi pada waktu yang tepat dapat meningkatkan keterikatan antara pemain dan AI.

## **2.4. Fuzzy Logic**

Fuzzy Logic adalah AI pengembangan dari boolean sehingga hasil akan berupa 0 atau 1. Fuzzy sendiri pertama kali diperkenalkan oleh Dr. Lotfi Zadeh di Universitas California pada tahun 1965. Tetapi Fuzzy sendiri telah dipelajari sejak tahun 1920 secara khusus oleh Łukasiewicz dan Tarski.

### **2.4.1 Proses Logika Fuzzy**

- a. *Fuzzyfy* semua nilai input ke dalam fungsi keanggotaan fuzzy.
- b. Mengeksekusi semua aturan yang berlaku dari *rulebase* untuk menghitung fungsi output fuzzy
- c. *De-fuzzify* fungsi output fuzzy untuk mendapatkan nilai asli fuzzy.

### 2.4.2 Tipe-tipe Fuzzy

a. Tipe Mamdani : fuzzy tipe Mamdani adalah fuzzy yang pertama kali diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Fuzzy Mamdani sendiri berdasarkan oleh *paper fuzzy algorithms for complex systems and decision processes* yang ditulis oleh Lotfi Zadeh pada tahun 1973.

b. Metode Sugeno: fuzzy tipe Sugeno adalah fuzzy yang pertama kali diperkenalkan oleh Takagi Sugeno pada tahun 1985. Proses fuzzy Sugeno hampir sama dengan fuzzy Mamdani, yang membedakan adalah bentuk keluaran dari fuzzy Sugeno berbentuk fungsi keanggotaan linear ataupun konstan. bentuk aturan Fuzzy Sugeno dapat dilihat seperti pada persamaan 2.1

$$\text{If Input 1 is } x \text{ and Input 2 is } y, \text{ then Output is } z = ax + by + c \quad (2.1)$$

Untuk model Sugeno dengan urutan nol, tingkat keluaran  $z$  adalah konstanta ( $a = b = 0$ ). Dengan output level  $z_i$  dan rule weight  $w_i$ . Contoh, untuk aturan AND dengan Input 1 =  $x$  dan Input 2 =  $y$ , aturan yang akan digunakan dapat dilihat seperti pada persamaan (2.2) .

$$w_i = \text{AndMethod}(F_1(x), F_2(y)) \quad (2.2)$$

Dimana  $F_{1,2}(\cdot)$  Adalah fungsi keanggotaan untuk Input 1 dan 2. Adapun hasil akhir dari sistem ini adalah *weight average* dari semua rule yang telah diinputkan, dengan perhitungan seperti pada persamaan 2.3 Dimana  $N$  adalah jumlah rule.

$$\text{Final Output} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i} \quad (2.3)$$

### 2.4.3 Bentuk-bentuk membership logika fuzzy

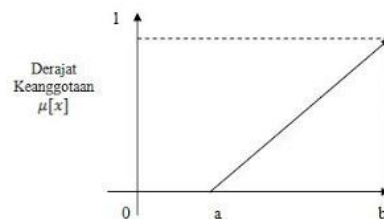
Fungsi Keanggotaan atau membership fuzzy merupakan suatu kurva yang menunjukkan titik input ke dalam nilai keanggotaan. Adapun Kurva yang akan dibahas adalah kurva linear dan kurva segitiga.

### a. Representasi Linear

Pada representasi Linear . pemetaan input ke derajat keanggotaan akan digambarkan sebagai sebuah garis lurus. Representasi Linear terbagi menjadi dua yaitu Representasi linear naik dan representasi linear turun.

- Linear Naik

Pada representasi linear naik nilai keanggotaan yang memiliki derajat keanggotaan 0 akan bergerak ke kanan mengarah kearah keanggotaan yang lebih tinggi. Adapun contoh representasi linear naik dapat dilihat seperti pada gambar 2.3



Gambar 2.3 bentuk representasi dari Kurva Naik

Adapun bentuk persamaan dari kurva ini dapat dilihat pada persamaan 2.4 di mana a merupakan nilai domain yang mempunyai derajat keanggotaan 0, b merupakan n nilai domain yang mempunyai derajat keanggotaan 1, dan x merupakan nilai input.

$$\mu[x, a, b] = \begin{cases} 0; & x \leq a \\ (x - a) / (b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.4)$$

Keterangan

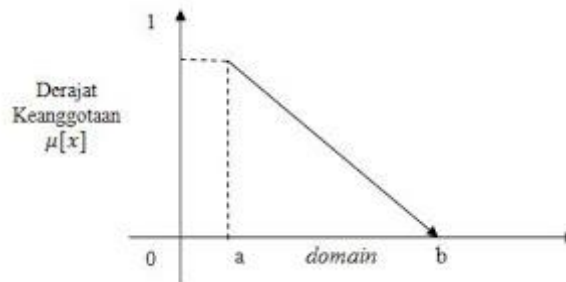
a= nilai domain yang mempunyai derajat keanggotaan 0

b= nilai domain yang mempunyai derajat keanggotaan 1

x= nilai input

- Linear Turun

Pada representasi linear naik nilai keanggotaan yang memiliki derajat keanggotaan 1 akan bergerak ke kiri mengarah ke arah keanggotaan yang lebih rendah. Adapun contoh representasi linear naik dapat dilihat seperti pada gambar 2.4.



Gambar 2.4 bentuk representasi dari Kurva Turun

Adapun bentuk persamaan dari kurva ini dapat dilihat pada persamaan 2.5 di mana a merupakan nilai domain yang mempunyai derajat keanggotaan 1, b merupakan nilai domain yang mempunyai derajat keanggotaan 0, dan x merupakan nilai input.

$$\mu[x, a, b] = \begin{cases} (b-x) / (b-a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.5)$$

**Keterangan**

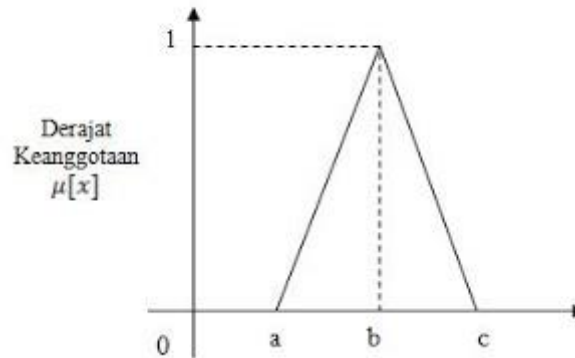
a= nilai domain yang mempunyai derajat keanggotaan 1

b= nilai domain yang mempunyai derajat keanggotaan 0

x= nilai input

## b. Representasi Kurva Segitiga

Representasi Kurva Segitiga, pemetaan input ke derajat keanggotaannya digambarkan dengan bentuk segitiga di mana segitiga merupakan gabungan antara 2 garis linear. Adapun bentuk representasi keanggotaannya dapat dilihat seperti pada gambar 2.5.



Gambar 2.5 Bentuk Representasi dari Kurva Segitiga

Adapun bentuk persamaan dari kurva ini dapat dilihat pada persamaan 2.6, di mana a dan c adalah nilai domain yang memiliki nilai keanggotaan 0, dan b adalah domain dengan nilai keanggotaan 1.

$$\mu [x, a, b, c] = \begin{cases} 0 & x \leq a \text{ atau } x \geq c \\ (x-a) / (b-a); & a \leq x \leq b \\ (c-x) / (c-b); & b \leq x \leq c \end{cases} \quad (2.6)$$

### Keterangan

a= nilai domain yang mempunyai derajat keanggotaan 0

b= nilai domain yang mempunyai derajat keanggotaan 1

c= nilai domain yang mempunyai derajat keanggotaan 0

x= nilai input

## **2.5. Rule Base(Basis Aturan)**

### **2.5.1 Pendahuluan**

Rule Base atau basis aturan adalah suatu metode untuk menyimpan dan memanipulasi pengetahuan atau aturan yang kemudian akan digunakan untuk memperoleh informasi yang dapat membantu menyelesaikan berbagai permasalahan. Pada umumnya *rule base* akan diwujudkan dalam bentuk sebagai berikut

IF sebuah kondisi THEN lakukan sebuah aksi

### **2.5.2 Struktur**

Dalam sebuah rule base system terdapat 4 komponen utama yaitu sebagai berikut

- Sekumpulan aturan atau basis aturan yang akan mengatur dasar pengetahuan.
- Sebuah *inference engine* atau *semantic reasonener* yang akan mengambil keputusan berdasarkan pada aturan-aturan yang telah dibuat. Biasanya berpola pencocokan, penyelesaian masalah, aksi.
- *Temporary Working Memory*.
- *User Interface* yang akan digunakan dalam proses *input output*.

## **2.6. Unity**

### **2.6.1 Pendahuluan**

Unity adalah aplikasi *game engine* lintas *platform* yang saat ini tengah dikembangkan oleh *Unity Technologies* yang dapat digunakan untuk membuat sebuah video game berbasis *PC*, *mobile*, *website* ataupun *console*. Unity pertama kali dikembangkan dan diluncurkan hanya untuk system operasi *OS X*, tetapi setelah konferensi *Apple's Worldwide Developer Confrence* pada tahun 2005, Unity dapat digunakan untuk 21 *Platform* lainnya.

## 2.6.2 Fitur-fitur Unity

Berikut merupakan Tabel fitur yang terdapat pada unity dapat dilihat pada Tabel 2.1.

Tabel 2.1 Fitur *Unity*

| Feature             | Personal            | Plus                     | Pro                      | Enterprise               |
|---------------------|---------------------|--------------------------|--------------------------|--------------------------|
| All Engine Features | Yes                 | YES                      | Yes                      | Yes                      |
| All Platforms       | Yes                 | YES                      | Yes                      | Yes                      |
| Continuous Updates  | Yes                 | YES                      | Yes                      | Yes                      |
| Royalty Free        | Yes                 | YES                      | Yes                      | Yes                      |
| Splash Screen       | MWU Splash Screen   | Custom Animation or None | Custom Animation or None | Custom Animation or None |
| Revenue Capacity    | \$100k              | \$200k                   | Unlimited                | Unlimited                |
| Unity Analytics     | Personal Analytics  | Plus Analytics           | Pro Analytics            | Custom Analytics         |
| Unity Cloud Build   | Standard Queue      | Priority Queue           | Concurrent Builds        | Dedicated Build Agents   |
| Unity Multiplayer   | 20 Concurrent Users | 50 Concurrent Users      | 200 Concurrent Users     | Custom Multiplayer       |
| Unity Ads           | Yes                 | YES                      | Yes                      | Yes                      |
| Beta Access         | Yes                 | YES                      | Yes                      | Yes                      |
| Pro Editor UI Skin  |                     | YES                      | Yes                      | Yes                      |