

BAB 2 LANDASAN TEORI

2.1 Muay Thai

Muai Thai atau yang biasa disebut *Thai Boxing*, merupakan salah satu seni bela diri atau *combat sport* yang menggunakan hampir semua bagian tubuh dengan menghasilkan 8 kontak tubuh (tinju, siku, lutut, dan kaki) atau yang bisa disebut *Art of 8 limbs*. Muay Thai digunakan untuk melindungi diri. Selain meningkatkan kesehatan tubuh, Muay Thai juga berpengaruh terhadap kesehatan mental, Muay Thai dapat melatih kesabaran serta keberanian bagi orang yang mempelajari bela diri Muay Thai [10]. Dalam melatih teknik dasar Muay Thai, terdapat beberapa jenis pukulan yang dapat dipelajari, yaitu [11]:

1. *Jab*

Jab merupakan pukulan yang dilakukan dengan cara meninju dengan kurus ke arah wajah atau bagian lain dari tubuh lawan.

2. *Cross / Straight*

Cross / Straight ini merupakan pukulan yang dapat digunakan untuk menjatuhkan lawan. Sama seperti pukulan jab, teknik pukulan ini juga memukul kedepan, namun dengan tenaga yang lebih besar.

3. *Hook*

Hook merupakan tehnik pukulan yang memukul musuh dari samping. Teknik pukulan ini dilakukan dengan membuat serangan seperti *hook* atau pengkait.

4. *Uppercut*

Uppercut ini merupakan pukulan dari bawah yang menargetkan bagian dagu dari lawan.

Selain teknik pukulan, terdapat juga teknik tendangan yang terdapat dalam bela diri Muay Thai, diantaranya adalah sebagai berikut.

1. *High Kick*

High Kick merupakan teknik tendangan yang digunakan untuk menyerang bagian kepala dari lawan.

2. *Low Kick*

Low Kick ini merupakan tendangan yang dilakukan dengan menargetkan bagian pinggang ke bawah dari lawan.

3. *Middle Kick*

Middle Kick merupakan tehnik teknik tendangan yang dilakukan untuk meny-erang bagian pinggang ke atas dari lawan.

4. *Push Kick*

Push Kick merupakan tendangan yang dilakukan untuk menjatuhkan lawan ataupun untuk mengambil jarak dengan lawan dengan cara mendorong lawan.

Lalu terdapat juga teknik menyikut yang terdapat pada bela diri Muay Thai, yang terdiri atas 5 teknik, yaitu.

1. Sikutan Vertikal

Sikutan vertikal merupakan teknik menyikut lawan yang dilakukan dengan gerakan menyikut dari atas ke bawah.

2. Sikutan Horizontal

Sikutan Horizontal merupakan teknik menyikut yang sama seperti sikutan vertikal, namun teknik sikutan horizontal dilakukan dengan cara menyikut dari samping.

3. *Flying Smashing Elbow*

Flying smashing elbow merupakan teknik sikutan yang dilakukan dengan cara melakukan lompatan terlebih dahulu sebelum melakukan teknik sikutan terhadap lawan.

4. *Uppercut Elbow*

Uppercut elbow merupakan sikutan yang dilakukan dengan cara menyikut dari bawah ke atas.

5. *Smashing Elbow*

Smashing elbow merupakan teknik sikutan yang sama seperti sikutan vertikal, namun sikutan ini menggunakan tenaga yang lebih kuat.

Teknik lainnya yaitu teknik mendengkul yang merupakan gerakan yang di-lakukan menggunakan dengkul. Berikut merupakan teknik mendengkul yang ter-dapat pada Muay Thai.

1. *Hook Knee*

Hook Knee merupakan teknik mendengkul yang dilakukan dengan cara mendengkul lawan pada bagian samping dari tubuh lawan.

2. *Uppercut Knee*

Uppercut Knee merupakan teknik mendengkul lawan yang dilakukan dengan cara mengambil ancang-ancang terlebih dahulu sebelum melakukan gerakan mendengkul dengan menargetkan bagian dagu lawan.

3. *Circle Knee*

Circle Knee merupakan teknik mendengkul yang dilakukan dengan cara memutar seluruh anggota tubuh dengan tenaga yang berpusat pada area lutut atau dengkul, yang dilanjutkan dengan serangan pada tubuh bagian samping lawan. Teknik mendengkul ini menargetkan bagian pada rusuk maupun bagian paha lawan.

2.2 Game

Game secara umum merupakan sebuah bentuk permainan atau olahraga terutama dalam segi kompetitif yang dimainkan berdasarkan aturan yang telah ditentukan yang memerlukan keterampilan, kekuatan atau keberuntungan dalam memenangkan kompetisi tersebut. *Game* terkait dengan imajinasi yang menyenangkan, sehingga memungkinkan pemain untuk bereksperimen dengan skenario kehidupan nyata dengan pemain yang lain, serta mampu meningkatkan pemahaman kognitif dan motivasi intrinsik dalam memainkan suatu *game* [12]. Selama lima dekade terakhir, *game* berkembang terus berkembang namun dalam bentuk *video game* yang dibangun dengan mengimplementasikan game kedalam ponsel maupun konsol dan pc. *Video game* berbentuk interaksi antara pemain dengan mesin yang dihubungkan dengan tampilan visual menggunakan media elektronik yang dimediasi oleh konteks fiksi, serta didukung oleh keterikatan emosional antara pemain dengan hasil tindakan yang dilakukan dalam *video game* yang dimainkan [13].

Seluruh tipe *game* yang berbeda selalu memiliki elemen dasar yang sama, elemen dasar tersebut yang membangun *game* serta memberikan pengalaman tertentu saat pemain memainkan *game* tersebut. Elemen dasar ini biasa disebut sebagai *formal elements* dari suatu permainan, *game* tidak dapat disebut sebagai *game* bila tidak memiliki *formal elements* ini. Sama halnya seperti sebuah *game* tanpa adanya pemain, tujuan, peraturan serta prosedur, maka *game* ini tidak seperti *game* sama

sekali. *Formal elements* tersebut terdiri dari [14]:

1. *Players*

Player atau Pemain merupakan orang yang berinteraksi dengan sistem dalam suatu *game*. Dalam suatu *game*, rancangan yang dibuat untuk memberikan pengalaman ditujukan kepada pemain yang akan memainkannya, selain itu pemain tersebut juga secara sukarela menerima peraturan serta batasan yang ada dalam suatu *game* agar *game* tersebut dapat dimainkan. Dalam membuat suatu *game*, perlu diperhatikan jumlah pemain yang dapat memainkan *game* tersebut, karena desain *game* memiliki pertimbangan yang berbeda berdasarkan jumlah pemain yang dapat memainkan *game* tersebut secara bersamaan, seperti permainan tic-tac-toe yang memerlukan 2 pemain tidak lebih dan tidak kurang.

2. *Objectives*

Objectives atau yang biasa disebut sebagai tujuan, merupakan sesuatu yang harus diperjuangkan. Dalam *game* tujuan dijelaskan sebagai capaian yang akan dicapai dalam aturan permainan, tujuan tersebut dibuat agar menambahkan tantangan dalam memainkan suatu *game* namun tujuan tersebut harus dapat dicapai oleh pemain. Tujuan dalam *game* dapat memberikan suasana yang berbeda dalam setiap *game*, seperti *game* yang memiliki tujuan untuk menangkap akan berbeda dengan *game* yang memiliki tujuan untuk menumbuhkan tanaman. Pemilihan tujuan dalam pembuatan *game* harus dilakukan secara hati-hati karena dapat berdampak pada *formal elements* dan *dramatic aspects*.

3. *Procedures*

Procedures atau prosedur merupakan metode bermain dan aktivitas yang dapat dilakukan oleh pemain untuk mencapai tujuan dalam suatu *game*. Prosedur dari *game* digunakan agar tetap menjaga batasan dari environment yang ada pada *game* yang dibuat. Prosedur dapat disebut sebagai "*core loop*", *core loop* ini merupakan kumpulan dari prosedur yang dilakukan secara berulang saat *game* yang dimainkan, seperti menggerakkan dadu atau saat menentukan kondisi dalam suatu *game*. misalnya saat pemain menabrak sesuatu. Dalam desain *game*, diperlukan kemampuan untuk membuat prosedur yang intuitif untuk diakses dan mudah diingat oleh pemain.

4. *Rules*

Rules atau yang biasa disebut sebagai aturan menentukan permainan serta tindakan yang diperbolehkan oleh pemain dalam memainkan suatu *game*. Penentuan aturan juga disesuaikan dengan *environment* yang terdapat desain *game* yang dibuat. Pembuatan aturan perlu dibuat secara jelas agar *game* yang dibuat terlihat adil dan mampu responsif terhadap situasi yang ada. Pembuatan aturan juga perlu mudah dipahami oleh pemain agar pemain dapat mengerti makna dari *game* yang dibuat dan masuk akal.

5. *Resources*

Resources atau sumber daya dalam desain *game* merupakan aset yang dapat digunakan oleh pemain untuk mencapai tujuan suatu *game*. Sumber daya dalam *game* harus memiliki nilai kegunaan dan kelangkaan dalam sistem permainan, sehingga sumber daya harus dikelola dan ditentukan bagaimana serta kapan akan digunakan dalam suatu permainan, misalnya *lives, units, health, currency, actions, power-ups, inventory*, dan *time* dalam suatu permainan.

6. *Conflict*

Conflict atau yang disebut sebagai konflik maupun permasalahan yang muncul saat pemain mencoba untuk menyelesaikan *game* yang terikat dengan peraturan dan batasan yang telah dibuat oleh perancang *game*. Konflik dalam *game* dapat berupa *obstacles, opponents*, maupun *dilemmas*.

7. *Boundaries*

Boundaries atau batasan, merupakan penentuan bagaimana *game* dapat dimainkan, menerima aturan yang terdapat dalam *game*, serta kapan *game* tersebut dapat dimulai, berakhir atau pemain dapat berhenti atau keluar jika pemain tidak ingin memainkan *game* itu lagi. Batasan ini digunakan untuk membangun pengalaman bermain dari seorang pemain.

8. *Outcome*

Outcome atau yang biasa disebut sebagai hasil dibentuk untuk menarik perhatian pemain untuk memainkan suatu *game*, secara umum hasil yang didapat dapat diselesaikan dan tidak seimbang, dalam artian ada yang menang dan ada yang kalah. *Outcome* perlu dipikirkan dengan baik oleh perancang *game* agar mengetahui bagaimana menciptakan penyelesaian *game* yang mampu memuaskan pemain berdasarkan usaha dalam menyelesaikan *game*.

2.3 Virtual Reality

Virtual reality mengacu pada simulasi yang dihasilkan komputer dimana seseorang dapat berinteraksi langsung didalam lingkungan tiga dimensi buatan menggunakan perangkat elektronik, seperti kacamata khusus dengan layar atau sarung tangan yang dilengkapi dengan sensor, sehingga pengguna dapat mengalami pengalaman secara realistis. Selain itu didapatkan bahwa 3 fitur dasar yang dimiliki oleh *virtual reality* adalah *immersion*, *perception* didalam suatu *environment* serta *interaction* dengan *environment* tersebut [15]. Dalam membangun pengalaman *virtual reality* diperlukan komponen-komponen yang membangun pendukung yang membentuk dunia *virtual reality* agar pengguna dapat mendapatkan pengalaman yang sepenuhnya imersif. Komponen-komponen tersebut adalah [16]:

1. *Stereoscopic Displays*

Stereoscopic Displays atau yang biasa dikenal sebagai *Head Mounted Device* (HMD), digunakan untuk menghasilkan gambar stereo yang membuat mata orang yang menggunakannya melihat interpretasi dari dunia tiga dimensi.

2. *Motion Tracking Hardware*

Motion Tracking Hardware ini digunakan untuk menerima pergerakan tubuh dan kepala dari gerakan pengguna sehingga aplikasi yang digunakan mampu terus memperbarui *view* atau penglihatan didalam skema 3D.

3. *Input Devices*

Input devices digunakan agar dapat mengenali pergerakan dan gestur yang dilakukan oleh pengguna. Namun *input devices* yang digunakan harus memiliki *game controller* dan sensor pelacak tubuh, tidak seperti pada *game* biasanya yang hanya menggunakan mouse dan keyboard untuk menerima input dari pengguna.

4. Desktop and Mobile Platforms

Desktop and Mobile Platforms ini merupakan perangkat yang dibutuhkan untuk membangun serta menjalankan aplikasi yang dibuat, seperti *software tools* yang digunakan untuk membangun aplikasi dan seperangkat komputer yang digunakan untuk menjalankan aplikasi tersebut.

2.4 Algoritma Sattolo Shuffle

Berdasarkan penelitian [9], algoritma *sattolo shuffle* merupakan modifikasi dari Algoritma *fisher-yates shuffle* oleh Sandra Sattolo pada tahun 1986, algoritma ini melakukan permutasi acak dari sebuah himpunan berhingga. Hasil modifikasi yang dilakukan menghasilkan permutasi yang berasal dari satu siklus secara lebih optimal. Sama seperti algoritma *fisher-yates shuffle*, algoritma *sattolo shuffle* juga menghasilkan permutasi yang tidak bias, yaitu tidak memiliki permutasi yang sama dengan hasil sebelumnya. Selain itu konsep yang sederhana serta pengacakan yang dilakukan pada data yang sama sehingga dapat menghemat penggunaan *resources*. Berikut merupakan langkah-langkah dari metode dasar algoritma *sattolo shuffle*.

- (a) Masukkan kumpulan data yang akan diacak ke dalam *array*.
- (b) Simpan panjang *array* dari kumpulan data kemudian masukkan ke dalam bilangan i .
- (c) Pilih bilangan acak r antara bilangan 0 dan $(i - 1)$ yang akan digunakan untuk menentukan indeks *array*.
- (d) Tukarkan elemen yang terdapat pada *array* indeks ke- r dengan elemen yang terdapat pada *array* indeks ke- $(i - 1)$.
- (e) Kurangi bilangan i dengan 1 dan lakukan pengecekan apakah bilangan i lebih besar dari 1 atau tidak, jika bilangan i lebih besar dari 1 maka ulangi langkah ketiga sampai bilangan i lebih kecil dari 1.

2.5 Game User Experience Satisfaction Scale (GUESS)

Game User Experience Satisfaction Scale (GUESS) merupakan instrumen yang digunakan untuk mengukur tingkat kepuasan *video game* yang diukur menggunakan 5 proses, yang mengukur 9 faktor tingkat kepuasan *video game* yang terdiri atas *Usability/Playability*, *Narratives*, *Play Engrossment*, *Enjoyment*, *Creative Freedom*, *Audio Aesthetics*, *Personal Gratification*, *Social Connectivity*, dan *Visual Aesthetics*. GUESS menggunakan 55 pertanyaan yang mencakup *7-point rating scale* untuk mengukur tingkat kepuasan yang terdiri atas [17]:

- (a) Sangat tidak setuju dengan nilai 1.
- (b) Tidak setuju dengan nilai 2.

- (c) Agak tidak setuju dengan nilai 3.
- (d) Biasa saja dengan nilai 4.
- (e) Agak setuju dengan nilai 5.
- (f) Setuju dengan nilai 6.
- (g) Sangat setuju dengan nilai 7.

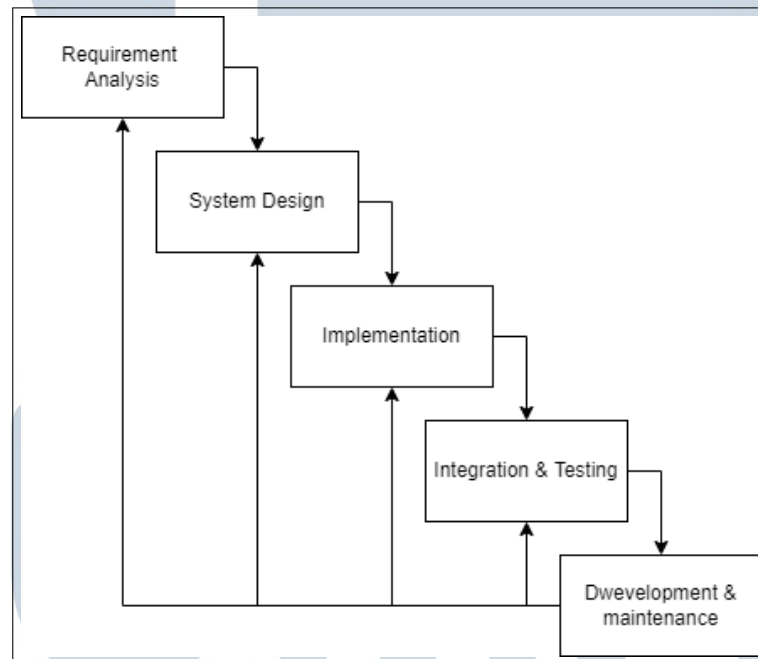
Setelah itu didapatkan penilaian menggunakan GUESS dengan menghitung rata-rata dari seluruh *rating scale* yang diperoleh, dengan menghitung keseluruhan *point* yang didapatkan lalu dijumlahkan untuk mendapatkan *point* total, kemudian dibagi dengan total maksimal yang didapat dari jumlah responden dikalikan dengan *point* tertinggi. Nilai yang didapatkan tersebut dibagi menjadi beberapa tingkat kepuasan, yaitu:

- (a) Sangat tidak puas dengan nilai 0% sampai 14%.
- (b) Tidak puas dengan nilai 15% sampai 28%.
- (c) Agak tidak puas dengan nilai 29% sampai 42%.
- (d) Biasa saja dengan nilai 43% sampai 56%.
- (e) Agak puas dengan nilai 57% sampai 70%.
- (f) Puas dengan nilai 71% sampai 84%.
- (g) Sangat puas dengan nilai diatas 84%.

Namun, dikarenakan GUESS menggunakan 55 pertanyaan yang digunakan untuk mengukur tingkat kepuasan *video game* memerlukan waktu pengisian 10 sampai 15 menit. Maka diciptakan GUESS-18 yang hanya menggunakan 18 pertanyaan untuk mengukur tingkat kepuasan *video game* yang hanya memerlukan waktu pengisian 3 sampai 5 menit untuk mempermudah responden dalam menjawab seluruh pertanyaan yang diberikan [18].

2.6 Waterfall Model

Waterfall model merupakan salah satu *Software Development Life Cycle* (SDLC) yang paling sering digunakan dalam pengembangan sistem. *Waterfall model* merupakan model yang cocok digunakan dalam membangun sistem dengan alur yang jelas, hal ini dikarenakan *waterfall model* menghasilkan rincian kegiatan dari suatu proyek yang sedang dikerjakan ke dalam kegiatan linier, sehingga setiap fase bergantung pada hasil yang dicapai pada fase sebelumnya [19]. Model ini cenderung menjadi salah satu pendekatan yang fleksibel karena dijalankan dengan cara berurutan seperti air terjun yang dibagi menjadi beberapa fase, yaitu *requirement analysis*, *system design*, *implementation*, *integration & testing*, dan *operation & maintenance* yang terlihat seperti pada Gambar 2.1.



Gambar 2.1. *Waterfall Model*

2.7 Black Box Testing

Black box testing atau dapat disebut *Behaviorial Testing* merupakan metode pengujian terhadap fungsionalitas dari suatu aplikasi dengan melihat hasil input dan output yang dihasilkan. Pengujian ini dapat dilakukan tanpa mengetahui kode yang terdapat pada aplikasi yang diuji. Terdapat 3 tipe *black box testing*, yaitu [20]:

1. Functional Testing

Merupakan pengujian yang dilakukan dengan menguji fungsi tertentu pada perangkat yang diuji. Functional testing berfokus pada aspek utama, integrasi antar komponen maupun sistem secara keseluruhan pada suatu aplikasi.

2. Non-functional Testing

Merupakan pengujian yang dilakukan pada aspek tambahan pada aplikasi yang berada diluar fitur dan fungsionalitas, seperti pengujian terhadap kemudahan dalam memahami aplikasi, kinerja aplikasi saat dibawah beban yang diharapkan atau beban puncak, kompatibilitas terhadap seluruh perangkat, ukuran layar, browser maupun sistem operasi, serta tingkat keamanan aplikasi.

3. Regression Testing

Merupakan pengujian yang dilakukan untuk menguji apakah aplikasi terbaru yang dikembangkan mengalami kemunduran, atau penurunan kemampuan dari versi sebelumnya. Pengujian regresi dilakukan pada aspek fungsional maupun non-fungsional pada aplikasi dengan versi terbaru.

