BAB 1
PENDAHULUAN

1.1. Latar Belakang Masalah

Metode *forward chaining* adalah algoritma yang dititikberatkan pada pendekatan yang berdasarkan data atau fakta. Metode *forward chaining* merupakan stategi pencarian yang dimulai dari informasi masukan seperti sekumpulan data atau fakta, dari informasi masukan tersebut dicari suatu kesimpulan yang menjadi solusi dari permasalahan yang dihadapi, sehingga metode ini juga sering disebut "*Data driven*". Suatu perkalian inferensi yang menghubungkan suatu permasalahan dengan solusinya disebut dengan rantai (*Chain*). Suatu rantai yang dicari atau dilewati/dilintasi dari suatu permasalahan untuk memperoleh solusinya disebut dengan *forward chaining* [1].

Metode *forward chaining* pada umumnya digunakan untuk sistem pendukung keputusan dan sistem pakar. Metode *forward chaining* memiliki kelebihan yaitu data baru dapat dimasukan ke dalam tabel database inferensi dan memungkinkan untuk melakukan perubahan *inference rule* [2] . Metode *forward chaining* memiliki karakteristik perencanaan, *monitoring*, *control*, disajikan untuk masa depan, data memandu, penalaran dari bawah ke atas, bekerja ke depan untuk mendapatkan solusi apa yang mengikuti fakta, dan *breadth first search* dimudahkan [3].

Hasil dari studi literatur mengenai metode *forward chaining,* ditemukan contoh kasus yang diselesaikan dengan metode *forward chaining* memiliki kesamaan dengan kasus penentuan status karakter pada *game* petualangan. Jurnal yang berjudul "*Game Chicken Roll* dengan Menggunakan Metode *Forward Chaining*" membahas tentang implementasi metode *forward chaining* terhadap *game chicken roll* yang digunakan untuk proses *review* dan perpindahan *level*, hasil pengujian menunjukan nilai validasi sebesar 100% yang didapat dari komparasi data antara *rules* yang sudah dimasukan dalam metode *forward chaining* pada saat bermain *game*. *Game* petualangan yang ada selama ini menitikberatkan permainan dimana pemain berperan sebagai karakter protagonis

dalam sebuah cerita interaktif didorong oleh eksplorasi dan menyelesaikan berbagai tantangan, sehingga terjadi berbagai interaksi dan kondisi yang dialami karakter pada saat *game* dimainkan, sedangkan informasi mengenai kondisi permainan khususnya status karakter pada *game* petualangan sangatlah kurang bahkan tidak ada sama sekali.

Berdasarkan uraian yang telah dipaparkan, dan berdasarkan hasil studi literatur, maka dilakukan implementasi metode *forward chaining* untuk penentuan status karakter. Penerapan *rule* dari metode *forward chaining* akan menjadi penentuan status karakter pada saat bermain *game*.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, maka perumusan masalah dari penelitian ini adalah bagaimana mengimplementasikan metode *forward chaining* untuk penentuan status karakter pada *game* petualangan.

1.3. Maksud dan Tujuan

Berdasarkan masalah yang telah diteliti, maka maksud dari penelitian ini adalah untuk mengimplementasikan metode *forward chaining* untuk penentuan status karakter pada *game* petualangan. Tujuan yang ingin dicapai dari implementasi metode *forward chaining* ini yaitu menampilkan informasi karakter pada *game* petualangan.

1.4. Batasan Masalah

Batasan masalah dari implementasi metode *forward chainig* untuk penentuan status karakter pada *game* petualangan adalah sebagai berikut :

1. Status karakter berupa informasi karakter.
2. Informasi karakter terletak pada *HUD* (*Head Up Display*).
3. Informasi karakter mencakup *HP, EXP, LVL, SKILL*
4. *HP* (*Hit Point*), status yang menunjukan jumlah maksimal *damange* yang dapat diterima oleh karakter.

5. *EXP* (*Experience*), status yang menunjukan jumlah poin yang digunakan untuk kenaikan *level* karakter.
6. *LVL* (*Level*), status yang menunjukan tingkatan karakter untuk membuka *skill* baru.
7. *SKILL*, status yang muncul pada saat karakter naik level.
8. Data masukan berupa *event*/*collision* pada *game*.
9. Data keluaran berupa informasi karakter.
10. *Rule forward chaining* mulai bekerja dengan mencocokan antara *knowledge base* yang ada dengan *working memory* yang berisi sebuah fakta.
11. Proses penentuan status karakter dihasilkan berdasarkan komparasi data masukan yang dilakukan terhadap fakta.
12. *Genre game RPG* (*Role Playing Game*).
13. Aplikasi *game* yang dibangun berupa simulator.
14. Pembangunan *simulator* menggunakan *blender game engine*.
15. Pendekatan analisis berdasasarkan data terstruktur, yaitu menggunakan *Flowchart* dan DFD

1.5. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam mengerjakan penelitian ini adalah metode deskriptif, yaitu metode penelitian yang menggambarkan secara sistematika dan akurat mengenai keadaan-keadaan nyata yang berjalan pada saat penelitian. Gambaran tersebut diperoleh dengan cara mengumpulkan, mengklasifikasikan, menyajikan, serta menganalisis data sehingga dapat ditarik suatu kesimpulan. Metode ini dibagi menjadi dua, yaitu pengumpulan data dan pembangunan perangkat lunak.

1.5.1. Metode Pengumpulan Data

Metode pengumpulan data dengan cara mencari studi literatur, yaitu penggumpulan data dilakukan dengan cara mempelajari, meneliti, dan menelaah berbagai literatur yang bersumber dari buku, jurnal, *paper*, *internet* dan bacaan-bacaan lainnya yang berkaitan dengan judul penelitian.

1.5.2. Metode Pembangunan Perangkat Lunak

Metode pengembangan perangkat lunak ilustrasi model waterfall. Model ilustrasi waterfall adalah model sistematik untuk pembangunan perangkat lunak yang mempunyai tahapan-tahapan dalam proses pembangunan perangkat lunaknya mulai dari proses analisis, design, code, sampai proses testing [4]. Gambar 1.1 merupakan gambar model ilustrasi waterfall.

1. Tahap analisis

Tahap analisis mulai mencari fungsionalitas-fungsionalitas yang dibutuhkan untuk membangun *game* yang dalam hal ini meliputi analisis *game*, analisis algoritma forward chaining, analisis masalah, analisis masukan serta analisis kebutuhan fungsional dan nonfungsional.

2. Tahap *design*

Pada tahap ini melakukan perancangan dari game yang dibuat meliputi perancangan konten *game*, perancangan struktur menu, perancangan antarmuka, jaringan semantik, perancangan prosedural, perancangan sistem, serta mendesain.

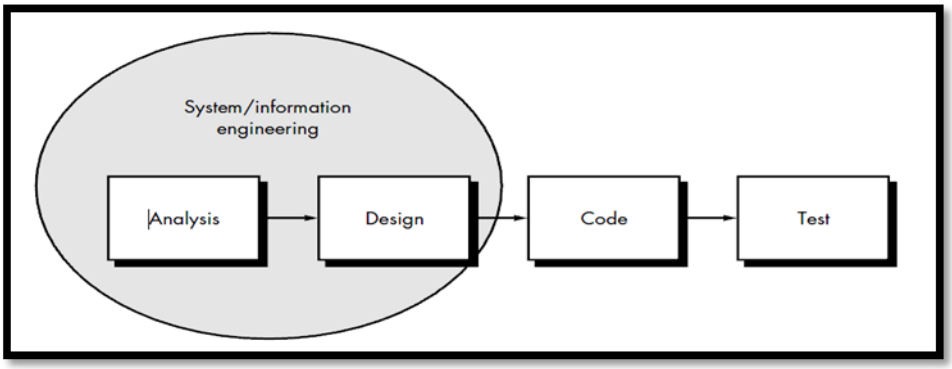
3. Tahap *code*

Tahap code yaitu mulai mengeksekusi perancangan-perancangan yang dilakukan sebelumnya ke dalam bahasa pemrograman *python.*

4. Tahap *testing*

Tahap testing bertujuan untuk menemukan kesalahan-kesalahan terhadap perangkat lunak untuk kemudian bisa diperbaiki.

Gambar model *waterfall* dapat dilihat pada Gambar 1.1.



Gambar 1.1 Ilustrasi Model Waterfall [4]

1.6. Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran umum tentang penelitian yang akan dijalankan. Sistematika penulisan tugas akhir ini sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini menguraikan dasar-dasar pemikiran untuk implementasi metode *forward chaining* pada *game* petualangan yang meliputi latar belakang permasalahan, merumuskan inti permasalahan, menentukan maksud dan tujuan penelitian yang harus dicapai, batasan masalah untuk memfokuskan pembuatan sistem ini, metode penelitian untuk mengumpulkan data, metode yang digunakan dalam pembangunan sistem serta sistematika penulisan laporan penelitian.

BAB 2 LANDASAN TEORI

Bab ini menguraikan tentang teori-teori yang relevan dengan kajian penelitian dan hal-hal lainnya yang mendukung penelitian serta berguna dalam implementasi metode *forward chaining* untuk penentuan status karakter pada *game* petualangan.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini menguraikan analisis domain masalah, analisis *game*, analisis masukan, analisis metode/algoritma dan analisis kebutuhan perangkat lunak.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini menguraikan hasil implementasi analisis dan kebutuhan algoritma yang telah dibuat kedalam bentuk simulator berupa *game*, kemudian dilakukan pengujian terhadap simulator berupa *game* yang telah dibangun.

BAB 5 KESIMPULAN DAN SARAN

Bab ini menguraikan kesimpulan hasil implementasi dan pengujian yang telah dibuat, serta saran-saran untuk pengembangan penelitian ini selanjutnya.