**PENERAPAN METODE FORWARD CHAINING PADA INVENTORY DI GAME ” ANA CERITA ” MENGGUNAKAN UNITY GAME ENGINE**

**Angga Bayu Pratama1, Wahyudi Harianto2, Amak Yunus3**

Program Studi Teknik Informatika – Fakultas Sains dan Teknologi

Universitas Kanjuruhan Malang

anggapretama12@gmail.com

**ABSTRAK**

Penggunaan metode forward chaining sangat mempengaruhi akan kelancaran dalam pembuatan sebuah game, dalam metode forward chaining sendiri akan menyimpulkan tindakan player guna untuk mendapat hasil yang baik di akhir suatu permainan atau game. Dalam inventory game ana cerita sendiri pemanfaatan metode forward chaining yaitu dari segi perhitungan kapasitas yang akan menentukan hasil dari pemanfaatan player terhadap inventory, selain itu deteksi suatu item yang di gunakan player juga akan mempengaruhi layak tidaknya item tersebut di input. peniliti melakukan 11 pengujian dengan 7 pengujian inventory skill dan 4 pengujian inventory relief, dengan mendapatkan 9 nilai validasi yang membuat penerapan forward chaining berhasil. Maka disimpulkan bahwa teknik forward chaining dapat di terapkan dalam pembuatan game terutama pada inventory, dengan menghitung item-item yang akan di input. Besar harapan bahwa penelitian ini dapat membantu pengembangan game digital dan penelitian lain dimasa mendatang.

**Kata Kunci :**  *Forward Chaining, Inventory skill, inventory relief, programmer development, kapasitas input item.*

1. **Pendahuluan**

Pada umumnya, inventory dalam Bahasa Indonesia ialah persediaan yang berupa barang yang akan di pergunakan untuk situasi yang akan di butuhkan. Di dalam game sendiri sama seperti halnya dalam di dunia nyata tetapi barang yang di dalam game berupa digital, dimana player akan mengumpulkannya atau mendapatkannya dalam perjalannya dalam melewati setiap tahap perjalannya yang akan di kumpulkan dalam inventory.

Dalam sebuah game RPG memiliki sebuah inventory yang dibutuhkan player untuk mendapatkan atau mengumpulkan sebuah item yang dibutuhkan player. Dalam game ini pengumpulan item atau kebutuhan player di kumpulkan dalam sebuah inventory, dimana kelengkapan semua item sesuai kebutuhan player.

Metode forward chaining merupakan sebuah penalaran yang di mulai dari sebuah fakta untuk mendapatkan kesimpulan (conclusion) dari fakta tersebut. Metode ini di butuhkan untuk pengambilan keputusan suatu player saat mendapatkan item atau kebutuhan lainnya.

Metode forward chaining sendiri memiliki kelebihan yang dapat membantu peneliti dalam memmbuat tugas penelitian, kelebihan metode fotward chaining yaitu sebagai berikut :

1. Kelebihan utama dari forward chaining yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan atau menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat di ambil dari informasi tersebut.

2. Metode ini mampu banyak sekali menyediakan informasi dari hanya jumlah kecil data.

Untuk membuat game, di perlukan aplikasi game engine untuk membuatnya. Unity merupakan salah satu software game engine yang didalamnya bisa menerapkan teknologi desain 2D atau 3D untuk mengembangkan pembuatan game. Game itu sendiri bermacam-macam jenisnya, salah satu game yang di gemari remaja adalah game RPG. Berdasarkan penjelasan tersebut penulis bermaksud membuat game role playing game ( RPG ) dengan menggunakan metode forward chaining pada inventory di dalam game tersebut.

Dari uraian tersebut Penulis akan terapkan pada skripsi dengan judul ” Penerapan metode forward chaining pada inventory di game ”Ana Cerita” menggunakan Unity game engine.

1. **Tinjauan Pustaka**
   1. **Pengertian Game**

Game merupakan aktifitas terstruktur atau semi terstruktur yang biasanya bertujuan untuk hiburan dan kadang dapat digunakan sebagai sarana pendidikan. Karakterisitik game yang menyenangkan, memotivasi, membuat kecanduan dan kolaboratif membuat aktifitas ini digemari oleh banyak orang.

Pada awal kemunculan game pertama kalinya, game masih disajikan secara sederhana dan di prakarsai oleh Steven Russel dan proyek yang bernama Computer Games pada tahun 1962 dengan produk andalannya bernama Star Wars. Beberapa puluh tahun kemudian, banyak game bermunculan dari game 2 dimensi dan game 3 dimensi. ( Wahyu Pratama, 2014 )

* 1. **Pemrograman Visual C#**

C# adalah bahasa pemrograman baru yang diciptakan oleh Microsoft (dikembangkan di bawah kepemimpinan Anders Hejlsberg yang notabene juga telah menciptakan berbagai macam bahasa pemrograman termasuk Borland Turbo C++ dan Borland Delphi). Bahasa C# juga telah distandarisasi secara *international* oleh ECMA.

Bahasa C# adalah gabungan yang kuat antara bahasa C++ dan Java, tidak heran jika bahasa C# memiliki kemiripan diantara C++ dan Java. ( Rachmat Adi Purnama dan Adi Tri Laksono Putra, 2018 )

* 1. **Inventory**

Salah satu fungsi utama dan fitur dari sebuah *game* RPG adalah *inventory system*. Sepanjang permainan , pengguna akan menemukan sejumlah besar barang-barang yang dapat digunakan untuk membantu mereka selama perjalanan. karena itu *game* perlu menyediakan mekanik untuk membantu pemain dalam mengambil, dan mengatur barang atau *item* untuk perjalanan *player*.

Inventory dalam *game* ini sendiri sangat berguna untuk menampung atau membawa *item-item* atau barang kebutuhan *player.* ( Nona Erlita Lambidju, Fanda Rundengan, 2018 )

* 1. **Forward Chaining**

Forward Chaining merupakan suatu penalaran yang dimulai dari fakta untuk mendapatkan kesimpulan (conclusion) dari fakta tersebut. Forward chaining bisa dikatakan sebagai strategi inference yang bermula dari sejumlah fakta yang diketahui. Pencarian dilakukan dengan menggunakan rules yang premisnya cocok dengan fakta yang diketahui tersebut untuk memperoleh fakta baru dan melanjutkan proses hingga goal dicapai atau hingga sudah tidak ada rules lagi yang premisnya cocok dengan fakta yang diketahui maupun fakta yang diperoleh.

Misal :

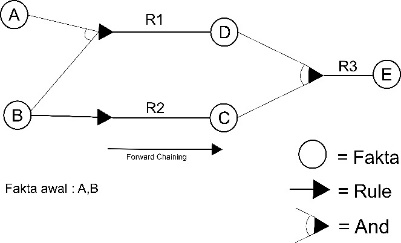
Terdapat 3 aturan yang tersimpan dalam basis pengetahuan yaitu :

R1 : if A and B then D

R2 : if B then C

R3 : if C and D then E

[Fakta](http://www.fakta.dk/) awal yang diberikan hanya A dan B, ingin membuktikan apakah E bernilai benar. Proses penalaran forward chaining terlihat pada gambar dibawah :



*Gambar 2.2 Contoh forward chaining*

Dari permasalahan di atas untuk mendapatkan nilai validasi yang berawal dari fakta-fakta maka dapat di dapat nilai validasi dengan perhitungan 2.1 berikut :

Persentasi validasi x 100% (2.1)

* 1. **Visual Studio**

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

**2.6 Unity 2D Game Engine**

*Game engine* adalah sebuah perangkat lunak yang di rancang untuk membuat sebuah *game*. Sebuah *game engine* biasanya di bangun dengan mengenkapsulasi beberapa fungsi standart yang umum di gunakan dalam pembuatan sebuah *game.* Misalnya, fungsi *rendering*, pemanggilan suara, *network*, atau pembuatan partikel untuk *special effect*. Sebagian besar *game engine* umumnya berupa *library* atau sekumpulan fungsi-fungsi yang penggunaanya terpadukan dengan Bahasa pemrograman.

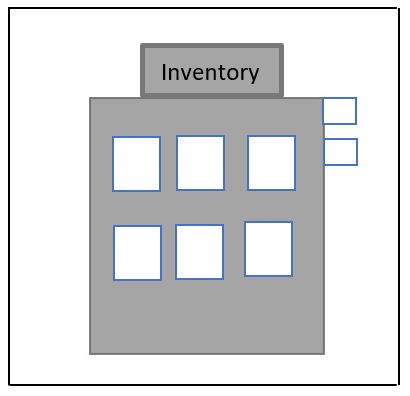
1. **Metode Penelitian**
   1. **Deskripsi Game**

*Game* ini mengangkat sebuah cerita legenda masyarakat jawa yang mengisahkan terbentuknya aksara jawa. Game ini ber*genre* RPG *(* *role-playing game )* yang dimainkan oleh *single player* dimana untuk memenangkan *game*, dibutuhkan ketangkasan *player* dalam mengalahkan *boss* *game* yang menjadikan *missi* utama dalam *game*. Selain melatih ketangkasan *player* dalam mengalahkan *boss* terakhir, *game* ini juga mengenalkan aksara jawa kepada *public*. Pada *game* “Ana Cerita” terdapat *item – item* pendukung untuk mempermudah dan memperkuat *player*, selain *item – item* pendukung yang membuat *player* kuat juga terdapat *item relief* atau aksara jawa yang di antara aksara tersebut yaitu aksara ha na ca.

Ajisaka di bekali *inventory* untuk mengumpulkan berbagi *item* yang di temui , terdapat dua macam *inventory* yaitu *inventory skill* dan *inventory* aksara atau *relief* dimana masing – masing *inventory* memiliki kapasitas jadi *player* harus bisa mempergunakannya dengan baik seperti kapan harus memakai *item* guna untuk memasukan *item* selanjutnya ketika ada kapasitas yang kosong.

* 1. **Rancangan Penelitian**

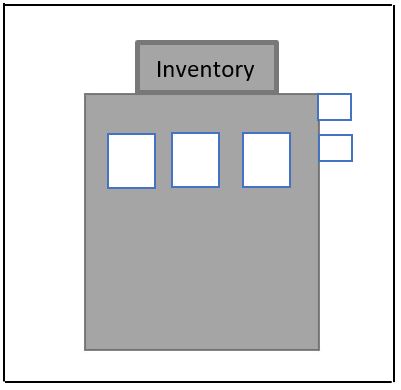
*Inventory* yang pertama di rancang untuk di teliti ialah *inventory skill* dimana *inventory* ini memiliki kapasitas enam *item*, berikut adalah rancangan mentah penilitian *inventory* *skill* dalam *game* “Ana Cerita” :



## *Gambar 3.1 Rancangan kapasitas inventory skill*

Pada gambar 3.1 terdapat *inventory skill* dengan kapasitas 6 *item* untuk penginputan *item – item* pendukung *player*, rancangan penelitian dalam gambar tersebut akan mendapatkan data yang akan di olah oleh metode *forward chaining*.

Sedangkan *inventory* yang ke dua yaitu *inventory* aksara atau *relief* dimana *inventory* tersebut di gunakan untuk menampung *item – item* aksara atau *relief* yang di antaranya terdapat relief ha, lalu kemudian relief na, dan yang terakhir yaitu relief ca, masing – masing *item* tersebut akan memenuhi kapasitas *inventory relief* yang akan mendapatkan data yang kemudian di olah oleh *forward chaining*, berikut adalah rancangan mentah penilitian *inventory relief* dalam *game* “Ana Cerita”:



## *Gambar 3.2 Rancangan kapasitas inventory relief*

Pada gambar 3.1 terdapat *inventory skill* dengan kapasitas 3 *item* untuk penginputan *item – item* pendukung *player*, rancangan penelitian dalam gambar tersebut akan mendapatkan data yang akan di olah oleh metode *forward chaining*.

* 1. **Populasi Dan Sampel**

Dalam pembahasan subbab ini peniliti akan mengidentifikasi kebutuhan – kebutuhan sampel yang akan di pergunakan dalam jalanya game “Ana Cerita”, yang akan di pergunakan oleh jalannya penelitian ini berdasarakan dengan metode *forward chaining*, berikut di ambil beberapa sampel yang akan membatasi kebutuhan.

Penelitian berdasar pada sampel berupa objek digital yang diteliti yakni :

1. Kebutuhan *item player* dalam *inventory.*
2. Kebutuhan senjata dan obat *player* dalam *inventory.*
3. *Inventory skill* guna pengumpulan *item-item* pendukung *player.*
4. *Inventory* aksara atau *relief* guna pengumpulan *item-item* pendukung *player.*
   1. **Definisi Oprasional**

Dari beberapa sampel yang diperoleh peneliti mendapatkan berbagai objek – objek atau keputuhan yang akan di pergunakan yang akan kemudian akan di peroses atau di olah datanya menggunakan *forward chaining*, pada table 3.1 dapat di lihat beberapa definisi kebutuhan – kebutuhan yang di ambil dari sampel :

**Tabel 3.1 Definisi oprasional**

|  |  |
| --- | --- |
| Variabel | Definisi Operasional |
| Senjata | Objek yang akan diteliti, diambil dari salah satu kebutuhan *player* dalam mengalahkan *enemy.* |
| Obat-obatan | Objek yang akan diteliti, diambil dari salah satu kebutuhan *player* untuk mengobati diri jika nyawa *player* berkurang. |
| Racun | Racun juga salah satu objek yang akan diteliti, di ambil dari salah satu kebutuhan *player* guna untuk memudahkan *player* dalam mengalahkan monster. |
| *Item* | Dalam perjalanan *player* akan menemukan berbagai rintangan dan akan menemmuakan berbagai *item* yang di gunakan untuk memudahkan *player* dalam menyelesaikan sebuah *game.* |
| *Inventory skill* | Menentukan *skill* yang berupa *item,* memudahkan *player* menyelesaikan permainan. |
| *Inventory relief* | Menentukan kelengkapan barang atau *item-item* kebutuhan *player* berupa aksara atau *relief.* |

* 1. **Instrumen Penelitian**

Dalam sebuah penelitian, sarana yang digunakan sebagai instrumen penelitian sangat berpengaruh terhadap kinerja peneliti melakukan kegiatannya dengan kata lain kebutuhan system pendukung pekerjaan sangatlah berpengaruh terhadap pekerjaan, berikut adalah sistem kebutuhan pekerjaan yang di gunakan untuk menyelesaikan penelitian :

### Kebutuhan Perangkat Keras *( hardware )*

Kebutuhan perangkat keras yang di gunakan untuk membangun aplikasi *game* ini adalah:

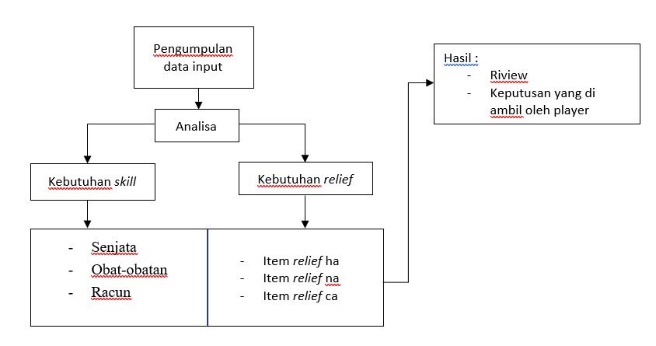
1. Processor : Intel corei5-9600K 4,60GHz,
2. RAM 6GB,
3. VGA 4GB,
4. Monitor,
5. Keyboard dan Mouse,

### 2 Kebutuhan Perangkat Lunak ( software )

Kebutuhan perangkat lunak yang di gunakan untuk membangun aplikasi *game* ini adalah:

1. Sistem Operasi : Windows 8.1 64 Bit,
2. *Game Engine* : Unity 5.5 +,
3. *Script Writer* : Visual Studio
   1. **Pengumpulan Data**

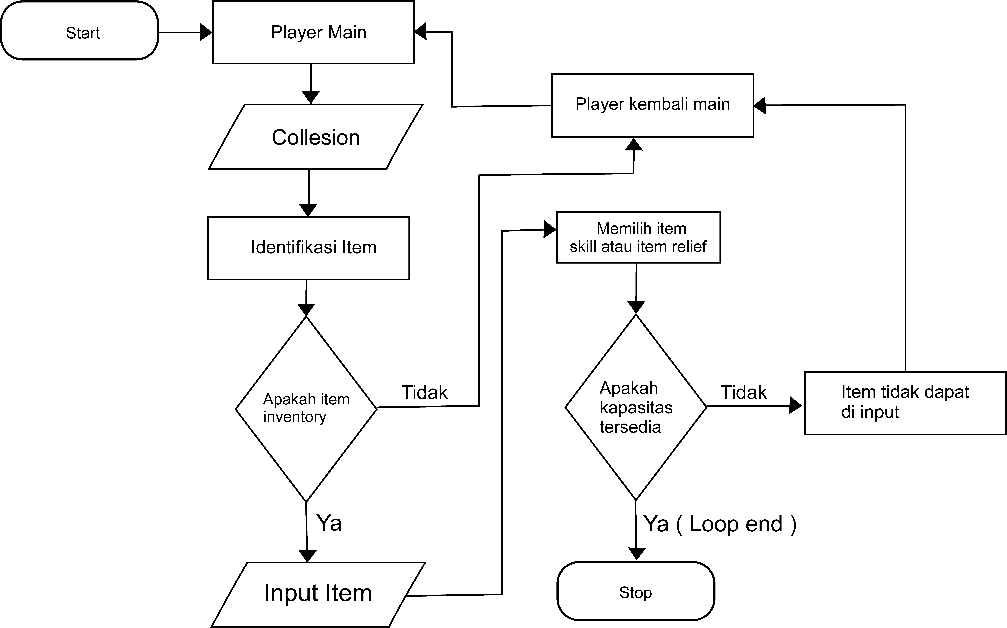
Demi kelancaran penelitian maka harus di gali berbagai data kebutuhan peneliti yang melibatkan dari berbagai proses pengambilan data yang akan di gunakan oleh peneliti, dari berbagai pengumpulan data maka peneliti mengumpulkan data dari membaca berbagai sejarah yang mengisahakan terciptanya aksara jawa dimana ajisaka yang pemeran utama dalam *game* yang akan di teliti ini adalah karakter baik yang akan mengalahkan dewatacengkar raja yang sangat jahat, Berikut adalah diagram pengumpulan data *input* :



## *Gambar 3.6 Diagram pengumpulan data*

* 1. **Analisis Data**

Dalam analisis data akan mendapatkan data input dalam *game* “Ana Cerita” yang kemudian akan di proses dalam metode *forward chaining,* brikut adalah *flowchard inventory* dalam *game* “Ana Cerita”:



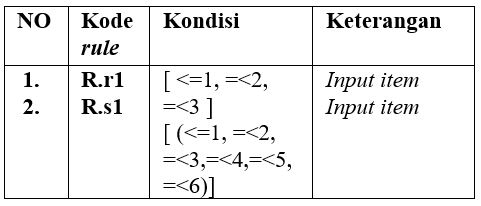
*Gambar 3.7 Flowchart inventory “Ana Cerita”*

Metode *Forward Chaining* dalam game “Ana Cerita” ini digunakan pada *system Inventory player*, dengan tahapan sebagai berikut :

1. Mulai permainana, jika *item* berupa item racun maka  *item* akan mengarah ke *inventory skill.*
2. Mulai permainana, jika *item* berupa item tombak maka  *item* akan mengarah ke *inventory skill.*
3. Mulai permainana, jika *item* berupa item obat maka  *item* akan mengarah ke *inventory skill.*
4. Mulai permainana, jika *item* berupa *item relief ha* maka *item* akan mengarah ke *inventory relief*.
5. Mulai permainana, jika *item* berupa *item relief na* maka *item* akan mengarah ke *inventory relief*.
6. Mulai permainana, jika *item* berupa *item relief ca* maka *item* akan mengarah ke *inventory relief*.
7. Permainan selesai.

Berikut adalah table variable jalannya penelitian ini di lakukan :

Tabel 3.2 Variabel *rule*



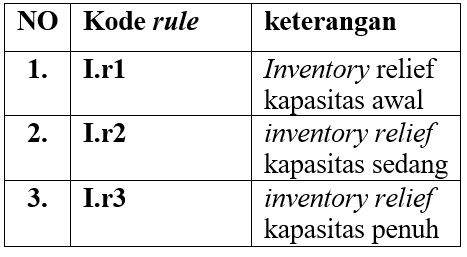
Keterangan kondisi *input item inventory relief* :

1. jika kapasitas = 3 dan *item relief* = 1 maka kondisi = (=<3)
2. jika kapasitas = 2 dan *item relief* = 2 maka kondisi = (=<2)
3. jika kapasitas = 1 dan *item relief* = 3 maka kondisi = (<=1)

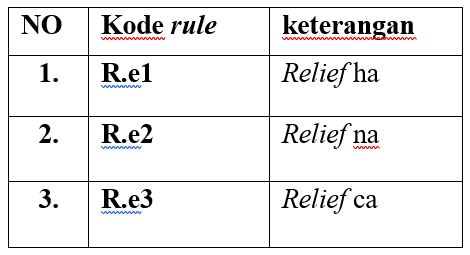
Keterangan kondisi *input item inventory skill* :

1. jika kapasitas = 6 dan *item skill* = 1 maka kondisi = (=<6)
2. jika kapasitas = 5 dan *item skill* = 2 maka kondisi = (=<5)
3. jika kapasitas = 4 dan *item skill* = 3 maka kondisi = (=<4)
4. jika kapasitas = 3 dan *item skill* = 4 maka kondisi = (=<3)
5. jika kapasitas = 2 dan *item skill* = 5 maka kondisi = (=<2)
6. jika kapasitas = 1 dan *item skill* = 6 maka kondisi = (=<1)

Tabel 3.3 Variabel *Inventory* dan *Item*

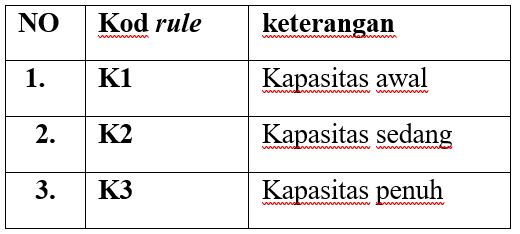








Tabel 3.4 Variabel Kapasitas



Tabel 3.5 *Representasi* Kondisi

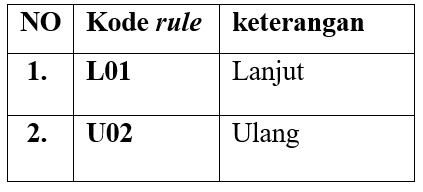


Table 3.6 *Representasi Review inventory skill*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Kondiai** | **Kode *rule*** | **Keterangan** |
| **1.** | *Input item* <= 1, == 6 kapasitas | **H01** | *Input Inventory* *skill* |
| **2.** | *Input item* =< 2, == 5 kapasitas | **H02** | *Input Inventory* *skill* |
| **3.** | *Input item* =< 3, == 4 kapasitas | **H03** | *Input Inventory* *skill* |
| **4.** | *Input item* =< 4, == 3 kapasitas | **H04** | *Input Inventory* *skill* |
| **5.** | *Input item* =< 5, == 2 kapasitas | **H05** | *Input Inventory* *skill* |
| **6.** | *Input item* =< 6, == 1 kapasitas | **H06** | *Input Inventory* *skill* |
| **7.** | Ulan*g / Game Over* | **H07** | Ulan*g / Game Over* |

Table 3.7 *Representasi Review inventory relief*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Kondiai** | **Kode *rule*** | **Keterangan** |
| **1.** | *Input item* <= 1, == 3 kapasitas | **I01** | *Input inventory relief* |
| **2.** | *Input item* =< 2, == 2 kapasitas | **I02** | *Input inventory relief* |
| **3.** | *Input item* =< 3, == 1 kapasitas | **I03** | *Input inventory relief* |
| **7.** | Ulan*g / Game Over* | **I04** | Ulan*g / Game Over* |

1. **Pengujian system**

Berikut ini adalah pengujian yang dilakukan terhadap sistem terutama pengujian terhadap kerja metode forward chaining. Dari hasil pengujian pada Tabel 5 jika dilakukan *review* terhadap *rules* terdapat 12 aturan untuk menentukan nilai goal pada fakta menuju ruangan keempat ( R.u4 ) yang sudah diterapkan pada metode *forward chaining dalam game* ini maka akan didapatkan penalaran sebagai berikut:

#### Pengujian system inventory ruangan petama

Rules input item :

1. R1 : IF O.b1 AND R.a1 THEN I.s1
2. R2 : IF R.e1 THEN I.r1
3. R3 : IF I.s1 AND I.r1 THEN R.u1

#### Pengujian system inventory ruangan kedua

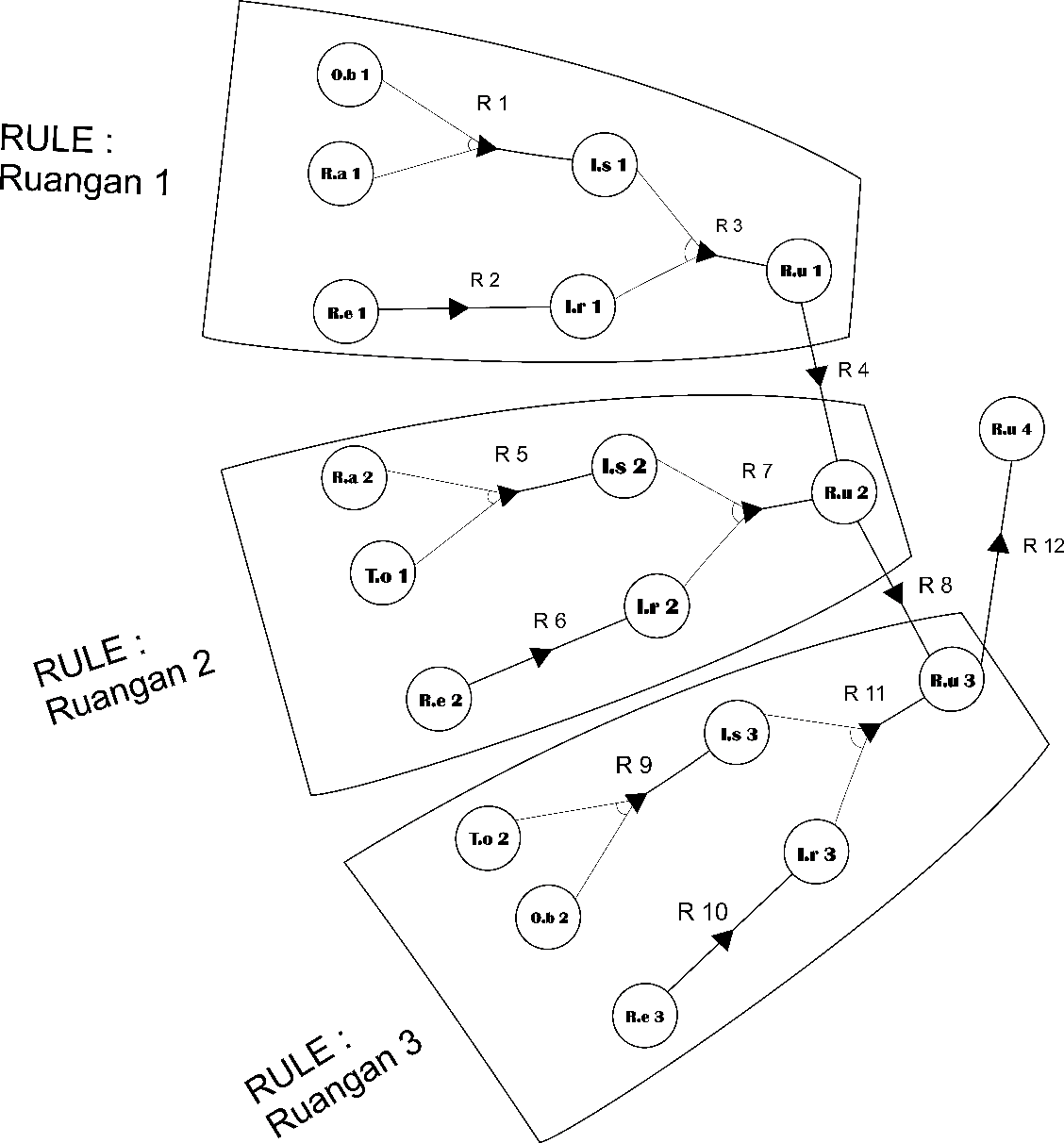
Rules input item :

1. R4 : IF R.u1 THEN R.u2
2. R5 : IF R.a2 AND T.o1 THEN I.s2
3. R6 : IF R.e2 THEN I.r2
4. R7 : IF I.s2 AND I.r2 THEN R.u2
5. Pengujian *system inventory* ruangan ketiga

Rules input item :

1. R8 : IF R.u2 THEN R.u3
2. R9 : IF T.o2 AND O.b2 THEN I.s3
3. R10 : IF R.e3 THEN I.r3
4. R11 : IF I.s3 AND I.r3 THEN R.u4
5. R12 : IF R.u3 THEN R.u4

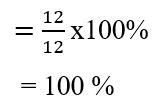
[Fakta](http://www.fakta.dk/) awal yang diberikan hanya O.b1 dan R.a1, ingin membuktikan apakah R.u4 bernilai benar. Proses penalaran forward chaining terlihat pada gambar 4.19 di bawah ini :



Gambar 4.19 Rule Inventory

jumlah\_valid = 12

sesuai dengan persentase pada gambar 2.1 didapat nilai *validasi* persentase :



Perhitungan nilai *validitas* diatas didasarkan pada perhitungan pengujian terhadap *rules* pada *forward chaining* pada *game.* Dari hasil perhitungan didapatkan nilai *validitas* sebesar 100%. Hal ini membuktikan bahwa Hasil dan *Review* yang didapat dalam *game* “Ana Cerita” ketika bermain sama dengan *rules* yang sudah dimasukkan dalam *metode forward chaining.*

Tabel 4.15 Pengujian sistem

|  |  |  |  |
| --- | --- | --- | --- |
|  | **INVENTORY** | **SKILL** |  |
| **No** | **Kondisi** | **Riview** | **Hasil** |
| **1** | R.s1 =’1’, K1 =’ok’, I.s1 =’ok’, H01 = ‘<=1. ==6  1 *item* masuk, *level* kapasitas awal, deteksi *item skill,* kondisi *rule* <=1, ==6 | H01  *Input Inventory* *skill* | L01  Lanjut |
| **2** | R.s1 =’2’, K1 =’ok’, I.s1 =’ok’, H02 = ‘<=2. ==5  2 *item* masuk, *level* kapasitas awal, deteksi *item skill*, kondisi *rule* =<2, ==5 | H02  *Input Inventory* *skill* | L01  Lanjut |
| **3** | R.s1 =’3’, K1 =’ok’, I.s1 =’ok’, H03 = ‘=<3, ==4  3 *item* masuk, *level* kapasitas awal, deteksi *item skill,* kondisi *rule* =<3, ==4 | H03  *Input Inventory* *skill* | L01  Lanjut |
| **4** | R.s1 =’4’, K2 =’ok’, I.s1 =’ok’, H04 = ‘=<4, ==3  4 *item* masuk, *level* kapasitas sedang, deteksi *item skill*, kondisi *rule* =<4, ==3 | H04  *Input Inventory* *skill* | L01  Lanjut |
| **5** | R.s1 =’5’, K2 =’ok’, I.s1 =’ok’, H05 = ‘=<5, ==2  5 *item* masuk, *level* kapasitas sedang, deteksi *item skill*, kondisi *rule* =<5, ==2 | H05  *Input Inventory* *skill* | L01  Lanjut |
| **6** | R.s1 =’6’, K3 =’ok’, I.s1 =’ok’, H06 = ‘=<6, ==1  6 *item* masuk, *level* kapasitas penuh, deteksi *item skill,* kondisi *rule* =<6, ==1 | H06  *Input Inventory* *skill* | L01  Lanjut |
| **7** | R.s1 =’7’, K3 =’ok’, I.s1 =’ok’, H07 = -  7 *item* masuk, level kapasitas penuh, deteksi *item skill,* kondisi *rule* ‘ –‘ | H07  Ulan*g / Game Over* | U01  Ulang |
|  | **INVENTORY** | **RELIEF** |  |
| **1** | R.r1 =’1’, K1 =’ok’, I.r1 =’ok’, I01 = ‘<=1. ==3  1 *item* masuk, *level* kapasitas awal, deteksi *item relief,* kondisi *rule* <=1, ==3 | I01  *Input inventory relief* | L01  Lanjut |
| **2** | R.r1 =’2’, K2 =’ok’, I.r1 =’ok’, I02 = ‘<=2. ==2  2 *item* masuk, *level* kapasitas sedang, deteksi *item relief*, kondisi *rule* =<2, ==4 | I02  *Input inventory relief* | L01  Lanjut |
| **3** | R.r1 =’3’, K3 =’ok’, I.r1 =’ok’, I03 = ‘=<3, ==1  3 *item* masuk, *level* kapasitas penuh, deteksi *item relief*, kondisi *rule* =<3, ==1 | I03  *Input inventory relief* | L01  Lanjut |
| **4** | R.s1 =’4’, K3 =’ok’, I.r1 =’ok’, I04 = -  4 *item* masuk, *level* kapasitas penuh, deteksi *item relief,* kondisi *rule* ‘ –‘ | I04  *Input inventory relief* | U01  Ulang |

1. **Pembahasan**

### Kapasitas *input item skill*

Dalam permainan “Ana Cerita” yang di terapkan dalam *unity game engine* memiliki tampilan *inventory skill* dengan enam kapasitas, berikut adalah berbagi tampilan penginputan *item skill* ajisaka yang di terapkan dalam *unity game engine* :



*Gambar 5.6 Input item tombak*

### b. Kapasitas *input item relief*

Dalam permainan “Ana Cerita” yang di mainkan dalam *unity game engine* memiliki tampilan *inventory relief* dengan tiga kapasitas, berikut adalah berbagi tampilan penginputan *item relief* yang di mainkan dalam *unity game engine* :



Gambar 5.9 *Input item relief* ca

## **6 Kesimpulan**

Berdasarkan pada hasil pembahasan hasil penerepam metode forward chaining pada inventory di game Ana Cerita menggunakan unity game engine, maka dapat di peroleh hasil atau kesimpulan sebagai berikut :

1. Penggunakan *rule* yang di terapkan terhadap *forward chaining* dalam *game* Ana Cerita terhadap *inventory skill* dan *inventory relief* dapat menghitung nilai *input* terhadap nilai kapasitas, dengan nilai *input* penuh memiliki enam kapasitas *inventory skill* dan tiga nilai input penuh *inventory relief*.

2. Konsep algoritma *forward chaining* pada *game* Ana Cerita di gunakan untuk proses *review* dan *rules* mendapatkan sebelas percobaan dengan tujuh percobaan terhadap nilai validasi *inventory skill* dan empat percobaan nilai validasi pada *inventory relief.*

3. *Rules* yang di gunakan pada *forward chaining* ini sebanyak duabelas *rules* yang menunjukan nilai validasi 100 %, menunjukan bahwa *riview* dalam *game* dengan *rules* yang di masukan dalam *forward chaining* sama.

4. Hasil penerapan metode *forward chaining* terhadap *game* “ Ana Cerita “ pada *unity game engine* ketika di jalankan sebanyak sepuluh percobaan berjalan dengan baik.

# **DAFTAR PUSTAKA**

Academy, Hemera. 2015. Fungsi Microsoft Visual Studio,(online), https://itlearningcenter [.id/fungsi-microsoft-visual-studio/](https://itlearningcenter.id/fungsi-microsoft-visual-studio/), diakses 20 Maret 2019.

Adi, Purnama, Rachmat. Tri, Laksono, Putra, Adi. 2018. Aplikasi Web Server Berbasis Bahasa C Sharp. *Jurnal Pengertian Bahasa Pemrograman* C#, 4 (1): 21 – 29

Aryani, Wuri, Tri. Budiono, Alexius, Endy. 2014. Rancang Bangun Game Survivor Pada Mata Pelajaran IPA Kelas IV Pada SD Negri Panggungsari 01 Pangungrejo Blitar, *Jurnal Pengembangan Game,* 1 (5): 1-20

Budiharto, Widodo. Suhartono, Derwin. 2104. Artificial Intelligence Konsep Dan Penerapannya. Yogyakarta.

Hidayat, Rahmat. 2018. Game-Based Learning: Academic Games sebagai Metode Penunjang Pembelajaran Kewirausahaan. *Jurnal Pengembangan Game*, 26 (2) : 71- 85

Jauhari, Moh Arif. Hamidin, Dini. Rahmatuloh, Marwanto. 2017. Komparasi Stabilitas Eksekusi Kode Bahasa Pemrogrman .Net C# Versi 4.0.3019 Dengan Google Golang Versi 1.4.2 Menggunakan Algoritma Bubble Sort dan Insertion Sort. *Jurnal Pengenal Pemrograman C#,* 9(1) : 13-20.

Lambidju, Nona Erlita. Rundengan, Fanda. 2018. Ipteks Pengendalian Sistem Pencatat Persediaan PadaPT Berlian Kharisma Pasifik. *Jurnal Pengendalian Persediaan ( Inventory )*, 2 (2): 108-112.

Marissa, Reny. Erlin. 2015. Implementasi Metode Forward Chaining untuk Menentukan Kenaikan Level pada Game Finding Selais. Jurnal Pengembangan Metode Forwrad Chaining, 1 (1): 1-9

Nugroho, A., Chandra, D. W., Prestiliano, J., Beeh, Y.R. 2014. Cara Mudah Dan Cepat Belajar Pemrograman C#.Net. Yogyakarta: ANDI.

Pratama, Wahyu. 2014. Game Adventure Misteri Kotak Pandor. *Jurnal Pengembangan Game*, 9 (2): 13 - 31.

Reodavan, Rickman. 2014. Unity Tutorial Game Engine. Bandung : Informatika Bandung.

Zamroni, M. Rosidi. Suryaman, Nizar. Jalaluddin, Ahmad. 2014. Rancang Bangun Aplikasi Permainan Untuk Pembelajaran Anak Menggunakan HTML 5, Jurnal Pengertian Game, 5 (2): 489 - 494