

Game Chicken Roll dengan Menggunakan Metode *Forward Chaining*

Yogie Susdyastama Putra, M. Aziz Muslim dan Agus Naba

Abstrak—Metode *forward chaining* pada umumnya digunakan untuk sistem pendukung keputusan dan sistem pakar. Penelitian ini menggunakan algoritma *forward chaining*, khususnya untuk proses review dan untuk menentukan apakah seorang pemain game layak melanjutkan ke level berikutnya. Algoritma *forward chaining* adalah algoritma yang berbasis pada fakta-fakta atau premise yang ada sehingga menghasilkan sebuah kesimpulan atau konsekuensi. Hasil pengujian menunjukkan bahwa nilai validitas mencapai 100%. Hasil didapat dari komparasi data antara rules dan hasil pengujian yang didapat saat bermain game.

Kata Kunci—*Game, Chicken Roll, Forward Chaining, Sistem Pakar, Sistem Pendukung Keputusan.*

I. PENDAHULUAN

PENGEMBANGAN *game* era ini mempunyai prospek yang besar. Banyak orang suka bermain *game* demi menghilangkan penat ataupun saat mempunyai waktu luang. Banyak peluang yang ditawarkan industri *game*. Salah satunya adalah perancangan *game*. *Game* merupakan salah satu industri besar di dunia saat ini. Perkembangan *game* begitu pesat dengan jenis yang beragam, mulai *game* yang hanya dapat dimainkan oleh satu orang saja hingga *game* yang dapat dimainkan oleh beberapa orang sekaligus.

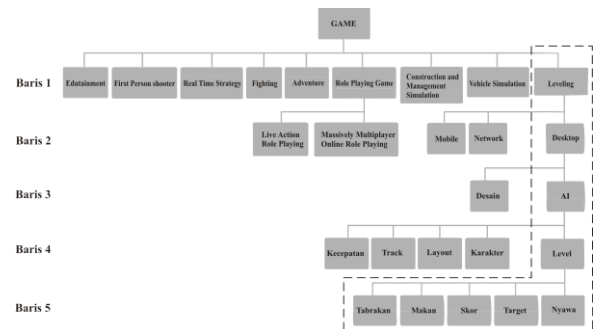
Game chicken roll yang dirancang pada penelitian ini merupakan *game* yang bertipe *levelling*, dimana kekuatan *game* terletak pada tingkat kesulitan yang berbeda pada tiap *level*. *Game* ini berjalan pada platform *desktop* (*Windows*). Pada sekali permainan *game* ini hanya dapat dimainkan oleh seorang *user*.

Game ini merupakan implementasi konsep AI khususnya metode *forward chaining*. Pada dasarnya metode *forward chaining* dimanfaatkan untuk menentukan apakah *user* dapat melanjutkan permainan ke *level* berikutnya atau tidak.

Metode *forward chaining* adalah sebuah algoritma yang dititikberatkan pada pendekatan yang dimotori data. Dalam pendekatan ini pelacakan dimulai dari

informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. 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]. Menurut Meenakshi Batra dan Vijay Batra dalam jurnalnya pada Maret 2006 menyatakan bahwa konsep *Forward Chaining* cukup efektif untuk melatih anak yang mempunyai keterbelakangan mental [2].

Gambar 1 adalah kerangka konsep penelitian *game chicken roll*:



Gambar 1. Kerangka Konsep Penelitian

Keterangan:

- Garis Putus-Putus : Penelitian yang dikerjakan.
- Baris 1 : *Game* mempunyai 9 jenis (*Edutainment, FPS, RTS, Fighting, Adventure, RPG, Construction and Management Simulation, Vehicle Simulation, Levelling*).
- Baris 2 : Penelitian ini berfokus pada pengembangan *game* untuk *desktop*.
- Baris 3 : Penelitian ini lebih mengutamakan pengembangan pada bidang AI (*Artificial Intelligence*) khususnya dengan menggunakan *forward chaining*.
- Baris 4 : Konsep AI dimanfaatkan untuk *levelling process*. Memanfaatkan *forward chaining* untuk menentukan seorang *user* dapat melanjutkan ke *level* berikutnya atau tidak.
- Baris 5 : Variabel yang dijadikan acuan sebagai bahan *input* untuk konsep AI dengan *forward chaining*.

Untuk menghasilkan *game* yang menarik diperlukan tampilan desain yang indah dan menarik. Namun *game* dalam penelitian ini hanya berfokus pada implementasi

Yogie Susdyastama Putra adalah mahasiswa Program Magister Fakultas Teknik Universitas Brawijaya Malang (085755078816, yogie.putra8@gmail.com)

M. Aziz Muslim adalah Dosen jurusan Teknik Elektro Universitas Brawijaya Malang (muh_aziz@ub.ac.id)

Agus Naba adalah Dosen pada jurusan Fisika Universitas Brawijaya Malang (anaba@ub.ac.id)

konsep AI untuk pengembangan *game* saja, karena saat ini penelitian dalam bidang *Game Development* masih sangat jarang. Secara spesifik dengan metode penerapan AI (*Artificial Intelligence*) *Forward Chaining*, yang dikembangkan menggunakan *tools* Adobe Flash dan bahasa pemrograman AS3 (*ActionScript 3*). *ActionScript 3* adalah bahasa pemrograman berbasis OOP (*Object Oriented Programming*) yang cukup handal dalam membuat aplikasi dan *game* [3].

II. LANDASAN TEORI

A. *Game*

Dalam kehidupan sehari-hari, manusia cenderung untuk menemukan beberapa hiburan untuk mengambil istirahat dari rutinitas sehari-hari mereka. Bayi mencari hiburan pada mainan, sedangkan dewasa lainnya, lebih memilih sesuatu yang lebih rumit dan menghibur seperti permainan atau *game*. Ada perbedaan antara permainan dan mainan. Sebuah permainan harus dikemas menyenangkan di dalamnya. Dan parameter yang bervariasi. Tantangan seperti mendulang poin, atau nilai replay bisa menjadi faktor menyenangkan dari permainan. Dan pada daftar yang sangat, menyenangkan dalam permainan juga harus seimbang dengan faktor keadilan, permainan harus menerapkan aturan yang adil untuk setiap pemainnya.

Berikut adalah beberapa kriteria permainan yang menyenangkan untuk dimainkan antara lain:

- Pemain harus bersenang-senang, bukan *programmer* atau *designer*.
- Permainan dimulai dengan pembukaan yang bagus.
- *Gameplay* yang bagus adalah aliran keputusan yang menarik yang harus diselesaikan oleh pemain.
- Tempatkan pemain dalam posisi seorang pahlawan. Proses pengembangannya:
 - *Prototype* dan bermain *game* untuk sementara waktu.
 - Selanjutnya minta orang lain untuk mencoba permainan.
 - Secara cermat mengamati pemain.
 - Secara terus-menerus beri penghargaan pemain dalam permainan.
 - Ujicoba ke penguji baru.
 - Membuat permainan lebih sukar sampai pemain meminta ampun [4].

Di dalam sebuah *game*, terdapat beberapa komponen dasar yang cukup penting:

- **Grafik**
Grafik merupakan suatu hal yang penting dalam *game*, dimana grafik ini akan membuat *game* menjadi daya tarik *user* untuk memainkan *game*. Pada awalnya *game* menggunakan grafik monokrom dengan layar hitam putih sampai sekarang sudah menggunakan teknologi 3D, semakin bagus grafik maka *game* tersebut semakin seperti dalam dunia nyata. Grafis yang bisa

digunakan sangat beragam, diantaranya adalah HOpenGL [5].

- **Suara**
Suara menjadi komponen *game* karena dengan adanya suara pada *game* maka *game* tersebut akan menarik.
- **AI (kecerdasan Buatan)**
Kecerdasan buatan atau AI (*artificial Intelligence*) menjadi komponen *game* karena dengan hal tersebut *game* menjadi lebih menantang.
- **Skenario *Game***
Skenario *game* menjadi komponen *game* karena si pemain seolah-olah bermain sesuai dengan petunjuk yang ada di *game* tersebut.
- ***Multiplayer***
Multiplayer adalah sebuah *mode* dalam *game* dimana *game* dimainkan oleh pemain lainnya sehingga bermain *game* secara bersama-sama. Era ini *multiplayer game* cukup populer [6].

Era ini konsep *game* banyak diterapkan untuk mensimulasikan beberapa aspek kejadian pada dunia nyata. Diantaranya adalah untuk simulasi pengambilan keputusan ketika terjadi bencana alam yang mengganggu *supply* sebuah produk dari produsen ke customer [7], simulasi untuk menganalisa kinerja *Mobile Ad Hoc Networks* [8], simulasi untuk manajemen *supply chain* [9], untuk simulasi keamanan jaringan [10], dan simulasi untuk menganalisa bagaimana manusia berperilaku [11]

Game chicken Roll merupakan *game* bertipe *levelling*. *Game* ini berfokus pada pengembangan *level*. *Game* ini memanfaatkan konsep *forward chaining* untuk pengembangan setiap levelnya.

B. *Forward Chaining*

Forward chaining adalah suatu rantai yang dicari atau dilewati/ dilintasi dari suatu permasalahan untuk memperoleh solusinya disebut dengan *forward chaining*. Cara lain menggambarkan *forward chaining* ini adalah dengan penalaran dari fakta menuju konklusi yang terdapat dari fakta [1].

Contoh sederhana dari *forward* dan *backward chaining* seperti berikut ini: misalkan anda sedang mengemudi dan tiba-tiba anda melihat mobil polisi dengan cahaya kelap-kelip dan bunyi sirine. Dengan *forward chaining* mungkin anda akan berkesimpulan bahwa polisi ingin anda atau seseorang untuk berhenti. Itu adalah fakta awal yang mendukung dua kemungkinan konklusi. Jika mobil polisi membuntuti dibelakang anda atau polisi melambaikan tangan memberhentikan anda, maka kesimpulan lebih lanjut adalah polisi ingin anda yang berhenti.

Dalam implementasinya, *forward chaining* sangat membantu *developer* aplikasi dalam membangun sebuah sistem. Karena dengan penggunaan metode ini jika *developer* ingin menambah beberapa kondisi dan aturan, *developer* tidak perlu membongkar lagi kode program dari awal.

Berikut adalah karakteristik *forward chaining*:

TABEL I.
KARAKTER FORWARD CHAINING.

<i>Forward Chaining</i>	
Perencanaan, <i>Monitoring</i> , <i>Control</i> .	
Disajikan untuk masa depan.	
Data memandu, penalaran dari bawah ke atas.	
Bekerja kedepan untuk mendapatkan solusi apa yang mengikuti fakta.	
<i>Breadth first search</i> dimudahkan.	

Pada Metode *forward chaining* juga di artikan sebagai pendekatan yang dimotori data. Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Sehingga metode ini juga sering disebut "Data driven".

Operasi dari sistem *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui ke dalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai *goal* atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui. Operasi tersebut dapat digambarkan seperti gambar berikut:

Langkah-langkah yang harus dilakukan dalam membuat sistem *forward chaining* berbasis aturan, yaitu:

- Pendefinisian Masalah.
Tahap ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.
- Pendefinisian Data Input.
Sistem *forward chaining* memerlukan data awal untuk memulai inferensi.
- Pendefinisian Struktur Pengendalian Data.
Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan[9].

Gambar 2 adalah konsep kerja *Forward Chaining* pada game *Chicken Roll*:



Gambar 2. Konsep Kerja Forward Chaining

Berikut adalah variabel yang digunakan untuk sebagai parameter dalam forward chaining untuk pengembangan game *Chicken Roll*:

TABEL II.
VARIABEL RULE.

No.	Kode Rule	Kondisi	Keterangan
1.	R01	[0, 1, 2, 3]	Nyawa
2.	RXX

TABEL III.
VARIABEL HASIL.

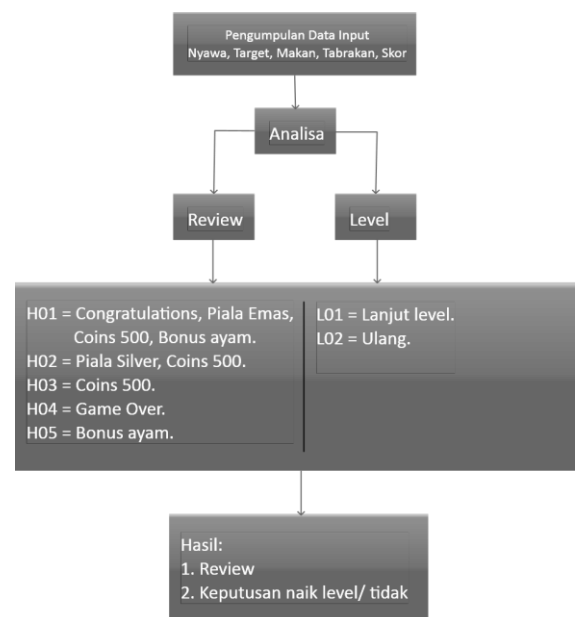
No.	Kode Rule	Review
1.	H01	Congratulations, Piala Emas, Coins 500, Bonus ayam.
2.	HXX	...

TABEL IV.
VARIABEL LEVEL.

No.	Kode Level	Keterangan
1.	L01	Lanjut Level
2.	LXX	...

III. METODE PENELITIAN

Proses representasi pengetahuan dilakukan dengan cara mengumpulkan kondisi untuk memperoleh hasil. Berdasarkan data *input* yang diperoleh, dilakukan analisis untuk menentukan review apa yang didapat dalam setiap levelnya serta layak atau tidaknya seorang *user* untuk naik *level* dalam *game*.



Gambar 3. Proses Representasi Cara Kerja Game

TABEL V.
REPRESENTASI RULE.

No.	Kondisi	Kode	Keterangan
1.	[0, 1, 2, 3]	R01	Nyawa
2.	[G, TG]	R02	Target
3.	[=>500, <500]	R03	Skor
4.	[M, TM]	R04	Makan
5.	[T, TT]	R05	Tabrakan

TABEL VI.
REPRESENTASI REVIEW.

No.	Kondisi	Kode
1.	Congratulations, Piala Emas, Coins 500, Bonus ayam.	H01
2.	Piala Silver, Coins 500.	H02
3.	Coins 500.	H03
4.	Game Over.	H04
5.	Bonus Ayam.	H05

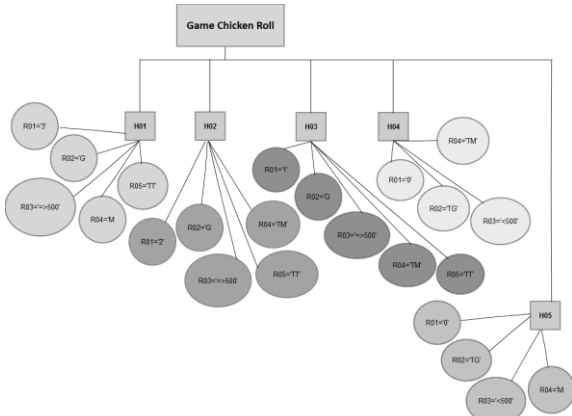
TABEL VII.
REPRESENTASI LEVEL.

No.	Kode Level	Keterangan
1.	L01	Lanjut Level
2.	L02	Ulang

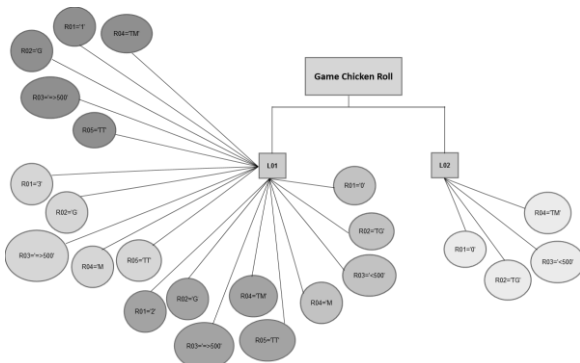
Hasil representasi pengetahuan *rules* kondisi disajikan pada Tabel V, representasi pengetahuan *review*

disajikan pada Tabel VI, sedangkan representasi pengetahuan kenaikan level disajikan pada Tabel VII.

Rules Kondisi dan Hasil direpresentasikan menjadi sebuah *Inference Tree* yang sangat membantu dalam pembentukan aturan untuk selanjutnya diproses oleh *inference engine*. Gambar 4 adalah merupakan representasi Rules Kondisi dan Review:



Gambar 4. Inference Tree Review dan Kondisi



Gambar 5. Inference Tree Hasil dan Kondisi

TABEL VIII
ATURAN GAME CHICKEN ROLL BERDASARKAN LEVELING

No.	Kode Level	Keterangan
1.	H01, H02, H03, H05.	Lanjut Level L01
2.	H04	L02
		Ulang

Sedangkan Gambar 5 adalah representasi Rules Kondisi dan Hasil.

Setelah melewati tahap *inference tree* tersebut selanjutnya akan diubah menjadi algoritma pemrograman.

Berikut adalah representasi dari *inference tree* kedalam tabel aturan yang akan digunakan untuk proses analisa pada *game chicken roll*:

Sedangkan pada Tabel IX akan dijelaskan aturan untuk kenaikan level pada *game*.

IV. HASIL DAN PEMBAHASAN

Proses pengujian dilakukan pada desain setiap tampilan *scene* pada *game*, kerja dari metode *forward*

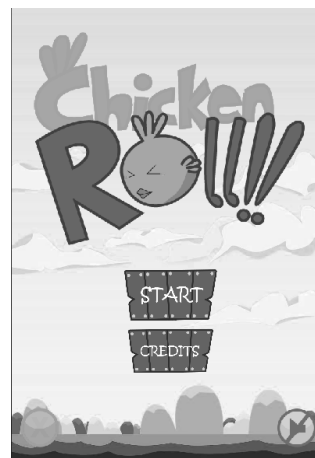
chaining, serta analisis mendalam terhadap algoritma.

A. Pengujian Tampilan Scene Game

Gambar 6 menunjukkan tampilan awal dari *game chicken roll*. Pada tampilan awal ini terdapat 4 tombol *Start* (memulai bermain), *Credits* (pembuat *game*), *Close* (Keluar dari *game*) dan *Sound* (mematikan *sound*).

TABEL IX
ATURAN GAME CHICKEN ROLL BERDASARKAN REVIEW

No.	IF	THEN
1.	R01='3' AND R02='G' AND R03='=>500' AND R04='M' AND R05='TT'	H01 Nyawa sisa 3, Target goal, skor => 500, makan, tidak tertabrak. Congratulations , Piala Emas, Coins 500, Bonus ayam.
2.	R01='2' AND R02='G' AND R03='=>500' AND R04='TM' AND R05='TT'	H02 Nyawa sisa 2, Target goal, skor => 500, tidak makan, tidak tertabrak.
3.	R01='1' AND R02='G' AND R03='=>500' AND R04='TM' AND R05='TT'	H03 Nyawa sisa 1, Target goal, skor => 500, tidak makan, tidak tertabrak. Coins 500.
4.	R01='0' AND R02='TG' AND R03='<500' AND R04='TM'	H04 Nyawa sisa 0, Target tidak goal, skor < 500, tidak makan.
5.	R01='0' AND R02='TG' AND R03='<500' AND R04='M'	H05 Nyawa sisa 0, Target tidak goal, skor < 500, makan. Bonus Ayam.

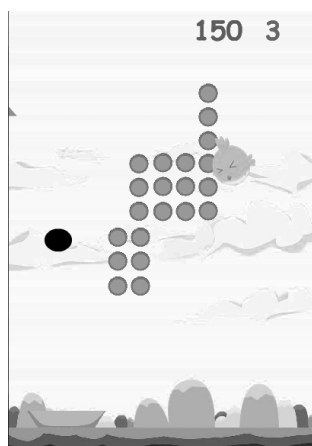


Gambar 6. User Interface Awal Game

Gambar 7 menunjukkan level 1 pada *game chicken roll*. Pada *scene* ini, *user* diminta untuk menyelesaikan permainan pada level 1 ini. Ketika *user* selesai memainkan level 1 ini maka algoritma *forward chaining*

akan bekerja sesuai dengan *data input* yang masuk. Penelitian ini hanya membahas sampai *level 4* saja.

Gambar 8 berikut ini menunjukkan tampilan *review* H01 pada *game chicken roll*. *Review* ini akan muncul ketika *user* selesai memainkan suatu *level* dalam *game* sesuai kondisi yang sudah ditentukan. Kondisi yang berbeda akan mendapatkan *review* yang berbeda pula sesuai dengan *rule* yang sudah ditetapkan.



Gambar 7. Tampilan Level 1



Gambar 8. Tampilan Review H01

B. Pengujian Sistem

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* yang sudah diterapkan pada metode *forward chaining* dalam *game* ini maka akan didapatkan perhitungan sebagai berikut:

Rules:

1. R01='3' AND R02='G' AND R03='>=500' AND R04='M' AND R05='TT' THEN H01
2. R01='2' AND R02='G' AND R03='>=500' AND R04='TM' AND R05='TT' THEN H02
3. R01='1' AND R02='G' AND R03='>=500' AND R04='TM' AND R05='TT' THEN H03
4. R01='0' AND R02='TG' AND R03='<500' AND R04='TM' THEN H01
5. R01='0' AND R02='TG' AND R03='<500' AND R04='M' THEN H05

jumlah_valid = 5

$$\begin{aligned} \text{Presentase Validitas} &= \frac{\text{jumlah_valid}}{\text{Jumlah_Pengujian}} \times 100\% \\ &= \frac{5}{5} \times 100\% \\ &= 100\% \end{aligned}$$

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* ketika bermain sama dengan *rules* yang sudah dimasukkan dalam metode *forward chaining*.

TABEL X.
PENGUJIAN SISTEM.

No.	Kondisi	Review	Hasil
1.	R01='3', R05='TT', R03='>=500', R04='M', R02='G'.	H01	L01
	Nyawa 3, Tidak Tertabrak, Koin >= 500, Makan Cacing, Goal.	Piala emas, koin emas, bonus ayam	Lanjut level
2.	R01='0', R02='TG'.	H04	L02
	Nyawa 0, Jatuh ke tanah.	Game Over	Ulang
3.	R01='2', R02	H02	L01
	Nyawa 2, Tidak Tertabrak, koin >= 500, Tidak Makan, Goal.	Piala silver, koin silver	Lanjut level
4.	R01='0', R02='TG', R03='<500', R04='M'.	H05	L01
	Nyawa 0, Koin < 500, Makan Cacing, Tidak Goal.	Bonus ayam	Lanjut level
5.	R01='1', R02='G', R03='>=500', R04='TM', R05='TT'.	H03	L01
	Nyawa sisa 1, Tidak Tertabrak, Koin >=500, Tidak Makan Cacing, Goal.	Koin emas	Lanjut level

V. KESIMPULAN DAN SARAN

Dari proses pengujian dan analisis yang telah dilakukan, kesimpulan yang dapat diambil pada penelitian ini antara lain konsep algoritma *Forward Chaining* pada *Game Chicken Roll* digunakan untuk proses *Review* dan perpindahan *level*, *rules* yang digunakan pada metode *forward chaining* ini sebanyak 5 *rules*, dan hasil pengujian menunjukkan nilai validitas

sebesar 100%. Sedangkan saran yang dapat diambil antara lain adalah *level* pada *Game* perlu ditambah lagi agar game semakin menarik, perlu ditambahkan beberapa variabel yang dapat membuat *game* semakin kompetitif, dan perlu dikembangkan untuk platform *mobile device*.

REFERENSI

- [1] Riskadewi & Hendrik, A. PENERAPAN SISTEM PAKAR FORWARD CHAINING BERBASIS ATURAN PADA PENGAWASAN STATUS PENERBANGAN, *10*(3). 2005.
- [2] Batra, M. & Batra, V. (2006). BETWEEN FORWARD CHAINING AND BACKWARD COMPARISON, *XXXVII*(3).
- [3] Radion, Kristo. 2012. Easy Game Programming Using Flash and ActionScript 3.0. Yogyakarta: Andi.
- [4] Rouse, Richard. Game Design: Theory & Practice. ISBN: 978-1-55622-912-1. United States: Wordware Publishing. 2001.
- [5] Cheong, M. H. The University of New South Wales School of Computer Science and Engineering Functional Programming and 3D Games Supervisor: Dr Manuel Chakravarty, (November). 2005.
- [6] Xin, C. Multiplayer Game in Mobile Phone Serious Game. *2009 International Joint Conference on Artificial Intelligence*, 56–58. doi:10.1109/IJCAI.2009.205. 2009.
- [7] Hu, M. Subjective Game Model of Supply Chain Emergency, 5283–5286. doi:10.1109/ICEE.2010.1323. 2010.
- [8] Li, Z., Member, S., & Shen, H. Game-Theoretic Analysis of Cooperation Incentive Strategies in Mobile Ad Hoc Networks, *11*(8), 1287–1303. 2012.
- [9] Merkuryev, Y., & Bikovska, J. (2012). Business Simulation Game Development for Education and Training in Supply Chain Management. *2012 Sixth Asia Modelling Symposium*, 179–184. doi:10.1109/AMS.2012.12. 2012.
- [10] Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., & Wu, Q. A Survey of Game Theory as Applied to Network Security. *2010 43rd Hawaii International Conference on System Sciences*, 1–10. doi:10.1109/HICSS.2010.35. 2010.
- [11] Camerer, C. F. Behavioral Game Theory: Thinking, Learning, and Teaching. 2001.