

# **GAME TUKKARMATIKA BERBASIS ANDROID MENGGUNAKAN METODE *FINITE STATE MACHINE* DAN ALGORITMA A\***

*Muhammad Murti Hanafi*  
*Jurusan Teknik Informatika,*  
*Fakultas Teknik*  
*Universitas Muhammadiyah Jember*  
[\*murthihanafi@gmail.com\*](mailto:murthihanafi@gmail.com)

## ***Abstract***

Emerging technology to make more advanced civilization. Technology used to build a life that is more useful and easier job.

Math is one subject which has been difficult in the community. Many originators of the technology to make people perceive others on these subjects.

Game is one of the technologies used by the public to make their work easier, in addition to assist students in mastering the material and implementation into real-world subjects. Game is very boring when the games are less active and tend to be static. To design a dynamic game that required a method for designing games. Finite State Machine (FSM is a method for designing a game to be dynamic. Finite State Machine (FSM) is used to create games have state, limits and level of play. FSM method has not been interesting if the game is won by user , the A \* algorithm is needed to create a dynamic enemy moving towards the player. That seem interesting and then the dynamic implementation of FSM methods and algorithms A \* in combination in a game that Game Tukkarmatika.

## **Abstrak**

Teknologi yang semakin berkembang membuat peradaban semakin maju. Teknologi digunakan untuk membangun kehidupan yang lebih bermanfaat dan mempermudah suatu pekerjaan.

Matematika merupakan salah satu mata pelajaran yang tergolong sulit dikalangan masyarakat. Banyak pencetus teknologi untuk membuat masyarakat mempunyai persepsi lain pada mata pelajaran ini.

*Game* merupakan salah satu teknologi yang digunakan oleh masyarakat untuk membuat pekerjaan mereka menjadi mudah, di samping itu untuk membantu para pelajar dalam menguasai materi maupun implementasi mata pelajaran kedalam dunia nyata. *Game* sangat membosankan apabila *game* kurang aktif dan cenderung statis. Untuk merancang *game* yang dinamis maka dibutuhkan metode untuk merancang *game*. *Finite State Machine* (FSM adalah sebuah metode untuk merancang *game* agar menjadi dinamis. *Finite State Machine* (FSM) digunakan untuk membuat *game* memiliki keadaan, batasan dan *level* permainan. Metode FSM belum menarik apabila permainan mudah dimenangkan oleh pengguna maka dibutuhkan algoritma A\* untuk membuat musuh yang dinamis bergerak ke arah *player*. Agar terkesan menarik dan dinamis maka implementasi dari metode FSM dan algoritma A\* di gabungkan dalam suatu *game* yaitu *Game Tukkarmatika*.

*Kata Kunci : Game Tukkarmatika, Game Matematika, Metode FSM (Finite State Machine), Algoritma A\* (A star), Construct 2, NPC, Game Android*

## PENDAHULUAN

Suatu *game* sangat membosankan apabila *game* kurang aktif dan cenderung statis, *game* mudah dimenangkan oleh pengguna dan *game* yang berisi tantangan yang kurang menarik seperti adanya musuh yang kurang cerdas, batasan waktu, *level* permainan dan sebagainya. Untuk merancang *game* dengan adanya tantangan musuh yang cerdas, maka dibutuhkan suatu metode untuk merangsang tingkah laku pengguna dalam menjalankan *game* tersebut dengan menggunakan *Finite State Machine* (FSM). FSM digunakan untuk mengarahkan pengguna agar berada pada keadaan yang dinamis. Keadaan yang dinamis juga belum bersifat signifikan untuk membuat pengguna *game* tertarik untuk memainkan *game* tersebut lagi. Maka, dibutuhkan algoritma yang memungkinkan adanya musuh yang cerdas sehingga pengguna dapat bermain sambil belajar dengan menyenangkan dan permainan tetap dilanjutkan. Algoritma A\* merupakan algoritma untuk memungkinkan musuh menjadi lebih aktif dan cerdas. Dengan adanya algoritma A\*, maka musuh akan mengejar pemain dengan jalur terpendek sehingga pemain dapat dikalahkan. *Game* berbasis android yang menggunakan metode *Finite State Machine* sebagai gambaran *game* yang dinamis dan algoritma A\* sebagai kecerdasan buatan yang dimiliki musuh ini akan menjadi menarik.

## METODE PENELITIAN

### A. Analisis *Game* yang Akan Dikembangkan

#### 1. Storyline

*Game* Tukarmatika merupakan *game* yang menceritakan tentang perjalanan Andi. Andi adalah anak yang tekun beribadah dan suka menolong, pada suatu hari dia sedang pergi ke hutan untuk mencari tanaman obat. Saat Andi masuk ke area hutan, tiba-tiba dia melihat seekor burung yang terjebak ke dalam ranting pohon yang lebat. Andi ingin menolong burung tersebut, tetapi banyak rintangan yang harus di hadapi oleh Andi

Pada jalan masuk hutan, terdapat soal penjumlahan dan pengurangan materi bentuk akar dengan 6 *level* permainan. Sedangkan pada jalan memasuki kawasan tanaman obat terdapat soal perkalian materi bentuk akar dengan 6 *level* tingkat kesulitan. Serta jalan keluar hutan yang memuat soal pembagian materi bentuk akar. Tidak mudah melalui dan memenangkan *game* ini, pengguna harus lebih cepat dan tanggap agar dapat memenangkan *game* ini.

#### 2. Analisis Tingkat Kesulitan

Meliputi kegiatan analisa tingkat kesulitan *game* Tukarmatika seperti level permainan..

#### 3. *Gameplay*

*Game* Tukarmatika dimainkan oleh satu pemain, hal ini ditujukan untuk siswa dengan tingkatan pemula sampai mahir menguasai materi bentuk akar. Pada tampilan awal terdapat *opening* atau *intro* yang berisi cerita atau *storyboard game* yang akan dimainkan. Untuk fleksibilitas, terdapat tombol skip untuk menuju tampilan menu utama permainan. Pada tampilan menu utama terdapat menu Main, Nilai tertinggi, Bantuan dan Keluar.

#### 4. *Scoring*

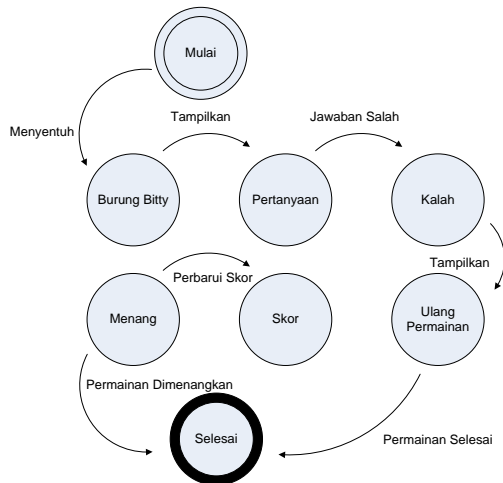
Penscoran dilakukan untuk menganalisa soal yang ada pada *game* tukarmatika agar dapat dinikmati oleh kalangan pelajar matematika khususnya materi bentuk akar.

### B. Metode *Finite State Machine* dan Algoritma A\* (A-star)

#### 1. Metode *Finite State Machine* (FSM)

Metode FSM digunakan untuk membuat rancangan skenario *state* (keadaan), *event* (kejadian) dan *action* (aksi) pada *game* Tukarmatika yang akan dibangun. Keadaan ketika tampil dan jawab pertanya.

Metode FSM juga diimplementasikan pada karakter Andi ketika menolong burung yang sedang terjebak pada ranting ataupun akar pohon hingga tampil pertanyaan. Berikut ini adalah alur keadaan yang akan terjadi.



Gambar 1 FSM Tampil Kuis dan Jawab Pertanyaan

## 2. Algoritma A\*

Algoritma A\* digunakan untuk mencari rute terdekat yang akan mengejar karakter pemain agar tidak mudah menyelesaikan permainan. Algoritma A\* pada *game platform* TukkarMatika ini diimplementasikan pada karakter Kiko dan teman - temannya yang suka mengganggu. Kiko dan teman - temannya adalah NPC yang akan mengejar karakter utama yaitu Andi. Proses penggabungan FSM dan algoritma A\* tergolong sederhana karena memiliki perencanaan logika yang mudah dimengerti tanpa harus menghitung kalkulasi keseluruhan.

0,0	0,1	0,2	0,3	0,4	g=10 h=90 f=100	0,6	0,7	0,8	0,9	0,10
1,0	g=10 h=40 f=50	g=10 h=50 f=60	g=10 h=60 f=70	g=10 h=70 f=80		g=10 h=90 f=100	1,7	1,8	1,9	1,10
2,0	g=10 h=40 f=40	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10
3,0	g=10 h=20 f=30	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10
4,0	g=10 h=10 f=20	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10
5,0		5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10

Gambar 2 Contoh dari Algoritma A\*

## HASIL DAN PEMBAHASAN

### A. Implementasi Antarmuka

#### 1. Antarmuka Menu Utama

Pada tampilan awal *game* terdapat menu Main, menu *Rating*, menu Bantuan, dan menu Keluar. Pada tata letak samping terdapat button menuju sosial media dari *developer*.



Gambar 3 Antarmuka Menu Utama

#### 2. Antarmuka Menu Pilihan Stage

Ketika berada pada menu *Stage* maka akan tampil pilihan *stage* mulai dari Jalur masuk hutan, Jalur kawasan tanaman obat dan Jalur keluar hutan. Ketika User memenangkan *stage 1* maka *stage 2* terbuka, begitupun untuk *stage 3*.



Gambar 4 Antarmuka Menu Pilihan Stage

#### 3. Antarmuka Menu Pilihan Level

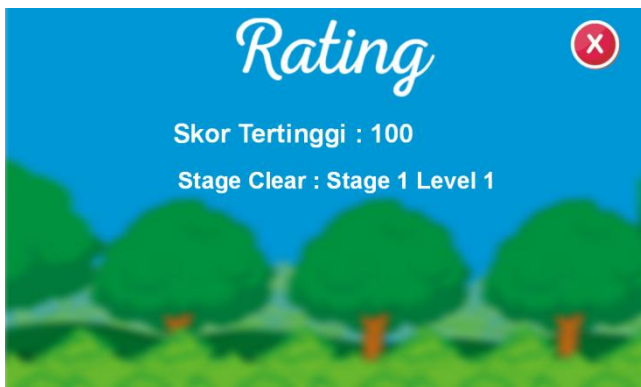
Setelah memilih *stage* maka akan ditampilkan menu *level* yang didalamnya terdapat 6 *level*.



Gambar 5 Antarmuka Menu Pilihan Stage

#### 4. Antarmuka Menu *Rating*

Pada menu *rating*, antarmuka yang ditampilkan adalah skor tertinggi dan *stage* yang aktif saat ini.



Gambar 6 Antarmuka Menu Pilihan *Stage*

#### 5. Antarmuka Menu Bantuan

Didalam antarmuka menu bantuan terdapat beberapa kegunaan atau fungsi tombol dan apa saja tombol cara bermain, dan bagaimana cara menjawab soal



Gambar 7 Antarmuka Menu Pilihan *Stage*

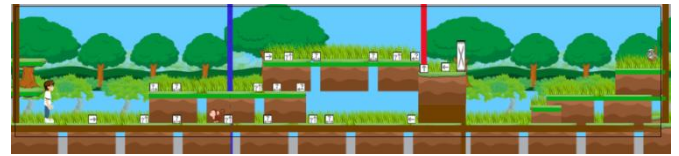
#### 6. Antarmuka Menu Keluar



Gambar 8 Antarmuka Menu Keluar

#### 7. Antarmuka pada *Stage 1 Level 1*

Rancangan pada *Stage 1 level 1* adalah sebagai berikut :









Gambar 9 Antarmuka Permainan *Stage 1 Level 1*

### B. Implementasi Metode *Finite State Machine* dan Algoritma A\*



#### 1. Implementasi *Finite State Machine*.

*Finite State Machine* merupakan aturan dimana pada setiap waktu dapat berubah - ubah atau dinamis. Keadaan yang dinamis inilah yang ada pada *game* ini sehingga mampu memberikan perhatian pada pengguna untuk bermain dan belajar memecahkan masalah. *Finite State Machine* diimplementasikan pada kera atau NPC yang akan mengejar pemain yang nantinya akan digabungkan dengan proses algoritma A\*.

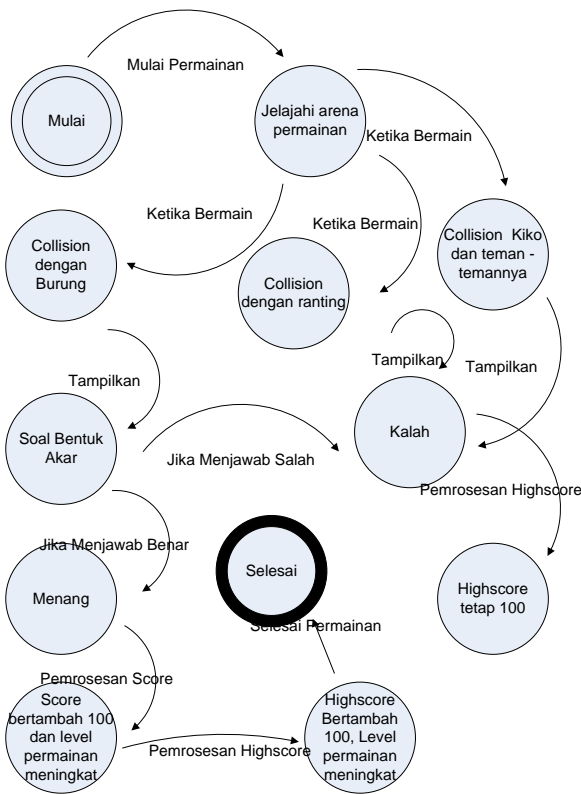
Berikut ini adalah simbol dimana keadaan dapat berubah sesuai aturan yang berlaku pada *Finite State Machine* (FSM) dimana aturan tersebut berlaku ketika NPC menyentuh aturan tersebut. Berikut ini adalah symbol dari aturan tersebut :

-  , simbol untuk menentukan aturan pencarian NPC kearah kanan.
-  , simbol untuk menentukan aturan pencarian NPC kearah kiri.
-  , simbol untuk menentukan aturan pencarian NPC kearah atas atau lompat.
-  , simbol sebagai aturan untuk berhenti dan tidak menuju kearah tersebut.
-  , simbol untuk menentukan aturan pencarian NPC kearah kanan dengan pengecekan apakah *player* berada disebelah kanan NPC.
-  , simbol untuk menentukan aturan pencarian NPC kearah atas atau lompat, dengan melakukan pengecekan apakah *player* berada di koordinat atas atau positif.



- g)  , simbol untuk menentukan aturan pencarian NPC kearah kanan atau kekiri dengan perhitungan apakah posisi *player* condong kea rah kanan atau kiri berdasarkan koordinat.
- h)  , simbol untuk menentukan aturan pencarian NPC kearah kiri dengan pengecekan apakah *player* berada disebelah kiri NPC.

Berikut ini merupakan penerapan metode *Finite State Machine* pada *game* Tukarmatika yang membentuk aturan pada jalannya permainan seperti di bawah ini :



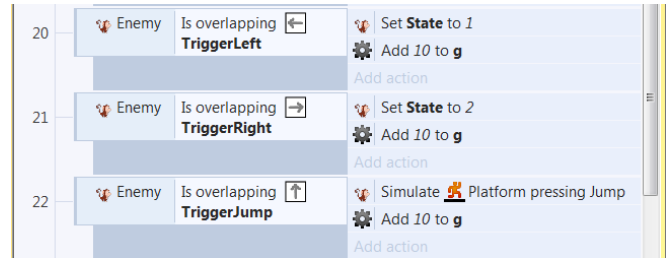
Gambar 10 *Finite State Machine Game* Tukarmatika

**2. Implementasi Algoritma A\***

Algoritma A\* merupakan algoritma pencarian yang kompleks. Pada *game* tukarmatika ini algoritma sangat membantu NPC dalam proses pencarian *player*.

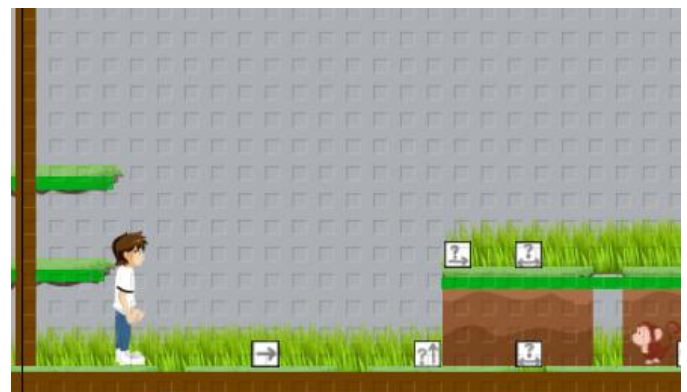
Berikut ini adalah hasil implementasi algoritma A\* pada *game* Tukarmatika :

1. Tentukan nilai awal dari karakter NPC sebagai musuh atau *enemy* dan Andi sebagai *player* dengan posisi tujuan akhir. Setelah itu tentukan nilai g dengan nilai 10 pada aturan horizontal maupun vertikal. Baris program terdapat pada gambar di bawah ini :



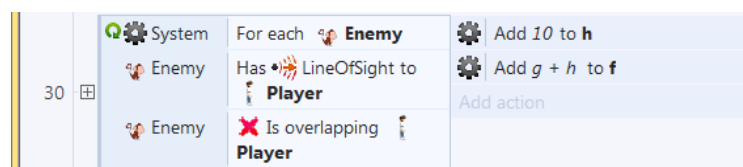
Gambar 11 Aturan A\* ketika Vertikal maupun Horizontal

Implementasi algoritma ditunjukkan pada *gridline* yang ada pada gambar di bawah ini :



Gambar 12 Gridline pada *Game Engine*

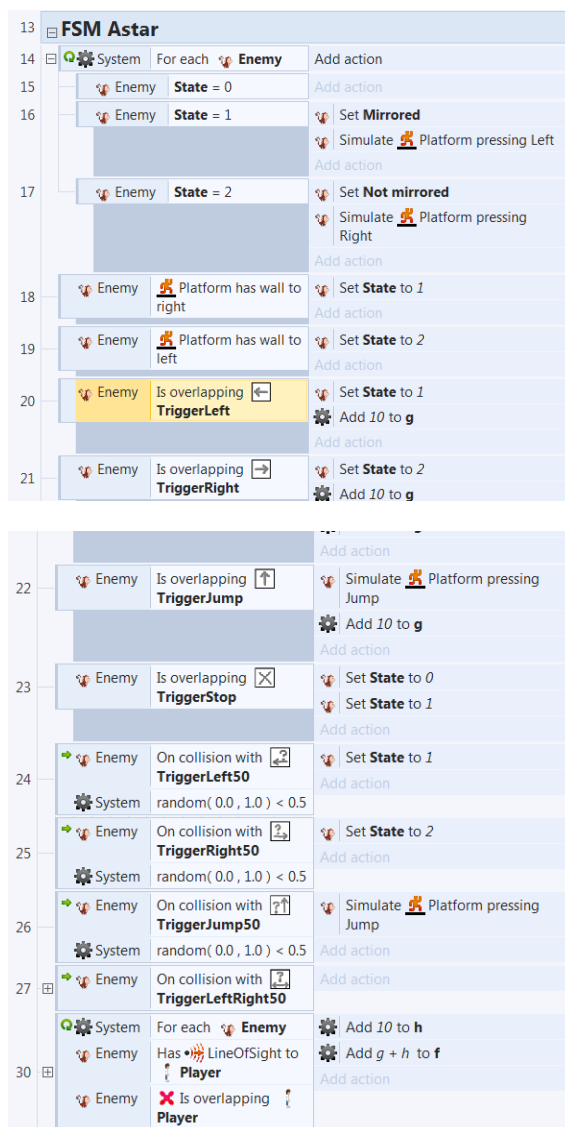
2. Setelah didapatkan hasil dari perhitungan g pada setiap koordinat yang berdekatan, maka proses pencarian nilai h dilakukan dengan jumlah *line of sight* x 10, setelah itu dilakukan penjumlahan pada g dan h sampai menghasilkan nilai f dan membentuk baris program seperti di bawah ini :



Gambar 13 Baris Program pencarian nilai h

### 3. Proses penggabungan ke dalam game

Penggabungan *Finite State Machine* dan algoritma A\* membentuk pola yang dinamis, sehingga pengguna dapat interaktif dalam memainkan game tersebut. Cara menggabungkan FSM dan algoritma A\* adalah dengan menerapkan algoritma A\* terlebih dahulu sehingga NPC dapat mengejar pemain dengan rute terpendek. Jika A\* sudah diimplementasikan, diberikannya aturan berupa FSM agar pengguna tidak terbebani oleh musuh yang terlalu cerdas.



Gambar 14 Baris Program Metode FSM dan Algoritma A\*

### KESIMPULAN

Metode *Finite State Machine* dan Algoritma A\* dapat bersinergi dalam *Game Tukkarmatika* sehingga menjadikan game lebih dinamis. Unsur edukasi yang kuat pada materi bentuk akar matematika juga menjadikan game Tukkarmatika dapat dipakai oleh pengguna yang sudah berumur 15 tahun keatas atau sudah menempuh materi tersebut.

### DAFTAR PUSTAKA

- [1] R. Y. Arindiono, *Perancangan Media Pembelajaran Interaktif Matematika untuk siswa kelas 5 SD*, *Jurnal Sains Dan Seni Pomits*, Vol. 2 No.1, 2013
- [2] Aditya P.(2015, Nov 25).*Android Kuasai 85 Persen Pasar Ponsel Pintar*. (*Kompas Tekno*)[online].Available : <http://tekno.kompas.com/>
- [3] Ernest Adams, *Fundamentals of Game Design : The Definition of a Game*, Berkeley : CA, New Riders. 2010.
- [4] Arief S. Sadiman. *Media pendidikan: pengertian, pengembangan dan pemanfaatannya*. Jakarta: Rajawali Pers . 2010.
- [5] Clark, Donald. (2006, Nov 23). *Games and e-learning* [online] . Available : [www.caspianlearning.co.uk/Whtpcaspian-games\\_1.1.pdf](http://www.caspianlearning.co.uk/Whtpcaspian-games_1.1.pdf).
- [6] Anggra. *Memahami Teknik Dasar Pembuatan Game Berbasis Flash*. Yogyakarta : Gava Media.2008.
- [7] Kim, C.H., Jeong, S.M., Hur, G.T., dan Kim, B.G. *Verification of FSM using Attributes Definition of NPCs Models*, *IJCSNS International Journal of Computer Science and Network Security*. VOL.6 No.7A. July 2006.168-174.
- [8] T.H. Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, *Introduction to Algorithms, Third Edition*.The MIT Press, 2001.
- [9] Indriani W., Nelly. *Membangun Game Edukasi Sejarah Wali Songo*. Teknik Informatika, Universitas Ilmu Komputer Indonesia. Bandung. 2013
- [10] Gullen, Ashley dan Thomas Gullen. (2011, Nov 10). *Construct 2* [online].Available : <http://www.scirra.com/construct2>