

# IMPLEMENTASI ALGORITMA A\* (*A Star*) PADA NPC (*NON-PLAYABLE CHARACTER*) GAME PACMAN MENGGUNAKAN GAME ENGINE UNITY 5 BERBASIS ANDROID

Fawaidul Badri<sup>1</sup>, Muhammad Farih Al Habib<sup>2</sup>

<sup>1</sup>Teknik Elektro, Fakultas Teknik  
Universitas Islam Malang, Indonesia  
e-mail: fawaidulbadri@unisma.ac.id

<sup>2</sup>Teknik Informatika, Fakultas Teknik  
Universitas Maarif Hasyim Latif, Sidoarjo, Indonesia  
e-mail : farihsangkobe@gmail.com

Diterima: 10 Oktober 2020. Disetujui : 22 Desember 2020. Dipublikasikan : 30 Desember 2020



©2020 –TESJ Fakultas Teknik Universitas Maarif Hasyim Latif. Ini adalah artikel dengan akses terbuka di bawah lisensi CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

## ABSTRAK

Memainkan *game* merupakan salah satu cara membuang stress pikiran pada saat tertentu ketika seseorang membutuhkan. Smartphone merupakan perangkat modern yang memiliki banyak *developer* aplikasi, salah satunya adalah *developer game*. Sebelum munculnya *smartphone*, terlebih dahulu muncul *game-centre* yang banyak diminati oleh masyarakat, mulai dari muda hingga tua, dan bahkan kaum hawa pun ikut tertarik. *Pacman* merupakan salah satu nama *game* terkenal pada tahun 1980-an, yang saat ini juga dikembangkan ke *platform android*. Dalam penelitian ini dilakukan implementasi algoritma A\* kepada *NPC Ghost* yang akan mengejar dan memburu *pacman* dalam labirin hingga nyawa *pacman* habis dan *game over*. Dilakukannya implementasi algoritma A\* adalah agar *ghost* dapat menemukan jalur tercepat ke *pacman* agar permainan terasa lebih intense.

**Kata kunci** : pacman, a star, unity, c#, c sharp

## PENDAHULUAN

Permainan Video (*video game*), adalah interaksi antara *user* dengan *interface* melalui sebuah perangkat keras *input-output*. *Video Game* umumnya menyediakan sistem penghargaan seperti skor paling banyak dalam satu waktu dengan memenuhi persyaratan tertentu yang ada dalam permainan tersebut.

Pada zaman sekarang, *game* sudah biasa terdengar. Berbagai macam *game* dapat dimainkan oleh berbagai kalangan masyarakat baik dengan *Personal Computer* (PC) maupun *mobile* (phone). Jenisnya pun beragam, mulai dari *arcade*, petualangan, strategi, *fighting* dan lainnya. Di Indonesia sendiri memainkan sebuah *game* merupakan hal yang dominan dilakukan oleh para pengguna *smartphone* dimana *game mobile* pada masa sekarang ini sudah *portable*, karena *smartphone* dapat dibawa ke manapun dan digunakan di manapun tidak seperti komputer yang harus diam di satu tempat untuk menggunakannya.

*Platform*, dalam ilmu komputer diartikan sebagai serambi atau wahana yang merupakan

kombinasi antara sebuah arsitektur perangkat keras dengan sebuah kerangka kerja perangkat lunak dapat dijalankan dalam perangkat keras tersebut. Contoh jenis *platform* pada ilmu komputer atau IT (*Information and Technology*) atau TI (Teknologi dan Informasi) adalah *linux*, *windows*, *apple macintosh*, *apple ios*, *blackberry*, *symbian*, *java*, hingga yang terbaru dan yang sangat populer saat ini adalah *android*. *Android* sendiri merupakan sebuah *platform* yang dikembangkan dari sistem operasi Linux yang dibuat untuk ponsel pintar (*smartphone*) yang banyak digunakan oleh kalangan masyarakat zaman sekarang.

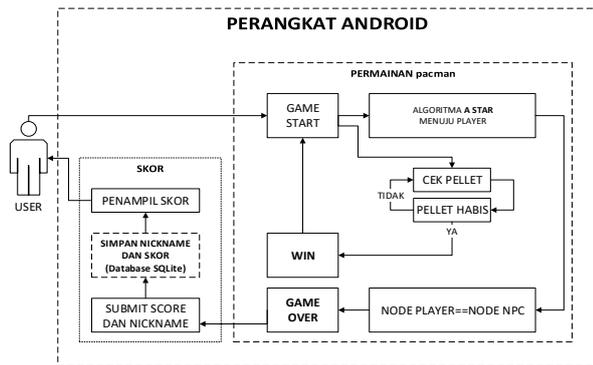
*Game* pada zaman sekarang, memiliki banyak sekali *genre game* (aliran permainan). Salah satu *genre* yang ada yaitu *maze* (labirin), dalam *genre* ini pemain diharuskan untuk menemukan jalan keluar dalam labirin, juga ada *genre* lain yang menerapkan peraturan bahwa *player* diharuskan memakan, atau mengambil seluruh poin dalam labirin tersebut, dengan berbagai NPC (*Non-Playable Character*) yang sedang mengejar *player* tersebut, ketika NPC telah menyentuh *player* maka permainan berakhir. *Game*

*pacman*, adalah salah satu *game* retro klasik yang menerapkan peraturan tersebut, dengan genre *action*, *arcade*, *maze*. *Game pacman* menggunakan algoritma yang diimplementasikan pada *NPC game* tersebut kurang efektif dalam pencarian *player*.

### METODE PENELITIAN

#### Diagram Blok

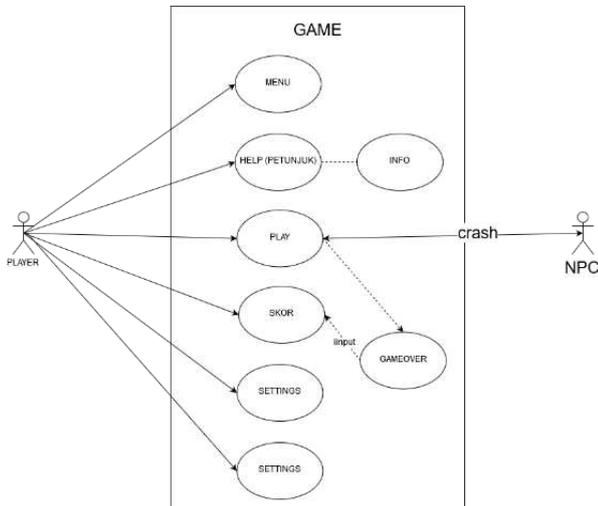
Diagram Blok adalah suatu alat perencanaan yang merupakan inti dari pembuatan modul tersebut. Berikut pada Gambar 1 adalah diagram blok untuk *game pacman android*.



Gambar 1. Sistem Blok Diagram

#### Use Case Diagram

*Use case* menjelaskan interaksi yang terjadi antara *PLAYER*, *SYSTEM*, dan *NPC*. Dari diagram di atas dapat dilihat, terdapat *MENU*, *HELP*, *PLAY*, *SKOR*, *SETTINGS*, *PLAYER*, *NPC*.

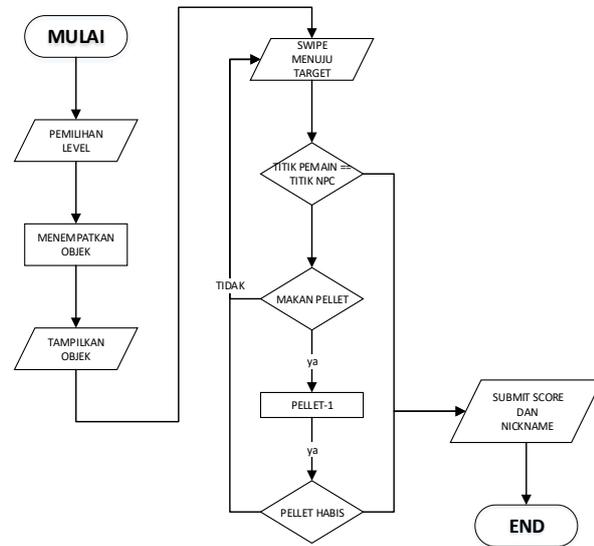


Gambar 2. Use Case Diagram

#### Flowchart Player

Pada Gambar 3, dapat dipahami bahwa pemain melakukan pemilihan level, untuk menggerakkan objek pemain pada *game*, pemain cukup melakukan *swipe* atau mengusap layar pada *swipe area*, ke arah yang diinginkan oleh pemain, hingga karakter atau objek pemain memakan

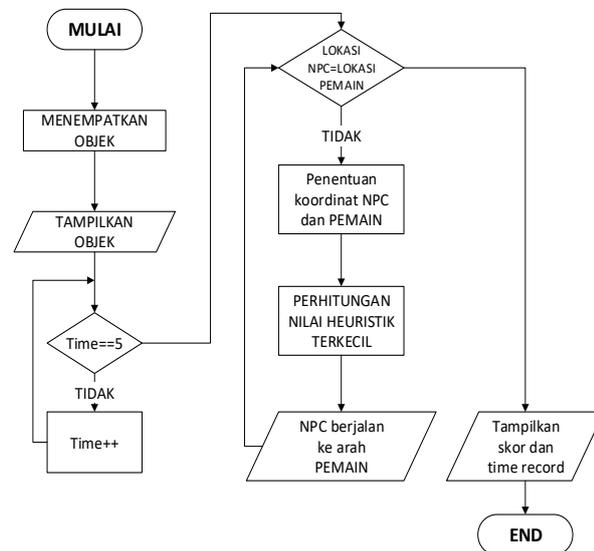
semua pellet yang ada. Permainan akan Berakhir ketika pemain memakan semua pellet yang ada pada labirin, dan juga ketika pemain bertemu dengan *NPC*.



Gambar 3. Flowchart Player

#### Flowchart NPC

Dari Gambar 4, dijelaskan bahwa *NPC* akan berjalan dari tempatnya berada menuju pemain setelah 5 detik sesuai dengan algoritma yang diterapkan, dan akan membuat permainan berakhir ketika *NPC* telah sampai di titik pemain berada.



Gambar 4. Flowchart NPC

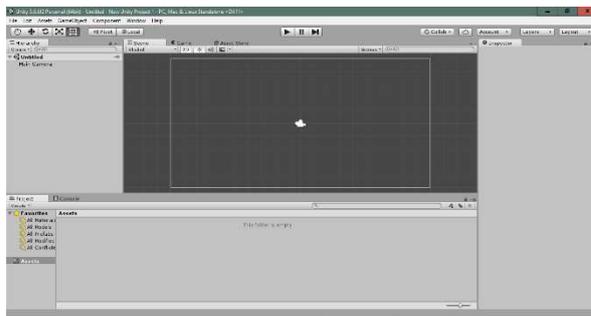
#### Algoritma A\* (A Star)

Algoritma *A Star* atau *A\** adalah salah satu algoritma pencarian yang menganalisa *input*, mengevaluasi sejumlah jalur yang mungkin dilewati dan menghasilkan solusi. Algoritma *A\** adalah algoritma komputer yang digunakan secara luas dalam *graph traversal* dan penemuan jalur serta proses perencanaan jalur yang bisa dilewati

secara efisien di sekitar titik-titik yang disebut *node*. (Reddy, 2013)

**Unity**

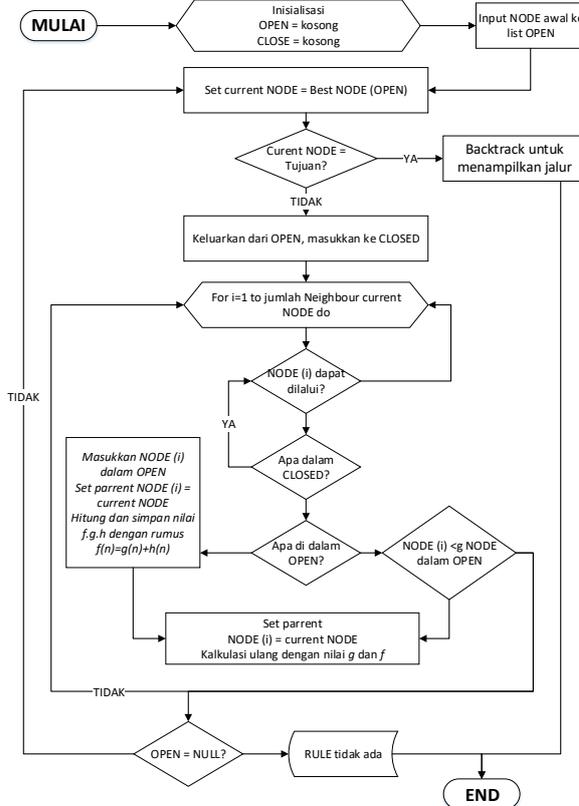
Unity merupakan *Game Engine* yang didesain untuk membuat *game* dengan *interface* yang mudah digunakan oleh *user*, dan dengan *high quality graphics OpenGL* dan *DirectX*, cocok untuk sistem operasi 64-bit. Berikut Gambar 5 merupakan tampilan *visual* dari Unity.



Gambar 5. Tampilan Unity

**Flowchart A\***

Dalam pengembangan *game pacman* ini, penulis menggunakan algoritma *A\** sebagai algoritma pencarian untuk *NPC GHOST*. Berikut Gambar 6 adalah *flowchart A\**. (Tilawah, 2011)



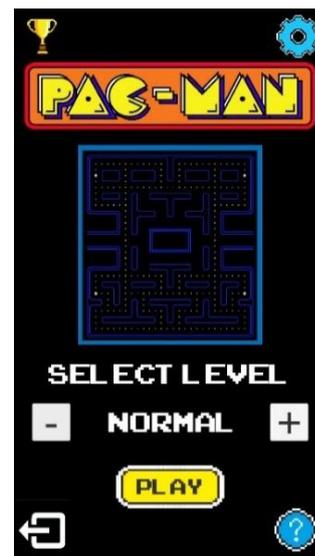
Gambar 6. Flowchar A\*

1. Tambahkan *start node* ke dalam *open set*.
2. Lakukan *loop*:
  - a. Cari nilai *F* terendah dari setiap *node* dalam *open set*. *Node* dengan nilai *F* terendah dijadikan *current node*.
  - b. Masukkan ke *node closed set*.
  - c. Setiap 4 *node* terdekat (*neighbor node*) yang berdekatan dengan *current node* :
    - i. Jika tidak *walkable* atau jika termasuk *closed set*, maka abaikan.
    - ii. Cek dalam *open set*, tambahkan ke *open set* bila belum ada.
    - iii. Jika sudah dalam *open set*, periksa apakah nilai dari *current node* ke *neighbor node* yang lebih baik (menggunkan nilai *G* sebagai nilai ukur). *Node* dengan biaya *G* yang lebih rendah merupakan jalur terbaik. Jika iya, buat *list neighbor node* sebagai *previous node* dari *current node*, dan hitung ulang nilai *G* dan *F* dari *node* ini.
  - d. Stop ketika :
    - i. Tambah target *node* ke dalam *closed set*, dari sini jalur telah ditemukan, atau
    - ii. Gagal menemukan *target node*, dan *open set* kosong. Dalam kasus ini, tidak ada jalan.
  - e. Simpan jalur. Mundur dari *target node*, pergi dari masing-masing *node* ke *previous node* hingga *start point*. Itu adalah jalan Anda.

**HASIL DAN PEMBAHASAN**

**Scene Awal (Main Menu)**

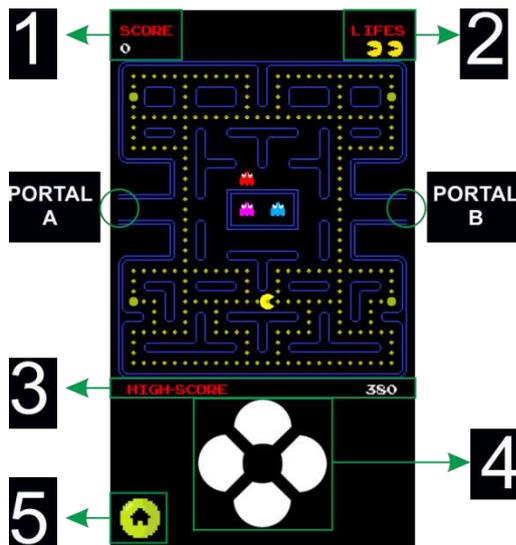
*Main Menu* adalah menu awal setelah *spashscreen* aplikasi dijalankan. Disini *user* dapat memilih banyak menu yang terdapat ddalam aplikasi. Berikut Gambar 7 adalah tampilan *main menu*.



Gambar 7. Main Menu

**Scene Gameplay**

Sesuai dengan rumus algoritma *A\**,  $F(n)=G(n)+H(n)$ , sistem akan memproses langkah-langkahnya seperti berikut:



Gambar 8. Scene Gameplay

Tabel 1. Komponen Scene Gameplay

No	Nama	Fungsi
1	Score	Menunjukkan skor yang didapat dalam permainan saat ini
2	Lives	Indikator nyawa cadangan dari karakter <i>pacman</i> . Terdapat 3 nyawa, 2 nyawa cadangan yang ditampilkan. Dan 1 nyawa yang digunakan oleh pemain.
3	Highscore	Menampilkan skor tertinggi dari permainan.
4	Tombol Navigasi	Tombol ini berwarna putih, terdapat 4 tombol, atas – bawah – kiri – kanan.
5	Home Button	Digunakan untuk kembali ke menu utama.
Portal A		Portal A dan Portal B digunakan untuk jalur pintasan di saat pemain terdesak untuk kabur, maupun di saat pemain ingin pergi ke sisi lain dengan cepat
Portal B		

### Algoritma Pada NPC GHOST

Algoritma yang diterapkan pada musuh *pacman* (NPC GHOST) adalah menggunakan algoritma A\*. Algoritma akan dijalankan oleh *ghost* hanya beberapa detik saja dalam *ChaseMode*, hal ini dilakukan agar tercapainya keseimbangan permainan.

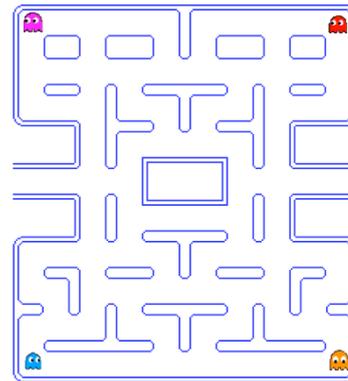
*Ghost* dalam game *pacman* mempunyai 3 mode, dan mode ini dijalankan setelah munculnya tulisan "ARE YOU READY!" di bagian tengah labirin, di bagian atas dan bawah markas *ghost*.

Berikut 3 mode yang terdapat pada *game pacman* ini :

1. *ScatterMode*, mode dimana *ghost* akan berpencar ke arah sudut yang berbeda selama beberapa detik, semakin tinggi tingkat kesulitan yang dipilih oleh pemain maka semakin singkat waktu berpencar *ghost*.

Terdapat 4 *ghost* dalam permainan ini yang masing-masih memiliki warna dan tujuan *ScatterMode* yang berbeda, yaitu :

- a. *Blinky, ghost* berwarna merah. Titik *ScatterMode* berada pada sudut kanan atas labirin.
- b. *Pinky, ghost* berwarna pink. Titik *ScatterMode* ada pada sudut kiri atas labirin.
- c. *Inky, ghost* berwarna biru muda. Titik *ScatterMode* berada pada sudut kiri bawah labirin.
- d. *Clyde, ghost* berwarna jingga. Titik *ScatterMode* berada pada sudut kanan bawah labirin.



Gambar 9. ScatterMode

2. *ChaseMode*, mode dimana *ghost* memburu *pacman*. Mode ini dieksekusi oleh *ghost* setelah waktu *ScatterMode* habis.



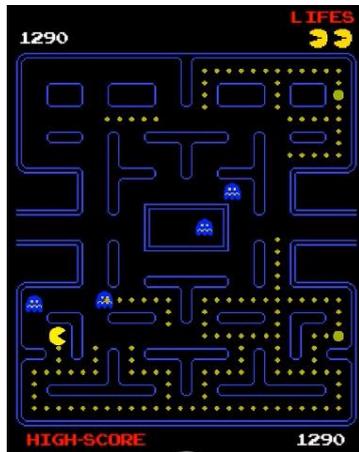
Gambar 10. ChaseMode

Semakin tinggi tingkat kesulitan yang dipilih oleh pemain, maka semakin lama waktu *ChaseMode*. *ChaseMode* dan *ScatterMode* saling bergantian dijalankan oleh sistem *Ghost*. Dalam mode inilah algoritma A\* digunakan untuk mencari jalur terdekat ke *pacman*.

3. *FrightenedMode*, yaitu mode ketika *pacman* telah memakan *SuperPellet*. Dalam mode ini, *ghost* menjadi berwarna biru tua.

Selama *ghost* masih dalam *FrightenedMode*, *ghost* dapat dimakan oleh *pacman* untuk mendapat poin lebih, poin yang didapatkan dari memakan *ghost* ialah 200 poin.

Dalam mode ini, kecepatan *ghost* menurun drastis agar dapat dimakan oleh *pacman*. Semakin tinggi tingkat kesulitan yang dipilih, maka semakin singkat waktu dari *FrightenedMode*.



Gambar 11. *FrightenedMode*

Selama 2 detik sebelum *frightened mode* berakhir, selanjutnya warna *ghost* akan berubah putih dan kembali berwarna biru lagi sampai waktu dari *frightened mode* berakhir, perubahan warna ini menciptakan efek kedip.



Gambar 12. *Ghost Berwana Putih*

Selama warna *ghost* masih berkedip, *ghost* masih tetap dalam *FrightenedMode* dan masih dapat dimakan oleh *pacman*, tetapi pemain perlu waspada, karena ketika *ghost* berkedip-kedip pertanda bahwa *ghost* akan kembali mendaji normal, dan memasuki *ScatterMode*.

## HASIL DAN PEMBAHASAN

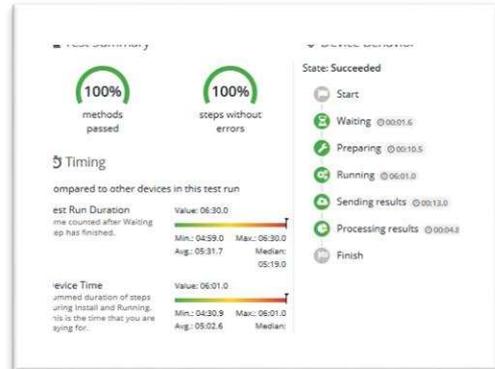
### Uji Coba Online

Uji coba aplikasi dilakukan secara online di website <https://cloud.bitbar.com>. Hasil uji coba aplikasi grafik yang menunjukkan ada atau tidak

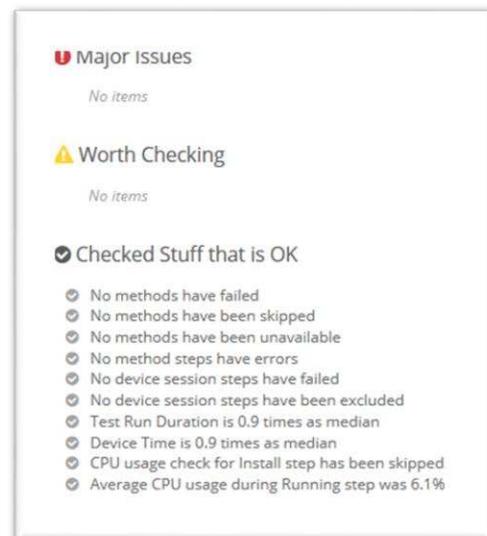
adanya error yang terjadi saat *game* dijalankan. (Widodo & Ahmad, 2018)

### Uji Coba Proses dan Method

Uji coba online ini dilakukan kepada perangkat android LG Google Nexus 4 D820, dengan OS: Android 5.0 Lollipop.



Gambar 13. *Test Summary, Timing Dan Device Behaviour*



Gambar 14. *Issue*

Dari Gambar 13 dan Gambar 14 dapat diketahui bahwa semua *method* dan proses dari *script* tidak ada error, dan uji coba online berjalan dengan lancar.

### Uji Coba Performa

Uji coba online ini dilakukan kepada perangkat android LG Google Nexus 4 D820 dengan spesifikasi sebagai berikut :

1. Processor: Quad-core 2.3Ghz Krait 400
2. Ram: 2048 MB (2GB)

Dari Gambar 15 dapat diketahui bahwa menggunakan *processor* tertinggi hampir mencapai 50% pada *processor quad-core* dengan rata-rata 20%. Dan penggunaan *memory (ram)* tertinggi hanya 100MB-an dari total *ram* 2048MB, dapat

disebut juga 5% dari total ram pada perangkat android.



Gambar 15. Hasil Tes Performa

Dari analisa di atas, dapat disimpulkan bahwa

*game pacman* ini kemungkinan dapat berjalan lancar pada perangkat android dengan *processor* satu tingkat lebih rendah dari *processor quad-core 2.3Ghz Krait*, juga dapat berjalan dengan lancar pada perangkat android dengan *ram* 1024MB mau, dan sedikit berat pada perangkat android dengan *ram* 512MB.

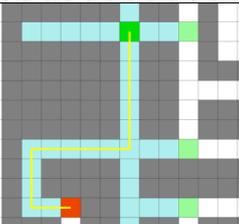
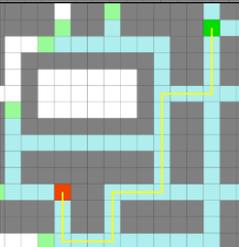
**Uji Akurasi**

Untuk menghitung nilai akurasi sistem ini maka penulis melakukan uji coba sebanyak 3 kali terhadap sistem yang telah dibangun, setiap uji coba akan dilakukan dengan titik yang berbeda, hasil dari uji coba ditunjukkan dalam bentuk gambar dan tabel.

**Hasil Uji Akurasi**

Dari hasil uji akurasi di atas, diringkas pada Tabel 2. Dari tabel hasil uji akurasi didapatkan rata-rata nilai 91,5%. Dari sini dapat disimpulkan bahwa implementasi algoritma *A\** berhasil 91,5%, dan masih memerlukan improvisasi agar akurasi

Tabel 2 Hasil Uji Akurasi

Nomor Uji	Jalur Uji	Akurasi
1		100%
2		100%
3		100%
4		66%

mencapai nilai yang lebih tinggi. (Zikky, 2017)

### PENUTUP

Berdasarkan pembahasan sebelumnya, maka penerapan algoritma  $A^*$  kepada *NPC Ghost* tidak berpengaruh besar dalam permainan, karena bila *NPC* selalu memburu pacman, tidak akan ada yang dapat mengalahkan pacman karena kecepatan pacman jauh lebih unggul ketimbang kecepatan musuh.

Program aplikasi permainan pacman dengan penerapan algoritma  $A^*$  kepada *NPC* ini membutuhkan pengembangan lebih lanjut mengenai desain yang lebih menarik dan penerapan algoritma dapat disempurnakan agar permainan lebih intense

### DAFTAR PUSTAKA

- Reddy, H. (2013). PATH FINDING - Dijkstra's and  $A^*$  Algorithm's. *International Journal in IT and Engineering*.
- Tilawah, H. (2011). Penerapan Algoritma A-star (  $A^*$  ) Untuk Menyelesaikan Masalah Maze. *Jurnal Teknik Informatika, Institut Teknologi Bandung*, 1-5.
- Widodo, W., & Ahmad, I. (2018). Penerapan Algoritma A Star ( $A^*$ ) pada Game Petualangan Labirin Berbasis Android. *Khazanah Informatika: Jurnal Ilmu Komputer Dan Informatika*, 3(2), 57. <https://doi.org/10.23917/khif.v3i2.5221>
- Zikky, M. (2017). Review of  $A^*$  (A Star) Navigation Mesh Pathfinding as the Alternative of Artificial Intelligent for Ghosts Agent on the Pacman Game. *EMITTER International Journal of Engineering Technology*, 4(1), 141-149. <https://doi.org/10.24003/emitter.v4i1.117>

halaman ini sengaja dikosongkan