Penerapan Algoritma Genetika dan Algoritma Ghost Framework pada Game Ms. Pacman

Application of Genetic Algorithm and Ghost Framework Algorithm on Ms. Pacman Game

Yongky Budi Setiawanda[1], Muhammad Khulqi Rasyid[2], Muhammad Jauharul Ramadhan[3], Anggit Dwi Hartanto[4]
[1,2,3,4]Informatika Universitas Amikom, Universitas Amikom
E-mail: ***[1]yongky.se@students.amikom.ac.id, [2]muhammad.rasyid@students.amikom.ac.id, [3]muhammad.0128@students.amikom.ac.id, [4]anggit@amikom.ac.id**

Abstrak

Pacman merupakan salah satu game klasik yang terkenal pada dekade 80-an dan masih sampai sekarang menarik untuk menjadi objek penelitian tentang penerapan berbagai konsep Artificial Intellegent (AI). Tujuan utama dari game ini adalah, mendapatkan poin sebanyak-banyaknya sambil menghidar dari kejaran ghost dan mendapatkan poin tambahan saat memakan ghost setelah mendapatkan pil super. Pergerakan masing-masing ghost ditentukan oleh algoritma DFS dan BFS yang diterapkan pada Ghost Framework. Algoritma tersebut sering digunakan untuk membandingkan algoritma lain dalam hal efektifitas. Penulis memanfaatkan Algoritma DFS dan BFS pada ghost digame Pacman untuk membandingkan keefektifan Algoritma Genetika (GA). Penulis menerapkan Algoritma Genetika pada player Pacman sebagai pengganti control. Terminal akan bertugas untuk mengambil nilai semua kemungkinan arah gerakan di setiap waktu selama permainan. Berdasarkan nilai-nilai ini pengontrol dapat mengontrol MS. Pacman melawan Ghost framework. Dengan desain ini, Penulis mengurangi kompleksitas solusi GA dengan menghapus semua control tindakan tugas manajemen dari system. Setelah melakukan beberapa pengujian, Penulis mendapatkan hasil dimana GA menghasilkan rerata score 8,330 yang lebih tinggi dibandingkan dengan player amatir dan kontroler lain. Dari hasil yang tersedia, Penulis menyimpulkan bahwa kinerja GA sebagai controller MS. Pacman dapat dikatakan baik.

Kata Kunci *— Algoritma Genetika, Pacman, Ghost Framework*

Abstract

Pacman is one of the famous classic games in the 80s and is still interesting to be the object of research on Artificial Intelligence (AI). The main goal of this game is to get as many points as possible while avoiding ghost chases and get extra points when eating ghost after getting super pills. The movement of each ghost is determined by the DFS and BFS algorithms that are applied to Ghost Framework. The algorithm is often used to compare other algorithms in terms of effectiveness. Writer use the DFS and BFS Algorithm on ghost in the Pacman game to compare the effectiveness of the Genetic (GA) Algorithm. The terminal will take the value of all possible direction of movement during the game. Based on these values the controller can control MS. Pacman against Ghost framework. With this design, Writer reduce the complexity of the GA solution by removing all management actions from the control system. After doing some testing, Writer get results where GA produces a higher score compared to other amateur players and controllers. From available results, Writer conclude that the performance of GA as a MS. Pacman controller is good.

Keywords *— Genetic Algorithm, Pacman, Ghost Framework*

Dikirim:28 Desember 2018; Direvisi:5 Januari 2019, 8 Januari 2019; Diterima: 16 Januari 2019

1. PENDAHULUAN

Pacman adalah game arcade original yang telah dipublikasikan oleh Perusahaan asal jepang yaitu Namco pada tahun 1980. Hingga kini, Pacman telah menjadi salah satu game yang paling terkenal sepanjang masa [1].

Prinsip permainan pacman cukup sederhana yaitu, menelusuri maze sambil mencari skor dengan memakan pil kecil. Saat Pacman telah mengumpulkan skor yang cukup, player dapat naik ke level selanjutnya. Dalam proses pengumpulan poin, player harus menghindari ghost. Apabila ghost berhasil menangkap Pacman, permainan akan diulang dari level yang sama. Player juga dapat menambah poin dengan memakan ghost setelah mendapatkan pil super [2].

Dari penjelasan diatas, dapat diketahui bahwa secara umum pergerakan Ghost ada 2, yakni pergerakan saat mengejar dan pergerakan saat menjauhi Pacman. Dalam bergerak, setiap Ghost mempunyai algoritma sendiri karena setiap Ghost memang didesign dengan kepribadian masing-masing. Setiap Ghost mengejar dengan cara yang berbeda-beda. Akan tetapi, algoritma DFS dapat diimplementasikan untuk membuat pergerakan Ghost lebih

menarik. Dengan BFS, Ghost mencari solusi jalur mana yang merupakan jalur terbaik untuk mendekati Pacman ataupun jalur terbaik untuk kabur dari Pacman [3].

Pada tahun 1981, Game sequel tidak resmi dari Pacman yaitu MS. Pacman diperkenalkan oleh Midway. Dibekali dengan labirin baru, efek suara baru, dan perubahan mendetail lainnya. Selain itu pola pergerakan Ghost juga mengalami perubahan sehingga susah untuk diprediksi oleh player manusia, seperti pada game originalnya [4]. Meskipun prinsip dasar permainannya sangat sederhana, sulit untuk mendapatkan banyak poin karena sifat yang susah diprediksi. Pola pergerakan acak dan susah diprediksi tadi menjadikannya sebagai tempat favorit untuk uji coba penelitian dengan menggabungkan konsep Kecerdasan Buatan yang diimplementasikan pada Controller.

Algoritma Genetika pertama kali dikembangkan oleh John Holland pada tahun 1970-an di New York, Amerika Serikat. Dia beserta murid-murid dan teman kerjanya menghasilkan buku berjudul "Adaption in Natural and Artificial Systems" pada tahun 1975 [5]. Algoritma Genetik khususnya diterapkan sebagai simulasi komputer dimana sebuah populasi representasi abstrak (disebut kromosom) dari solusi-solusi calon (disebut individual) pada sebuah masalah optimisasi akan berkembang menjadi solusi-solusi yang lebih baik [6]. Secara tradisional, solusi-solusi dilambangkan dalam biner sebagai string '0' dan '1', walaupun dimungkinkan juga penggunaan penyandian (encoding) yang berbeda. Evolusi dimulai dari sebuah populasi individual acak yang lengkap dan terjadi dalam generasi-generasi. Dalam tiap generasi, kemampuan keseluruhan populasi dievaluasi, kemudian multiple individuals dipilih dari populasi sekarang (current) tersebut secara stochastic (berdasarkan kemampuan mereka), lalu dimodifikasi (melalui mutasi atau rekombinasi) menjadi bentuk populasi baru yang menjadi populasi sekarang (current) pada iterasi berikutnya dari algoritma [7].

Algoritma genetika merupakan kelas khusus dari algoritma evolusioner dengan menggunakan teknik yang terinspirasi oleh biologi dan rekombinasi atau crossover [7]. Seperti dalam evolusi, banyak dari proses algoritma genetika yang bersifat acak, namun tentu saja teknik optimasisasi ini memungkinkan satu untuk mengatur tingkat pengacakan dan tingkat control. Algoritma ini jauh lebih kuat dan efisen daripada pencarian acak dan algoritma pencarian lain [6]. Namun tetap saja akan membutuhkan informasi ekstra tentang masalah yang dibutuhkan. Fitur ini memungkinkan mereka untuk menemukan solusi masalah yang tidak dapat diatasi oleh metode pengoptimalan lainnya karena kurangnya kontinuitas, derivative, linearitas atau fitur lainnya [8].

2. METODE PENELITIAN

2.1. Tinjauan Pustaka

Koza, adalah orang pertama yang mulai mengembangkan implementasi dari genetika dengan menggunakan genetic programming (GP) pada controller di game pacman. Prioritas tugas sebagai salah satu karakter utama pada project penelitian tersebut [9].

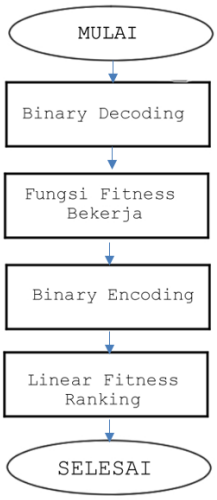
Di dalam buku yang ditulisnya, Koza mendefinikasikan satu set 15 operator GP primitive, untuk pengambilan informasi (Ms. Distance-To-Pill) dan Pacman Control (Ms. Advence-To-Pill). Pada Pacman Control system dapat secara otomatis menghitung rute terbaik (terpendek) ke target yang telah di tentukan (Ghost atau Pill power). Jadi, karakter Pacman dapat dikendalikan menggunakan gerakan abstrak (tingkat tinggi) berdasarkan kondisi permainan.

Pada paper lain Alhejali dan Lucas [10] mengembangkan hasil penelitian Koza dengan menambahkan terminal kerja pada GP, seperti Is-In-Danger atau To-Safety, di mana menggunakan tingkatan abstrak yang lebih tinggi lagi utnuk operator dapat mengambil informasi dan control gerakan Pacman. Penelitian ini didasarkan pada fitur yang lebih lengkap hamper mirip dengan video arcade game yang asli di bandingkan dengan MS. Pacman. Mereka melaporkan jika hasil yang dicapai program pengontrol mereka memiliki keunggulan daripada versi terdahulunya, yang kemudian menginspirasi paper ini untuk terus membuat controller baru dengan metode yang baru juga.

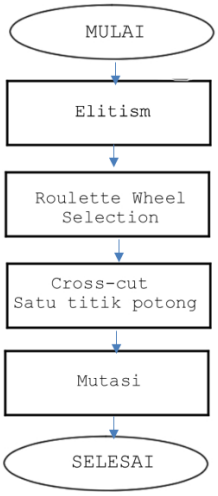
Terdapat juga metode lain menggunakan teknik Evolutionary Computation (EC) yang telah diterapkan dengan baik. Dalam [11] sebuah system yang diimplementasikan pada pengontrol jarigan saraf yang berkembang menggunakan algoritma genetika sederhana. Menghasilkan MS. Pacman yang telah diterapkan dengan algoritma genetika namun bukan pada kontrollernya [12].

Terdapat juga sumber dari paper lain Matthias [13] yang menggagas reactive control menggunakan Genetic Programming, yang dapat menghapus semua action control yang tidak diperlukan lagi untuk mempermudah management task pada system, sehingga controller dapat bergerak lebih complex dan cerdas bahkan jika di bandingkan dengan pemain amatir, controllernya berhasil mendapatkan score yang lebih tinggi.

2.2. Algoritma Genetika



Gambar 1. Sub program Evaluasi



Gambar 2. Sub-program Reproduksi

Algoritma Genetika memang didesain untuk mensimulasikan apa yang dimaksud proses biologi atau alami, yang pada prakteknya memang sangat berkaitan dengan biologi. Bagaimanapun entitas dari terminology ini menunjuk pada algoritma genetika yang lebih

sederhana daripada proses biologi pada aslinya [14]. Dasar dari algoritma Genetika sendiri adalah sebagai berikut:

1. Fungsi fitness untuk optimasisasi
2. Populasi dari kromosom
3. Memilih dan menyeleksi kromosom mana yang akan bereproduksi
4. Crossover untuk memproduksi kromosom yang menjadi generasi berikutnya
5. Mutasi kromosom acak di generasi berikutnya

Di alam bebas, terdapat variasi kromosom pada individu dalam populasi. Variasi terebut mempengaruhi kemampuan berkembang biak dan bertahan hidup suatu individu. Individu yang dapat beradaptasi dengan baik akan berkembang biak dalam taraf yang tinggi, begitu pula sebaliknya. Begitulah bunyi dari teori evolusi Darwin. Dalam kurun waktu yang lama dan melalui beberapa generasi, populasi akan menjadi hanya berisi individu yang memiliki kromosom yang memberikan individu tersebut kemampuan bertahan hidup sehingga dapat berkembang biak. Oleh karena itu, pada akhirnya struktur indiviu tersebut berubah karena seleksi alam. Ketika penulis dapat melihat perbedaan dalam struktur yang dapat diamati dan diukur yang muncul dari perbedaan kebugaran, penulis dapat mengatakan bahwa populasi telah berevolusi. Dalam proses ini struktur muncul dari kebugaran.

Ketika mempunyai sebuah populasi berisi banyak individu, keberadaan beberapa perbedaan pada individual yang mempengaruhi tingkat bertahan hidup tidak dapat dihindari. Maka dari itu, dakam praktek, keberadaan syarat pertama dari empat syarat utama (kemampuan untuk berkembang biak) menjadi sangat penting untuk memulai proses evolusi.

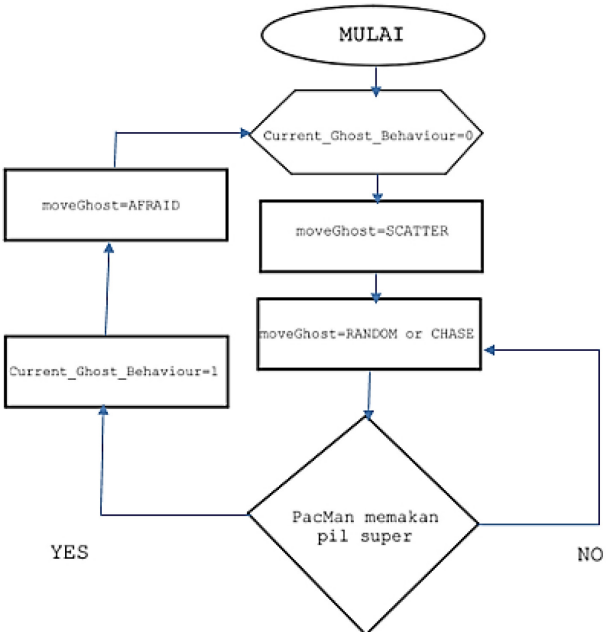
Buku John Holland yang berjudul Adaptation in Natural and Artificial Systems (1975) memberikan pandangan umum untuk mengamati seluruh sistem yang dapat beradaptasi (baik buatan maupun alami) dan kemudian menunjukkan bagaimana proses evolusi dapat diterapkan pada sistem buatan. Masalah apapun pada umumnya dapat di formulasikan pada ketentuan genetika. Setelah diformulasikan dalam ketentuan tersebut, permasalahan tersebut dapat diselesaikan dengan yang penulis sebut "algoritma genetika"

Algoritma genetika adalah algoritma matematika paralel tingkat tinggi yang mengubah sebuah set (populasi) dari objek matematika individu (biasanya fixed-legth character string yang didasarkan atas kromosom), dimana setiap populasi memiliki nilai kebugaran tersendiri, menjadi populasi baru. (generasi selanjutnya) menggunakan operasi yang didasarkan pada prinsip Darwinian dimana yang paling cocok akan bertahan hidup dan setalah melalui operasi genetik (rekombinasi seksual).

2.3. Ghost Framework

Kerangka khusus yang membiarkan Ghost bergerak secara Random, Chase dan Scatter. Ketika Pacman berhasil memakan pil power, Ghost akan otomatis berubah status ke Afraid yang merupakan status untuk kabur dari kejaran Pacman. Fungsi determineGhostBehaviour() dalam Pacman.java berisi kode yang dapat secara otomatis mengubah status hantu. Hal ini akan mengubah variable global yang disebut CURRENT_GHOST_STATE. Fungsi moveGhost() di Environtment.Java berisi perilaku actual yang terkait dengan status dari Hantu tersebut.

Penting untuk diketahui semua hantu memiliki status perilaku mereka sendiri, Jika status ini diatur ke -1, hantu akan menganggap status hantu global hingga langkah berikutnya. memanggil fungsi propagateGhostBehaviour() yang akan mengatur status semua hantu menjadi -1. Kerangka ini dibentuk untuk memungkinkan hantu memiliki pergerakan yang berbeda. Misalnya, status awal hantu adalah SCATTER, yang akan mengirim setiap hantu ke sudut maze. Segera setelah hantu mulai berhamburan, status hantu akan berubah menjadi RANDOM atau CHASE. Namun hantu individu akan terus menyebar, sampai mereka mencapai target mereka. Kemudian status individual mereka akan diatur ke-1, jadi pada langkah waktu berikutnya mereka akan menganggap status hantu global dan menjadi RANDOM atau CHASE. Cara kerja Ghost Framework ditunjukkan pada Gambar 3.



Gambar 3. Cara kerja Ghost Framework

2.3.1. Configuration

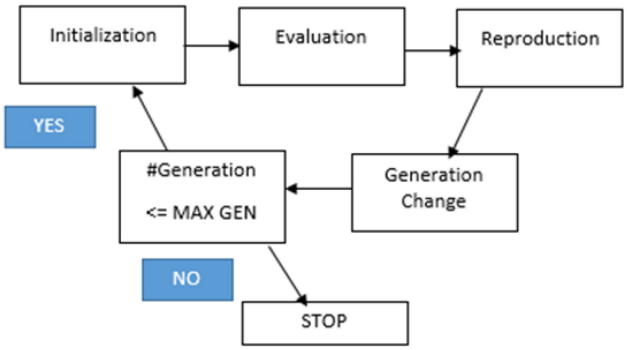
Berikut langkah- langkah dalam konfigurasi:

1. loadFile - Beberapa pengaturan yang mempengaruhi struktur jaringan dan mengubahnya dilain waktu akan membuat simulasi tidak dapat diandalkan lagi
2. pathlength – Berbagai input algoritma yang dimasukkan perlu untuk mengetahui Panjang jalur maksimum antara dua titik di labirin
3. enableTraining – Secara opsional, struktur game secara keseluruhan dapat dirubah dengan, mengatur variable ini menjadi true dan mengaktifkan TrainingRegimen
4. startingGhostState – Ketika hantu dibangkitkan kembali, akan memulai dengan status apa? Anda dapat mengaturnya lewat variable ini GHOST_RANDOM, GHOST_CHASE, GHOST_SCATTER or GHOST_AFRAID.
5. pacmanSpeed – Meskipun permainan ini bersifat continue, penulis perlu untuk mengizinkan jaringan syaraf tiruan untuk langkah-langkah terpisah
6. timeResolution – Ketika visualisasi game di nonaktifkan, kerangka inilah yang akan berjalan mengambil alih secara penuh.
7. eternalGame – Kerangka ini secara otomatis akan membiarkan agent memainkan semua rute labirin ke labirin direktori
8. durationPoWriterrpill – Berapa lama efek pill kekuatan bertahan dapat diatur degnan variable ini
9. effectPoWriterrpill - Hantu bergerak lebih cepat ketika MS. Pacman memakan pil daya dan mereka menjadi takut. Ketika hantu takut, kecepatan normalnya akan dikalikan dengan nilai pengaturan ini. Jadi pengaturannya menjadi 0,5 akan setengah kecepatan.
10. effectEaten - Ketika hantu dimakan, dengan cepat bergerak kembali ke titik bertelur. Selama waktu ini dia tidak dapat berinteraksi dengan objek atau pemain. Ketika hantu dimakan kecepatan normalnya akan dikalikan dengan nilai pengaturan ini. Jadi pengaturannya menjadi 2 kali kecepatan normal.
11. effectPill - Ketika MS. Pacman sedang makan pil, dia melakukan perjalanan sedikit lebih lambat. Ketika ini adalah kasus kecepatan normal MS. Pacman akan dikalikan dengan nilai pengaturan ini. Jadi pengaturannya menjadi 0,5 akan setengah kecepatan.
12. alwaysEnabled - untuk memaksa semua hantu ke keadaan tertentu, daripada membiarkan mereka secara otomatis menggilnya. Untuk mengaktifkannya, set selalu diaktifkan ke true

dan tetapkan GHOST_RANDOM, GHOST_CHASE, GHOST_SCATTER atau GHOST_AFRAID.

13. activeNet - kerangka kerja menggabungkan tiga jaringan neural yang dijuluki DODGE, ROBERT dan HUNT. Dengan mengatur activeNet ke NETWORK_DODGE, NETWORK_ROBERT atau NETWORK_HUNT, hanya jaringan neural yang sesuai yang akan digunakan. Dengan menyetelnya ke NETWORK_AUTO, kerangka kerja akan secara otomatis beralih di antara ketiganya menggunakan fungsi SelectProperNetwork di Pacman.java. Alternatifnya, activeNet dapat diset menjadi NETWORK_ACTION, untuk membiarkan framework menggunakan satu jaringan neural per tindakan (kiri, kanan, atas, bawah).

2.3.2. Implementasi Algoritma

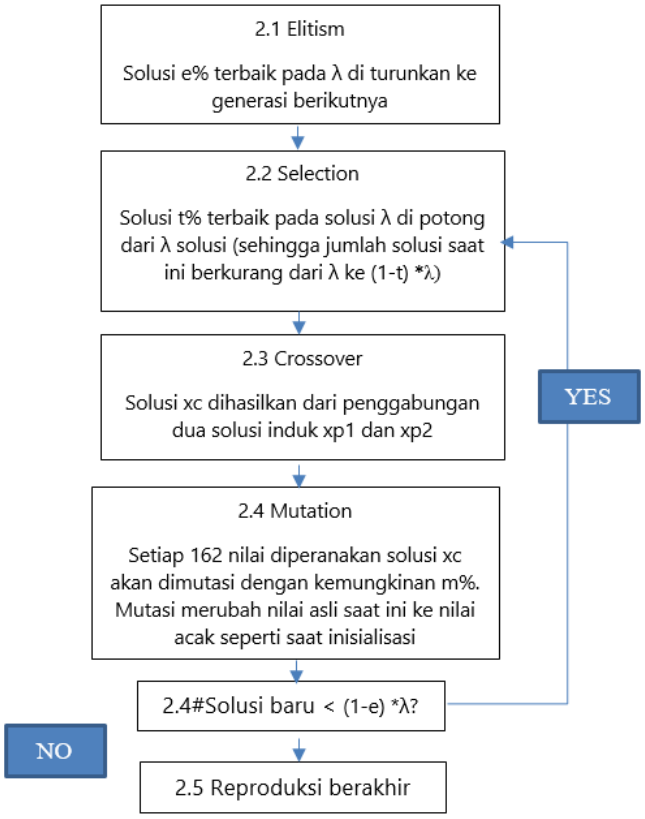


Gambar 4. Alur Implementasi Genetika

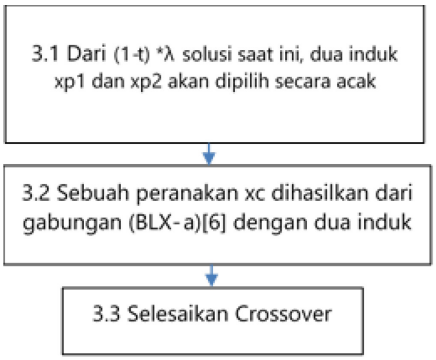
Dapat dilihat pada Gambar 4, algoritma genetika memiliki empat tahap dalam implementasinya. Tahapan tersebut adalah Initialization, Evaluation, Reproduction, dan Generation Change.

Berikut adalah penjelasan masing-masing tahapan.

1. Initialization
   Pada tahapan ini, dihasilkan beberapa solusi secara acak berupa x1, x2, ….x μ.
   Nilai dari xji (i = 1, 2, …, 162 (atau 405); j = 1, 2, …, μ) adalah sampel dari distribusi Gaussian dengan mean = 0 dan S. D=1. Setiap solusi memiliki nilai kebugaran. Solusi-solusi ini kemudian akan dievaluasi pada tahapan berikutnya.
2. Evaluation
   Pada tahapan ini, kesesuaian kebugaran tiap individu di seleksi. Kebugaran disini adalah skor permainan yang dimainkan oleh masing-masing otonom controller. Pada kasus ini adalah pada permainan Pacman.
3. Reproduction
   Pada tahapan ini, solusi dengan nilai kebugaran yang baik akan direproduksi. Gambar 5 dan Gambar 6 menunjukkan langkah-langkah reproduksi dan crossover dengan perspective GA. Solusi peranakan baru (1-e) λ diproduksi dengan menggunakan solusi induk λ. Perhatikan bahwa solusi e*λ disalin dari generasi baru oleh operasi elitism (sehingga proses produksi hanya menghasilkan (1- e) *λ solusi baru saja. Setelah dihasilkan generasi baru, terjadilah perubahan generasi lama ke generasi baru dan memasuki tahapan selanjutnya.



Gambar 5. Alur Reproduksi Genetika



Gambar 6. Alur Crossover Genetika

4. Generation change
   Pada tahapan ini, solusi μ generasi baru selanjutnya dipilih dari populasi solusi μ saat ini dan akan menghasilkan solusi λ yang baru. Terdapat dua perbedaan metode pada seleksi kali ini yaitu (μ+λ) serta (μ, λ). Sebagai generasi baru (μ+λ) memilih solusi μ terbaik di antara solusi (μ+λ) lainnya, sementara (μ, λ) memilih solusi μ terbaik di antara solusi μ terbaru. Pada project ini Penulis menggunakan kedua metode tersebut untuk optimasi menentukan jalur terbaik. Step 2 hingga 5 pada gambar 1 adalah waktu dimana MAX_GEN yang berulang diamana MAX_GEN adalah jumlah generasi yang telah ditetapkan diawal.

3. HASIL DAN PEMBAHASAN

Seperti dalam pendekatan berbasis GA lainnya, kebugaran dari kandidat harus dievaluasi melalui domain-spesifik fungsi kebugaran. Nilai kebugaran dari kandidat disistem yang disajikan dalam paper ini dievaluasi dengan menjalankan 20 generasi dalam 1 game pacman dengan pengontrol dan perhitungan yang sama dan mengambil rata-rata dari ke 20 generasi tadi diakhir. Sebagai evaluasi ke 20 generasi berjalan membutuhkan waktu yang relative lama, ukuran populasi dari satu generasi dibatasi hingga 100 individu. Evolusi dihentikan segera setelah skor tertinggi sepanjang masa dari kesemua generasi yang terbaik hingga tidak berubah selama lebih dari 50.

Parameternya sendiri, seperti probabilitas crossover dan mutase telah dievaluasi dengan mengikuti pendekatan trial dan error.

Pada project kali ini, Penulis menggunakan parameter yang telah di set sebelumnya yaitu:

1. Blend crossover: α =0.5,
2. Elitism: e =10%,
3. Truncation: t =70%, and
4. Mutation: m =7%.
5. Crossover probability: 80%

Berbagai percobaan telah dilakukan dengan parameter yang berbeda dan nilai probabilitas yang berbeda juga. Selama percobaan terdapat masalah utama yang muncul yaitu waktu yang dibutuhkan untuk penentuan kebugaran semua individu dari populasi sangat lama. Misalnya, evolusi satu generasi tunggal 100 individu membutuhkan waktu hingga setengah jam. Oleh karena itu, waktu eksperimen total menjadi sangat terbatas dan dirasa kurang. Demikian bagian ini hanya dapat menganalisis eksperimen yang terdiri dari beberapa simulasi yang telah Penulis jalankan. Sehingga Penulis sarankan lebih banyak waktu untuk eksperimen akan lebih banyak untuk mencapai hasil eksperimen yang lebih baik dan signifikan.

3.1. Ukuran Populasi

Pada umumnya GA memiliki prinsip semakin besar ukuran populasi maka manfaat yang diterima juga akan semakin besar. Misalnya menggunakan ukuran populasi beberapa individu dalam eksperimennya. Parameter ini sangat penting untuk suatu system dengan perhitungan waktu evaluasi kebugaran, seperti yang dijelaskan diatas. Oleh karena itu percobaan pertama mengevaluasi kinerja keseluruhan dari penerapan system berdasarkan ukuran populasi yang berbeda selama proses evolusi terjadi.

Tabel 1. Hasil Percobaan ukuran populasi

Best. Fit	Avg. Fitness	Avg. Gen	Pop. Size
5,171	4,053	123	25 Individuals
5,820	4,055	110	50 Individuals
6,120	4,211	113	100 dividuals

Seperti yang ditunjukkan pada Tabel 1 menunjukkan hasil evaluasi kinerja, berdasarkan (rata-rata) lima evolusi per ukuran populasi. Pada 25 individu rerata generasi mencapai 123 dan berkurang saat individu bertambah menjadi 50. Rerata nilai kebugaran (fitness) mengalami peningkatan dengan bertambahnya populasi dengan dilai tertinggi 4,211 pada 100 individu. Nilai kebugaran terbaik juga bertambah saat populasi bertambah dari 25 menjadi 50 individu, kemudian tetap berada pada nilai 6,120 meskipun nilai berubah menjadi 100.

3.2. Kemampuan Generalisasi

Eksperimen ini mengevaluasi kemampuan generalisasi dari system yang diimplementasikan. Secara umum Penulis ingin algoritma ini dapat bekerja dengan lebih baik tidak hanya pada

penentuan kebugaran yang telah ditentukan sebelumnya, tetapi juga pada kasus yang di mana dapat membantu seorang player atau pengendali pacman yang tidak terlatih untuk memperoleh skor yang lebih baik pada penaklukan ditiap maze.

Seperti yang ditunjukkan oleh tabel 2 berikut ini yang menunjukkan hasil dari individu terbaik untuk tiap 100 simulasi yang berjalan disetiap labiriin (A, B, C, D).

Tabel 2. Performa dari individual terbaik pada semua maze

Maze	Min. Score	Max. Score	Avg. Sc
Maze A	930	1,419	1,138
Maze B	892	1,476	1,173
Maze C	956	1,370	1,163
Maze D	857	1,300	1,078
All Mazes	857	1,476	1,139.5

3.3. Kinerja Controller

Tabel 3 mencantumkan kinerja controller terbaik Penulis sebagai kinerja pengendali MS. Pacman. Perhatikan bahwa pengontrol ini telah dievaluasi pada versi sebelumnya dari simulasi permainan MS. Pacman, perbandingan ini masih sangat terbatas untuk dijadikan sebagai tolak ukur yang utama.

Tabel 3. GA Controller Vs Controller Lain

Controller	Min.Score	Max. Score	Avg.Sc
GA Controller	2,380	14,280	8,330
Controller 1	1,520	13,680	7,600
Controller 2	1,340	12,060	6,700
Controller 3	1,570	10,990	6,280
Controller 4	1,300	13,000	7,150

Secara keseluruhan, seperti yang tercantum di atas, pengontrol Penulis berkinerja terbaik, meskipun pendekatan lain mungkin akan menghasilkan hasil yang lebih baik lagi pada permainan lainnya. Sepertinya, hamper semua eksperimen Penulis ditampilkan serta ditunjukkan oleh pengendali yang sudah Penulis tunjukkan ini untuk pengimplementasikan dalam game pacman, itu adalah hasil yang bagus melalui berbagai permainan yang penting. Dalam eksperimen lain, evolusi individu GA menghasilkan score yang lebih tinggi di bandingkan dengan player amatir manusia lain. Hal ini ditunjukkan pada Tabel 4 berikut ini.

Tabel 4. Best GA individual vs Human Players

Controller	Min.Score	Avg.Sc	Max. Score
GA Controller	2,380	8,330	14,280
Human 1	4,360	13,045	21,730
Human 2	1,620	2,680	3,740
Human 3	1,410	2,970	4,530
Human 4	590	990	1,390

4. KESIMPULAN

Hasil dari penelitian disimpulan bahwa dapat terciptanya sebuah mekanisme di dalam controller game Pacman dengan menggunakan algoritma Genetika dari segi perspektif player untuk membantu manusia atau player dalam menyelesaikan permainan Pacman untuk melawan Ghost Framework yang ada pada Ghost yang menjadi musuh di dalam game ini. Dari beberapa

kali trial di atas dapat dilihat bahwa controller menggunakan algoritma genetika berhasil mencatat score rata-rata 8,330 dibandingkan dengan controller lain, serta algoritma genetika melawan player manusia atau Human 1-4 berhasil mencatatkan score rata-rata terbaik yaitu 8,330. Penelitian ini menunjukkan bahwa algoritma genetika dapat diimplementasikan pada controller game Pacman untuk melawan Ghost framework yang ada di Ghost atau musuh Pacman dan berhasil mencatatkan score yang lebih baik dari Human Controller atau Controller lain.

5. SARAN

Untuk meningkatkan kinerja serta menyempurnakan penelitian yang telah dibuat ini, maka peneliti memberikan saran sebagai berikut:

1. Dalam perhitungan Algoritma Genetika, dapat memperbanyak nilai variable (kromosom, mitasi, crossover dll) karena hal itu akan sangat berpengaruh pada hasil akhirnya nanti serta diharapkan untuk menambah volume uji coba untuk mendapatkan nilai yang lebih bervariasi.
2. Pengembangan selanjutnya, system diharapkan dapat menggabungkan metode algortima genetika dengan metode yang lainnya untuk mendapatkan hasil yang lebih baik lagi, serta implementasi pada game yang lebih banyak lagi.

DAFTAR PUSTAKA

[1] M. J. Wolf, Ed., 11 2007. *The Video Game Explosion: A History from PONG to PlayStation and Beyond*. Greenwood.

[2] Müller, M., 2010, Pacman Writerreldrecord beklonken en het hele verhaal (in Dutch), *https://web.archive.org/web/20120227102904/http://www.ng-gamer.nl/game-nieuws/11117_pacman-wereldrecord-beklonken-en-het-hele-verhaal/*.

[3] Frayudha, A. D., As'ari, T. H., Implementasi Genetic Algorithm untuk pergerakan Ghost di permainan Pac Man, UIN Malang, *https://www.academia.edu/15458673/Implementasi_Genetic_Algorithm_untuk_pergerakan_Ghost_di_permainan_Pac_Man*, diakses tanggal 27 Desember 2018.

[4] Bell, N., Fang, X. H., Hughes, R., Kendall, G., O'Reilly, E., Qiu S. H., 2010, Ghost direction detection and other innovations for Ms. Pacman, *2010 IEEE Symposium on Computational Intelligence and Games (CIG)*, Copenhagen, September.

[5] Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems, An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, A Bradford Book, Cambridge.

[6] Kinnear, K. E. (1994). Kinnear, K., E, 1994, A Perspective on the Work in this Book. In K. E. Kinnear (Ed.), Advances in Genetic Programming, MIT Press, Cambridge.

[7] Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Writersley Publishing, Boston

[8] Carr, J, 2014, An Introduction to Genetic Algorithms, *https://www.semanticscholar.org/paper/An-Introduction-to-Genetic-Algorithms-Carr/e9f8d49686a4c8d99d0a5ceba85c4508c30d57c4*

[9] Koza, J. R., 1992. *Genetic Programming On the Programming of Computers by Means of Natural Selection*, A Bradford Book, Cambridge.

[10] Alhejali, A. M., Lucas, S. M., 2010, Evolving diverse Ms. Pac-Man playing agents using genetic programming, *UK Workshop on Computational Intelligence (UKCI)*, Colchester, 8-10 September.