

IMPLEMENTASI ALGORITMA GENETIKA DALAM OPTIMASI PERFORMA TRUK SAMPAH MENGGUNAKAN APLIKASI TRASH QUEEN

Arief Kelik Nugroho^{*1}, Muhammad Naufal Faza²

^{1,2}Universitas Jenderal Soedirman, Kabupaten Banyumas
Email: ¹arief.nugroho@unsoed.ac.id, ²muhammad.faza@mhs.unsoed.ac.id
^{*}Penulis Korespondensi

(Naskah masuk: 24 Juni 2021, diterima untuk diterbitkan: 27 Februari 2023)

Abstrak

Kebutuhan akan kendaraan dapat dibilang sebagai salah satu kunci dari berjalannya ekonomi dunia. Akan tetapi, efisiensi desain spesifikasi kendaraan masih menjadi topik hangat di kalangan desainer otomotif. Hal ini karena meskipun ada jutaan kendaraan di seluruh dunia, tidak semua desain kendaraan dapat menunjukkan performa yang optimal di segala keadaan. Aspek spesifikasi kendaraan yang memiliki pengaruh paling besar antara lain daya mesin, ukuran ban, dan berat kendaraan. Oleh karena itu, dibutuhkan suatu optimasi desain spesifikasi kendaraan yang hanya membutuhkan faktor ukuran ban dan berat kendaraan, di mana berat kendaraan tersebut dapat direpresentasikan oleh dua faktor besar, yaitu kapasitas kargo dan kapasitas bahan bakar. Dalam penelitian ini ditelusuri kemungkinan optimasi spesifikasi truk sampah berdasarkan faktor-faktor tersebut dalam tujuannya mengumpulkan sampah dan kembali ke sentra pengumpulan sampah tanpa kehabisan bahan bakar menggunakan aplikasi Trash Queen. Trash Queen memanfaatkan algoritma genetika untuk menjalankan simulasi secara berulang-ulang hingga didapatkan solusi yang optimal. Truk sampah yang kehabisan bahan bakar tanpa mampu mengantarkan sampah akan dianggap gagal karena tidak mampu mencapai tujuannya. Pada penelitian ini, ditemukan bahwa dalam 30 generasi fitness terbaik tiap generasi telah naik sebanyak 68.4% dengan trend ukuran ban yang makin kecil, kapasitas bahan bakar yang makin kecil, dan kapasitas kargo yang makin besar.

Kata kunci: *algoritma genetika, aplikasi, kromosom, kargo, Trash Queen*

TRASH TRUCK PERFORMANCE OPTIMIZATION WITH GENETIC ALGORITHM USING THE TRASH QUEEN APPLICATION

Abstract

The need for vehicles can be said to be the key in the continuity of the world's economy. Even so, the efficiency of vehicle design specifications remain a hot topic among automotive designers. This is caused by the many millions of vehicles all across the globe, yet not all of them are able to perform to factory standards at optimal efficiency due to variations in different situations. The parts that have the most significant roles in a vehicle's specification with respect to efficiency includes engine power, wheel size, and the weight of the vehicle itself. That is why a design optimization where it only accounts for the easier to find parts of a vehicle's specification is needed, the wheel size and the weight of the vehicle which can be represented by two major factors: cargo capacity and fuel capacity. This research aims to explore the possibilities of optimizing a trash truck's specifications in its conquest to collect trash and return to a trash collecting centre to deliver them without running out of fuel using the Trash Queen app based on those factors. Trash Queen utilizes genetic algorithm to run simulations repeatedly until an optimal solution is obtained. Trash trucks that run out of fuel before being able to deliver any trash will be considered as failed trucks due to being unable to accomplish their set goal. In this research, it was found that in just 30 generations, the best fitness result of each generation has risen by 68.4% with wheel sizes trending to a smaller size, fuel capacity to a smaller size, and cargo capacity to a larger size.

Keywords: *application, chromosome, cargo, genetic algorithm, Trash Queen*

1. PENDAHULUAN

Alat transportasi sudah menjadi bagian dari kehidupan sehari-hari masyarakat yang tidak dapat dipisahkan lagi. Tidak dapat dipungkiri lagi bahwa tanpa adanya alat transportasi, tentu ekonomi dan

kehidupan sehari-hari masyarakat akan sangat terhambat. Karena itulah performa dari alat transportasi yang menjadi salah satu poros masyarakat zaman sekarang menjadi sebuah topik yang terus menerus dikembangkan dan diteliti,

dengan harapan semakin berkembangnya performa alat transportasi umum, semakin berkembang juga kondisi ekonomi masyarakat secara keseluruhan.

Algoritma genetika ialah cara pencarian yang di dalam ilmu komputer untuk dapat menentukan penyelesaian optimisasi serta masalah pencarian. Dalam algoritma genetika, suatu populasi solusi yang mungkin terhadap suatu problematika optimisasi dievolusikan menuju solusi yang lebih baik. Setiap kemungkinan solusi memiliki suatu set properti yang dapat dimutasi dan diubah (Darrel Whitley, 1994). Komponen dari algoritma genetika terdiri dari kromosom, crossover, fitness, evolusi, dan populasi (Buddy Septyanto, Setyaningsih, & Bacharuddin, 2017). Implementasi algoritma genetika dalam menentukan konfigurasi spesifikasi kendaraan terbaik dalam melakukan sebuah tugas ini dimodelkan menggunakan truk sampah dalam aplikasi *Trash Queen* yang menunjukkan kemampuan sebuah truk sampah mengumpulkan sampah dan mengantarkannya ke pusat pengumpulan sampah tanpa kehabisan bahan bakar.

Performa dari suatu kendaraan sangat dipengaruhi oleh spesifikasinya, beberapa di antaranya adalah ukuran roda, berat kendaraan, dan kapasitas bahan bakar. Dengan roda yang lebih besar suatu kendaraan dapat mencapai kecepatan lebih tinggi dibanding dengan roda kecil, tetapi juga memiliki massa yang lebih besar sehingga akselerasi kendaraan akan menurun. Selain itu, roda yang lebih besar juga mempersulit kendaraan dalam berbelok, sehingga mengemudikan kendaraan dengan roda yang terlalu besar akan sangat sulit. Berat kendaraan dipengaruhi oleh jumlah bahan bakar tersisa dan beratnya kargo yang diangkut oleh kendaraan tersebut. Berat kendaraan berlebih memiliki dampak negatif terhadap kecepatan, akselerasi, konsumsi bahan bakar, dan kemampuan berbelok kendaraan. Sebuah kendaraan yang memiliki kapasitas kargo yang besar harus didampingi oleh kapasitas bahan bakar yang besar agar tidak kewalahan oleh beban kargo dengan kapasitas bahan bakar sedikit.

Unity adalah *game engine* yang memiliki support *cross-platform* sehingga memudahkan *developer* untuk mengembangkan aplikasi ke berbagai macam platform (R. Arrifahmi, 2019). *Unity* menggunakan *scripting system* dari *framework Mono* yang merupakan implementasi dari *framework open source* milik Microsoft dan dapat menunjang *scripting* dengan menggunakan bahasa C# dan *UnityScript* (J. Manning dan P. Buttfield-Addison, 2017). *Unity* juga memiliki sebuah *asset store* dengan banyak asset gratis. *Unity* juga merupakan sebuah *framework* yang mudah digunakan oleh tim kecil untuk mengembangkan aplikasi (T. T. Nguyen, 2021).

Aplikasi *Trash Queen* merupakan simulator yang dikembangkan oleh Muhammad Naufal Faza pada tahun 2021 di *Unity* untuk mensimulasikan sekumpulan truk sampah untuk mencari truk sampah

dengan performa paling optimal dalam mengantarkan sampah ke *Trash Queen* (Ratu Sampah).

Penelitian ini memiliki tujuan memanfaatkan algoritma genetika dalam mengoptimalkan performa suatu kendaraan untuk mencapai suatu tujuan dan kembali lagi tanpa kehabisan bahan bakar.

2. METODE PENELITIAN

Algoritma genetika menggunakan pemahaman mengenai evolusi alamiah pada cara pemecahan masalah (Arinanta, 2017). Ada tiga aspek paling penting dalam algoritma genetika demi jalannya algoritma dengan lancar, yaitu pendefinisian fitness, pendefinisian dan representasi genetik, dan pendefinisian dan implementasi operasi genetik (Wicaksono, S. A., 2018). Pendekatan yang diambil pada algoritma genetika ialah dengan menghimpun secara acak beragam solusi terbaik yang dipilih di himpunan untuk mendapatkan solusi terbaik, yaitu pada saat kondisi yang memaksimalkan kecocokan yang setelah itu akan disebut dengan nilai fitness. Generasi yang terpilih adalah generasi yang merepresentasikan pembaruan pada populasi sebelumnya (Nugroho and Permadi, 2020). Dengan melewati proses yang berulang-ulang, algoritma genetika mampu mendapatkan penyelesaian yang paling tepat untuk permasalahan yang dihadapi (Yunus & Thobias Rumlaklak, 2018).

2.1. Definisi Individu

Individu yang akan dilakukan penelitian algoritma genetika adalah masing-masing truk sampah yang akan dibangkitkan oleh algoritma.

2.2. Inisialisasi Kromosom

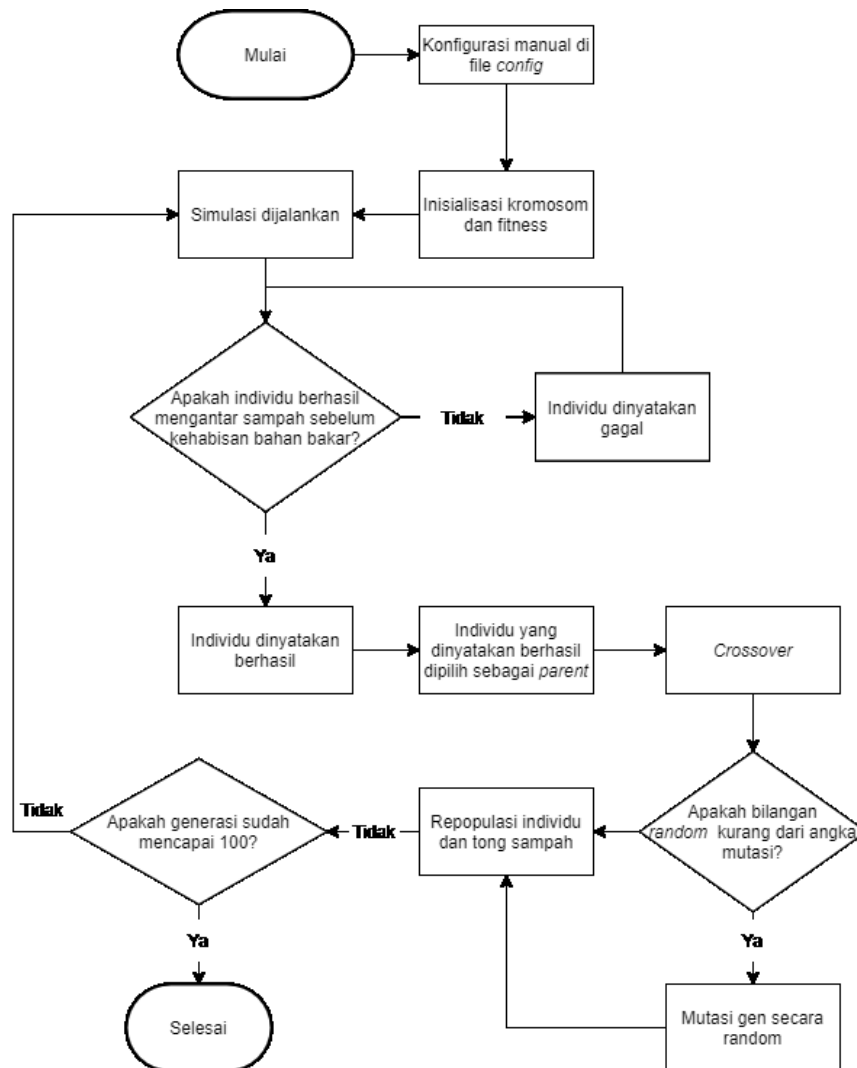
Langkah-langkah algoritma genetika pada umumnya dimulai dengan inisialisasi kromosom (Wang, Y. M. & Yin, H. L., 2018). Nilai kromosom ditentukan secara acak di antara nilai minimum dan nilai maksimum yang ditentukan.

2.3. Seleksi Kromosom

Dicari tiga nilai, yaitu nilai fitness, nilai probabilitas, dan nilai kumulatif probabilitas. Kromosom yang terpilih akan menjadi induk untuk crossover. Seleksi ini dilakukan dengan metode roulette wheel.

2.4. Crossover

Crossover (perkawinan silang) yaitu proses penggabungan dua kromosom untuk menghasilkan kromosom baru. Hasil dari crossover dua kromosom akan menghasilkan dua keturunan yang akan dibangkitkan pada iterasi berikutnya (Eiben, A. E. & Smith, J. E., 2003).



Gambar 1. Flowchart Jalan Aplikasi Trash Queen

2.5. Mutasi

Kromosom dipilih secara acak untuk diubah salah satu gen acak dalam kromosomnya. Kejadian mutasi dipengaruhi oleh mutation rate yang ditentukan terlebih dahulu.

2.6. Hasil Analisis Data

Data yang dihasilkan oleh aplikasi di-export ke dalam sebuah file dengan format CSV (Comma-Separated Value) dan ditampilkan individu dengan fitness paling tinggi setelah generasi paling akhir.

3. PERANCANGAN SISTEM

3.1. Definisi Individu

Dibentuk 1000 individu truk sampah, 200 tong sampah yang menjadi target pengumpulan truk sampah, 5 buah tempat pengisian bahan bakar yang tersebar di tengah dan setiap sudut peta, angka mutasi 0.2, dan sebuah tempat pengumpulan sampah yang dinamakan Trash Queen.

3.2. Inisialisasi Kromosom

Pada saat inisialisasi kromosom, dibangkitkan 3 gen untuk masing-masing kromosom setiap individu secara acak dengan nilai minimal 1.0 dan nilai maksimal 100.0. Kromosom menggunakan bilangan real yang dapat dibidang membentuk suatu *virtual alphabet*, karena metode ini dapat menghasilkan data yang lebih mudah dipahami dibanding menggunakan *bit string* (C. Janikow & Z. Michalewicz, 1991). Gen yang dibangkitkan secara acak adalah gen *wheelSize* yang mempengaruhi ukuran roda, gen *cargoCap* yang mempengaruhi kapasitas kargo, dan gen *fuelCap* yang mempengaruhi kapasitas bahan bakar.

Setiap gen di atas menentukan variabel sementara dari truk. Variabel *cargoSize* tidak dapat melebihi nilai gen *cargoCap* dan variabel *fuelSize* tidak dapat melebihi nilai gen *fuelCap*.

Truk memiliki beberapa aturan sederhana yang diikuti dalam kecerdasan buatan penentu kemudi truk:

- Jika belum memiliki tujuan dan kargo kosong, pilih tong sampah terdekat yang belum dipilih truk lain dan bergerak menuju tong sampah tersebut.
- Jika sudah berada dalam jarak 5 unit dari tong sampah yang dipilih, angkut sampah.
- Jika bahan bakar di bawah 20%, cari tempat pengisian bahan bakar terdekat dan bergerak menuju tempat tersebut untuk mengisi bahan bakar.
- Jika kapasitas kargo penuh sampah, cari Trash Queen dan bergerak menuju Trash Queen. Aturan ini memiliki prioritas lebih rendah dari aturan bahan bakar.
- Jika sudah berada dalam area Trash Queen, kosongkan kargo.

Pada setiap iterasi, semua individu dan tong sampah pada iterasi sebelumnya akan dihapus dan dibangkitkan ulang dengan informasi genetika dari iterasi sebelumnya pada lokasi yang acak sehingga jumlah populasi dan tong sampah tidak berubah.

Tabel 1. Contoh inisialisasi kromosom awal

Kromosom	Ukuran Ban	Kapasitas kargo	Kapasitas bahan bakar
1	10.276	9.719	54.288
2	29.399	1.050	96.513
3	4.931	3.259	93.719
4	4.931	3.259	93.719
5	29.399	1.050	96.513
6	29.399	1.050	96.513
7	76.217	5.733	85.520
8	29.399	1.050	96.513
9	81.582	11.131	74.807
10	76.217	5.733	85.520
11	29.399	1.050	96.513
12	17.753	11.131	74.807
13	29.399	1.050	96.513
14	10.276	9.719	54.288
15	15.291	9.827	57.966
16	29.399	1.050	96.513
17	10.146	5.069	71.133
18	76.217	5.733	85.520
19	76.217	5.733	85.520
20	23.294	5.733	54.288
21	10.146	1.050	85.520
22	12.817	5.069	74.807
23	29.399	3.259	40.104
24	10.276	10.297	54.288
25	29.399	5.733	74.807
26	10.146	1.050	85.520
27	63.042	9.719	93.719
28	12.817	5.069	74.807
29	10.146	1.050	85.520
30	73.464	7.601	46.185

Diperlukan juga sebuah rumus yang mengubah informasi yang dibawa setiap gen dari kromosom menjadi performa nyata di simulator ini. Rumus kecepatan setiap truk adalah:

$$speed = \frac{(wheelSize \times 20.0)}{(\frac{cargoSize}{30.0 + 1.0}) - (\frac{fuelNow}{30.0 + 1.0})} \quad (1)$$

Rumus untuk kecepatan berputar setiap truk adalah:

$$angularSpeed = \frac{(\frac{3000.0}{wheelSize})}{(\frac{cargoSize}{30.0 + 1.0})} \quad (2)$$

Rumus untuk percepatan setiap truk adalah:

$$acceleration = \frac{(\frac{250.0}{wheelSize})}{(\frac{cargoSize}{30.0 + 1.0})} \quad (3)$$

Rumus untuk konsumsi bahan bakar setiap truk adalah:

$$fuelConsumption = cargoSize \times 0.2 \quad (4)$$

Konsumsi bahan bakar setiap truk dibatasi pada minimal 0.0 dan maksimal sesuai dengan kapasitas maksimal bahan bakar setiap truk yang ditentukan pada inisialisasi kromosom.

Rumus untuk pengurangan bahan bakar adalah:

$$fuelNow -= fuelConsumption \times fixedDeltaTime \times timeScale \quad (5)$$

fixedDeltaTime adalah sebuah multiplier di Unity yang menjaga update proses di setiap frame terjaga pada hitungan per detik. timeScale adalah satuan skala waktu dalam Unity yang dapat diubah untuk menjalankan dan menjeda program. Dalam rumus ini didapatkan pengurangan bahan bakar truk setiap detiknya sebanyak fuelConsumption per detik.

Untuk setiap kali truk mencapai target tong sampah dan target Trash Queen, truk harus mengambil dan menyetor sampah. Rumus untuk pengambilan sampah adalah:

$$cargoSize += 10.0 \times fixedDeltaTime \times timeScale \quad (6)$$

Untuk setiap kali target tong sampah dikunjungi oleh truk, sejumlah sampah akan dikurangi dari kapasitas tong sampah tersebut. Rumus pengurangan sampah adalah:

$$TrashAmount -= 10.0 \times fixedDeltaTime \times timeScale \quad (7)$$

Untuk setiap kali truk berhasil mencapai Trash Queen, kargo truk akan dikosongkan dari sampah. Rumus pengurangan sampah di kargo truk adalah:

$$cargoSize -= 10.0 \times fixedDeltaTime \times timeScale \quad (8)$$

Untuk setiap kali truk berhasil menyetor sampah, skor truk yang akan digunakan sebagai nilai fitness juga bertambah. Rumus penambahan skor truk adalah:

$$\text{dumpedTrash} += 10.0 \times \text{fixedDeltaTime} \times \text{timeScale} \quad (9)$$

Ketika bahan bakar truk turun ke bawah 20% dari kapasitas maksimum bahan bakar dan bahan bakar belum habis maka truk akan mencari tempat pengisian bahan bakar terdekat. Ketika truk berhasil mencapai tempat bahan bakar, bahan bakar truk akan diisi sampai kapasitas maksimum bahan bakar secara instan.

3.3. Seleksi Kromosom

Untuk setiap solusi baru yang dibuat, sepasang induk dipilih untuk dikawinkan dari populasi sebelumnya. Keturunan yang dihasilkan akan berbentuk solusi baru yang umumnya memiliki sifat yang mirip dengan induk-induknya.

Kromosom diseleksi dengan metode roulette wheel berdasarkan kemampuan individu untuk mengumpulkan sampah ke Trash Queen tanpa kehabisan bahan bakar. Jika individu kehabisan bahan bakar sebelum mampu mengantarkan sampah sedikit pun, maka individu tersebut dianggap mati dan didiskualifikasi dari seleksi kromosom.

Fitness individu dinilai berdasarkan jumlah sampah yang dikumpulkan oleh tiap individu ke Trash Queen.

$$\text{fitness}[i] = \text{sampah yang terkumpul} \quad (10)$$

$$P[i] = \frac{\text{fitness}[i]}{\text{total fitness}} \quad (11)$$

Probabilitas kumulatif dihitung berdasarkan jumlah probabilitas kromosom dengan probabilitas kumulatif kromosom sebelumnya kecuali kromosom dengan index 0.

3.4. Crossover

Crossover dilakukan dengan sederhana, yaitu dengan memilih satu gen secara acak pada kedua induk dan menukarkan gen-gen tersebut. Kedua keturunan dari crossover ini kemudian dimasukkan ke populasi iterasi berikutnya.

3.5. Mutasi

Mutasi kromosom dilakukan secara acak berdasarkan mutation rate yang dapat ditentukan di file config terhadap gen yang dipilih secara acak.

4. HASIL ANALISIS DATA

Data yang dihasilkan oleh aplikasi ditampilkan dalam bentuk teks di layar setelah iterasi paling akhir dan dalam bentuk file CSV berupa konfigurasi spesifikasi truk paling optimal untuk mengumpulkan sampah.

Pada tabel 2 terlihat hasil nilai fitness terbaik dan data total fitness dari setiap generasi selama 35

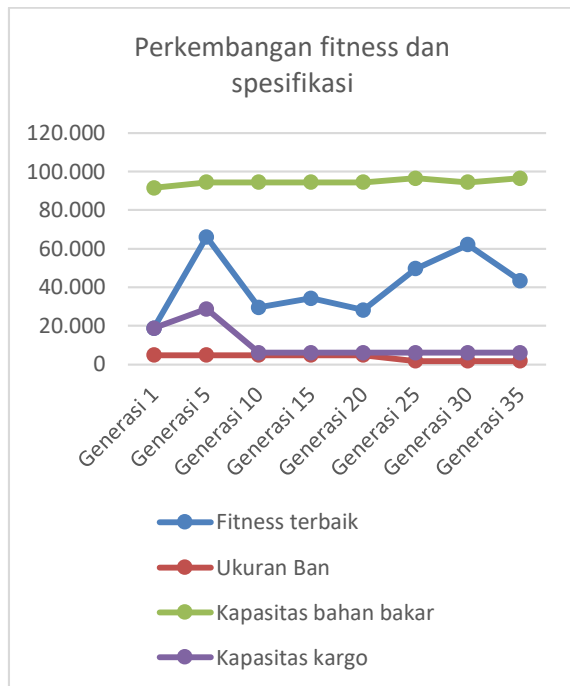
generasi. Dapat dilihat bahwa nilai fitness terbaik dan total fitness dari setiap generasi selalu memiliki trend naik. Nilai fitness terbaik yang meningkat menandakan bahwa algoritma mampu menghasilkan spesifikasi truk sampah yang lebih optimal. Nilai total fitness yang meningkat menandakan bahwa algoritma mampu menghasilkan lebih banyak keturunan dari induk generasi sebelumnya yang secara umum lebih *viable* atau memiliki kemampuan bertahan hidup lebih tinggi. Hasil perkembangan nilai fitness terbaik dan spesifikasi truk dari setiap generasi dapat dilihat pada gambar 2.

Tabel 2. File CSV hasil simulasi di generasi lanjut

Generasi	Fitness Terbaik	Total Fitness
1	18.8	57.8
2	25.2	290.4
3	67.4	688.0
4	49.6	524.0
5	66.0	716.2
6	22.2	643.8
7	28.0	642.3
8	31.0	655.0
9	40.2	695.4
10	29.6	553.0
11	40.2	718.4
12	36.4	674.6
13	34.2	794.6
14	33.2	778.0
15	34.2	674.0
16	31.0	781.8
17	12.3	440.2
18	31.0	808.0
19	49.6	876.4
20	28.2	772.0
21	24.8	670.2
22	49.6	998.6
23	37.2	924.4
24	55.8	884.8
25	49.6	884.6
26	43.4	661.0
27	74.4	1052.6
28	54.2	1083.0
29	43.4	1081.0
30	62.0	1241.0
31	37.2	1085.4
32	55.8	1438.2
33	44.6	1241.0
34	37.2	1243.2
35	43.4	1320.8

3.7. Implementasi Aplikasi

Implementasi aplikasi yaitu aplikasi yang sudah dibangun dan dijalankan untuk membantu proses penentuan konfigurasi spesifikasi truk paling optimal. Dalam aplikasi tersebut dilakukan simulasi selama waktu yang dapat ditentukan oleh peneliti. Hasilnya akan terlihat pada tampilan teks setelah iterasi terakhir.



Gambar 2. Grafik perkembangan fitness dan spesifikasi truk

Pada gambar 3 terdapat penjelasan komponen antarmuka dari aplikasi Trash Queen. Berikut lebih detailnya:

- A. Indikator generasi yang sedang diuji, sisa truk sampah yang masih hidup atau dianggap berhasil, dan indikator waktu sejak dimulainya pengujian pada generasi tersebut. Indikator waktu

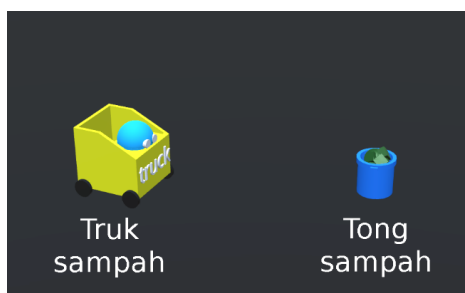
pada bagian ini akan kembali ke 0 pada generasi berikutnya.

- B. Objek ini adalah tempat isi ulang bahan bakar untuk truk sampah. Truk sampah akan mulai bergerak menuju tempat isi ulang bahan bakar ketika bahan bakar truk tersebut sudah di bawah 20% dari kapasitas maksimal.
- C. Ada dua tombol di bagian ini, yaitu tombol *pause* dan tombol *exit*. Tombol *pause* berfungsi untuk menjeda simulasi, dan dapat diklik lagi untuk melanjutkan simulasi. Tombol *exit* digunakan untuk menutup aplikasi Trash Queen.
- D. Objek ini adalah Trash Queen, yaitu sentra pengumpulan sampah yang menjadi tujuan setiap truk sampah ketika bak sampah sudah penuh.
- E. Pada bagian ini terdapat indikator waktu simulasi yang dihitung sejak dimulainya simulasi hingga tercapai generasi 100.

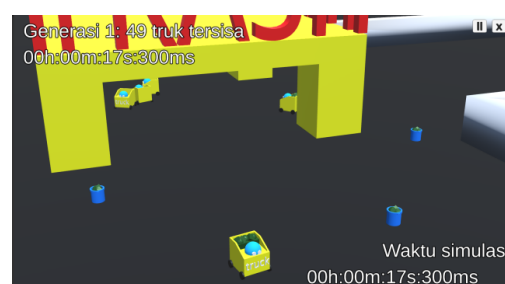
Pada gambar 4 terlihat contoh truk sampah dan target berupa tong sampah. Truk sampah aktif direpresentasikan dengan objek truk berwarna kuning dengan pengemudi yang berwarna biru. Tong sampah memiliki bagian wadah yang berwarna biru dan sampah yang berwarna hijau. Ketika sampah di dalam tong habis, tong sampah akan menghilang.



Gambar 3. Penjelasan komponen antarmuka



Gambar 4. Truk mengumpulkan sampah

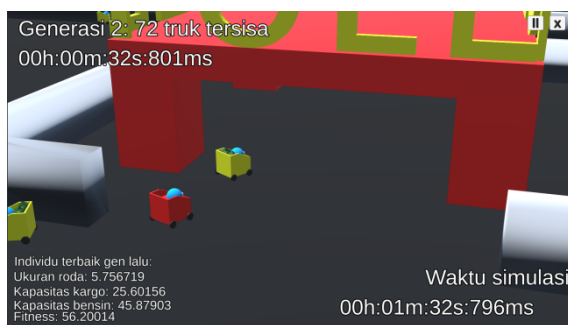


Gambar 5. Truk mengumpulkan sampah

Pada gambar 5 terlihat beberapa truk yang membawa sampah (warna hijau dalam bak) sedang menuju Trash Queen untuk mengantar sampah. Pada gambar 5 terlihat truk yang kehabisan bahan bakar, tapi karena truk ini mampu mengantar sampah sebelum kehabisan bahan bakar, truk ini tetap akan dipilih sebagai parent untuk generasi berikutnya. Truk yang kehabisan bahan bakar tapi mampu mengantarkan sampah ditandai dengan warna merah.

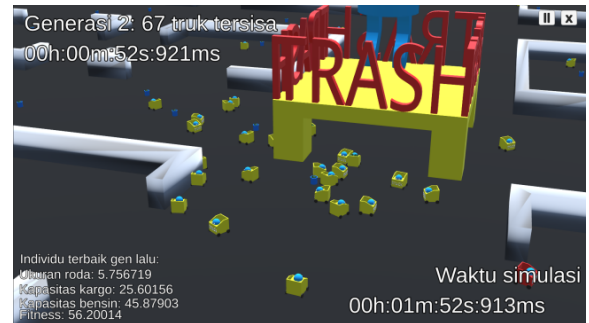


Gambar 6. Truk yang kehabisan bahan bakar tetapi mampu mengantar sampah



Gambar 7. Truk yang hendak mengisi bahan bakar

Pada gambar 6 terlihat truk kuning yang hendak mengisi bahan bakar karena bahan bakar truk tersebut sudah di bawah 20% kapasitas bahan bakar maksimalnya. Di belakangnya terlihat truk merah yang kehabisan bahan bakar tepat di depan tempat pengisian bahan bakar.



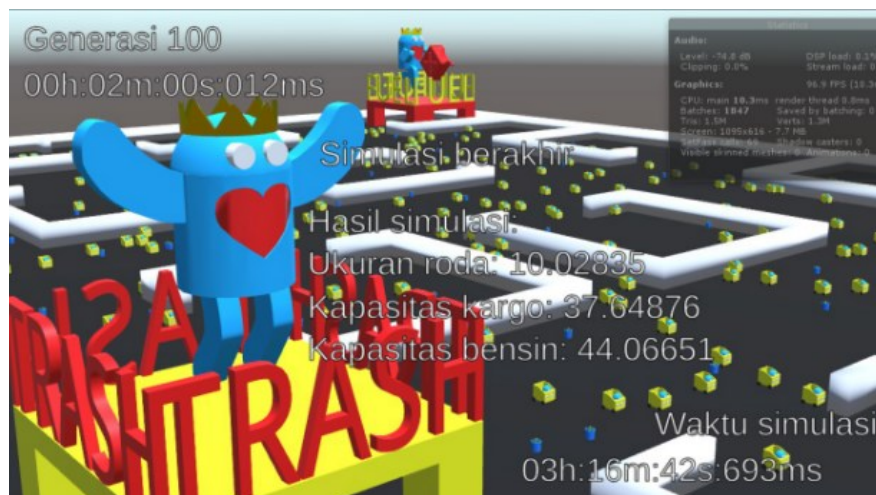
Gambar 8. Truk berlalualang

Pada gambar 8 terlihat bahwa bahkan pada generasi 2, truk sudah lebih cepat dalam memahami bahwa mengambil sampah yang ada di dekat Trash Queen terlebih dahulu lebih efisien.



Gambar 9. Truk mengambil sampah

Pada gambar 9 terlihat beberapa truk yang sedang berusaha mengambil sampah dari satu tong sampah secara bersamaan dan dua truk sampah yang kargonya sudah penuh bergerak menuju Trash Queen. Kedua truk tersebut menghadap ke arah yang berbeda dikarenakan kecepatan berputar kedua truk tersebut berbeda, sehingga salah satu truk membutuhkan area yang lebih luas hanya untuk berputar arah. Dalam gambar ini, truk dengan kargo penuh di kanan tong sampah sudah menghadap ke Trash Queen.



Gambar 9. Hasil di akhir simulasi

Pada gambar 9 terlihat hasil fitness di generasi paling akhir serta informasi genetiknya tampil di layar. Informasi lebih lengkap dapat dilihat di file CSV yang dihasilkan oleh program.

5. KESIMPULAN

Berdasarkan hasil dari penelitian yang telah dilakukan, dapat diambil kesimpulan adalah dengan memanfaatkan algoritma genetika dalam aplikasi simulasi Trash Queen, konfigurasi spesifikasi truk sampah optimal dapat dihasilkan dalam waktu yang relatif singkat. Konfigurasi dapat dikatakan optimal jika dalam simulasi truk sampah dapat menunjukkan performa pengumpulan sampah yang baik dalam batas waktu yang telah ditentukan. Dapat terjadi konfigurasi kurang optimal dalam kasus semua truk sampah penuh tanpa mampu mengumpulkan sampah. Pada kasus ini aplikasi memberikan pesan error agar peneliti mengulang simulasi.

DAFTAR PUSTAKA

- ARINANTA, FIRMANDA R. 2017. Penerapan Logika Fuzzy dengan Metode Logika Fuzzy Tsukamoto dalam menentukan Pertimbangan Berwisata di Pulau Lombok. Skripsi. Mataram. Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram.
- ARRIFAHMI, R. 2019. PEMBANGUNAN GAME EDUKASI KESENIAN SUNDA MENGGUNAKAN UNITY ENGINE (Studi Kasus: "Petualangan Pangeran Jawa Barat"). Universitas Pasundan Bandung.
- BUDDY SEPTYANTO, R., SETYANINGSIH, E., & BACHARUDIN, F. 2017. Analisis Penempatan Evolved Node B Area DKI Jakarta Dengan Menggunakan Algoritma Genetika Dan Evolutionary Programming. *TESLA*. 19(2), pp 108-123.
- EIBEN, A.E, SMITH, J.E. 2003. Introduction to Evolutionary Computing; Springer: Berlin/Heidelberg, Jerman.
- JANIKOW, C. Z., & MICHALEWICZ, Z. 1991. An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms. *Proceedings of the Fourth International Conference on Genetic Algorithms*: 31–36.
- MANNING, J., & BUTTFIELD-ADDISON, P. 2017. Game Development with Unity. California: O'Reilly Media.
- NGUYEN, TU T. "Tu Thanh Nguyen INTRODUCTION TO 2D GAME DEVELOPMENT WITH UNITY". 2021. South-Eastern Finland University of Applied Sciences.
- WANG, Y.M. & YIN, H.L. 2018. Cost-Optimization Problem with a Soft Time Window Based on an Improved Fuzzy Genetic Algorithm for Fresh Food Distribution. *Math. Probl. Eng.* 2018, pp. 1–16.
- WICAKSONO, S. A. 2018. Optimasi Sistem Penempatan Magang Menerapkan Algoritme Genetika. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*. 6(1) pp. 17-24.
- WHITLEY, DARREL. 1994. A Genetic Algorithm Tutorial. *Statistics and Computing*. 4 (2): 65-85.
- YUNUS, M., & THOBIAS RUMLAHLAK, R. 2018. Optimasi Waktu Eksekusi Penentuan Rute Menuju Obyek Wisata Di Malang Raya Dengan Algoritma Genetika. *Jurnal Teknologi Informasi*. 9(1) pp. 29-40.
- NUGROHO, A. and PERMADI, I., 2020. Implementasi Jalur Pendek Menggunakan Ant Colony Optimization. *Dinamika Rekayasa*, 16(1).