

Implementasi Algoritma Genetika dan Aplikasi Model Pelatihan Multilayer Perceptron

Rifqi Fauzi Rahmadzani
Departemen Teknik Elektro dan Teknologi Informasi
Universitas Gadjah Mada
Yogyakarta, Indonesia
email: rifqifauzi@mail.ugm.ac.id

Abstrak—Seiring berjalannya waktu, hingga saat ini berbagai pendekatan kecerdasan dan pemecahan optimasi masalah menjadi sebuah keharusan pada berbagai bidang. Perkembangan teknologi mesin semakin hari semakin efisien kinerjanya, banyak aplikasi dibuat untuk menangani suatu masalah yang kompleks. Algoritma genetika adalah teknik pencarian heuristik dalam kecerdasan buatan untuk menemukan solusi paling optimal pada suatu masalah yang diberikan berdasarkan *crossover*, *mutation*, *selection*, dan beberapa teknik lainnya yang terinspirasi oleh teori evolusi Darwin. Pada artikel ini menunjukkan bagaimana pendekatan algoritma genetika menangani permasalahan optimasi secara umum. Implementasi algoritma ini membangun sebuah AI dalam permainan *arcade* yang cukup terkenal yaitu Flappy Bird.

Kata kunci—algoritma genetika, multilayer perceptron, game, kecerdasan buatan

I. PENDAHULUAN

Implementasi algoritma genetika bisa dikatakan berhasil dalam penanganan suatu aplikasi yang kompleks. Hal ini karena melibatkan banyak objektif dan fungsi optimasi yang tidak terdefinisi dengan baik. Misalnya, desain sirkuit analog [1] dan optimalisasi antena ruang-lahir [2] yang dikembangkan menggunakan Algoritma genetika.

Algoritma optimisasi sederhana dapat dibagi menjadi tiga kelas: algoritma pemrograman matematika, heuristik, dan metaheuristik. Algoritma pemrograman matematika memiliki fondasi yang paling ketat, dan dimungkinkan untuk membuktikan bahwa algoritma tersebut benar-benar konvergen, algoritma ini memeriksa apakah solusi yang diusulkan mendekati setidaknya optimum lokal, dan untuk memperkirakan tingkat konvergensi. Algoritma metaheuristik gunakan metafora, biasanya berasal dari bidang yang tidak terkait untuk didefinisikan sebagai algoritma pemilihan. *Simulated annealing*, *ant colony*, *bee swarm*, *harmony search* adalah contoh dari algoritma yang berasal dari berbagai bidang ilmu komputer.

Algoritma genetika adalah teknik pencarian metaheuristik kecerdasan buatan yang berasal dari teori evolusi biologis organisme oleh Darwin. Adapun banyak metode metaheuristik lainnya, banyak penelitian membahas tentang algoritma genetika dikarenakan kurangnya fondasi matematika secara inti dan konkret.

II. ANALISA DAN KEBUTUHAN

A. Analisa Kebutuhan

Menganalisa kebutuhan apa saja yang diperlukan dalam membangun sebuah AI menggunakan algoritma genetika pada permainan Flappy Bird diantaranya.

1. Alat dan Bahan

Alat dan bahan yang digunakan untuk implementasi algoritma genetika adalah sebagai berikut.

Perangkat keras	Perangkat lunak
a. Processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHZ b. Hard Disk 1 TB, SSD 250 GB c. RAM 8 GB d. GPU NVIDIA Geforce 930MX	a. Sistem operasi Windows 10 Pro 64-bit b. Python 3.7 c. Pycharm CE 2018.3.5 d. Librari Matplotlib e. Librari Numpy f. Librari Pygame g. Librari Random h. Librari Sys i. Librari Scipy

III. SPESIFIKASI RANCANGAN SOLUSI

A. Metodologi Algoritma genetika

1. Masalah Optimasi

Banyak masalah dalam kehidupan nyata tidak memiliki rumus dan teknik untuk menghitung hasil secara tepat karena jumlah generik yang sangat banyak. Algoritma genetika berjalan pada sebuah populasi solusi yang pasti dan mengembangkannya menggunakan metode yang terinspirasi oleh teori Darwin dalam biologi. Artikel ini akan membahas perbedaan antara algoritma genetika dengan metode lain dan juga melakukan beberapa percobaan untuk memperkirakan efektivitas algoritma genetika.

Setiap permasalahan pada algoritma genetika membutuhkan fungsi kebugaran yang mengukur kualitas atas solusi menuju optimasi masalah.

2. Terminologi

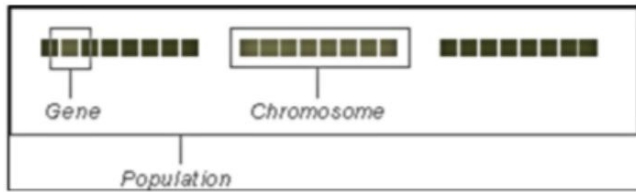
Pada algoritma genetika, populasi dari solusi kandidat (*phenotypes*) dikembangkan ke arah solusi yang lebih baik dari masalah optimisasi. Setiap solusi kandidat diwakili oleh kromosom yang merupakan sekumpulan gen yang dapat mengubah mutasi dan *crossover*. Satu set kromosom baru, juga dikenal sebagai generasi yang dibentuk menggunakan tanggul seleksi [3].

2.1. Populasi, kromosom, dan genom:

Populasi: Jumlah individu hadir dengan kromosom yang sama panjang. Dengan kata lain, mereka adalah satu set solusi pasti untuk masalah optimisasi.

Genom: Bagian dari kromosom. Nilai masing-masing gen berpengaruh pada kualitas dari sebuah solusi.

Kromosom: Satu set genom. Kromosom merupakan solusinya dalam bentuk gen.



Gambar 1: Populasi, Kromosom, dan Gen

2.2. Selection, mutation, dan crossover:

Selection: Populasi generasi berikutnya akan mempertahankan sebagian solusi terbaik pada generasi sebelumnya sehingga sifat-sifat terbaiknya akan dipertahankan.

Mutation: Mengubah genom dalam kromosom.

Crossover: Mencampur dua individu untuk menghasilkan pasangan baru keturunan yang memiliki sifat keduanya

B. Operator Genetik

1. Encoding dan Decoding

Teknik *encoding* sangat tergantung pada masalah. Secara umum, *encoding* adalah urutan setiap gen yang memiliki efek pada solusi. *Decoding* berfungsi menerjemahkan kromosom ke solusi masalah optimisasi. Encoding merupakan metode untuk membersihkan data sebelum memasukkannya ke dalam operator genetik.

Metode Encoding	Kromosom
Biner	1011000010010
Nilai	1 2 6 7 1 8 0 1 5
Nilai	ABDOMNABCDAC
Nilai	(back), (back), (right), (forward)

2. Selection

Proses seleksi utamanya bertanggung jawab untuk memastikan kelangsungan hidup individu yang paling cocok. Solusi terbaik akan dipertahankan di generasi berikutnya.

3. Metode pemilihan roda roulette

Pemilihan proporsional kebugaran, juga dikenal sebagai pemilihan roda roulette, adalah operator genetik yang digunakan dalam algoritma genetika dengan memilih solusi secara potensial untuk rekombinasi.

Dalam metode ini, setiap gen memiliki kemungkinan untuk dipilih sebagai generasi selanjutnya. Probabilitas dapat didefinisikan pada persamaan 1.

$$P_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (1)$$

P_i : Probabilitas individu dengan indeks i yang dipilih.

f : Fungsi kebugaran

f_i : Nilai kebugaran individu dengan indeks i

4. Metode pemilihan kebugaran terbaik

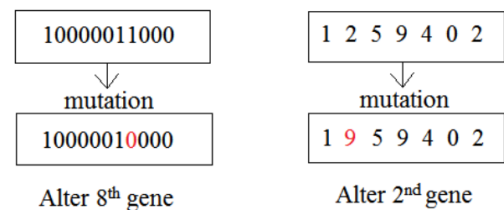
Dalam metode seleksi ini, individu-individu terbaik dengan kebugaran tertinggi dipilih. Metode ini merupakan pertukaran antara keanekaragaman populasi dan rata-rata kebugaran populasi dimana keragaman menurun dan

kebugaran rata-rata meningkat dibandingkan dengan metode pemilihan roda roulette.

5. Mutation

Mutation digunakan untuk mempertahankan keragaman genetik dari satu generasi populasi kromosom ke generasi berikutnya. Hal ini merupakan analogi untuk mutasi biologis.

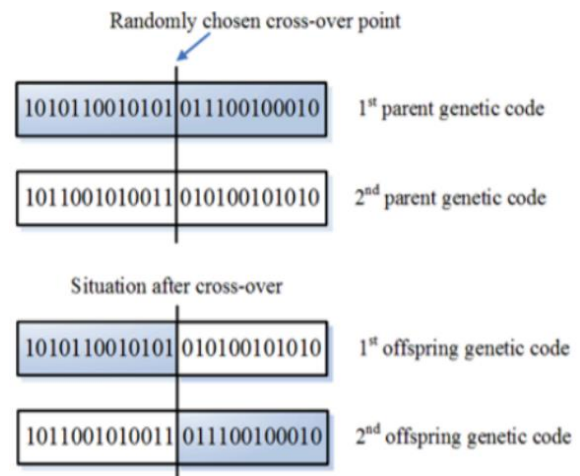
Tujuan mutasi dalam GA adalah melestarikan dan memperkenalkan keragaman. Mutasi harus memungkinkan algoritma dapat keluar dari minimum lokal dengan mencegah populasi kromosom menjadi begitu mirip satu sama lain, sehingga dapat memperlambat atau bahkan menghentikan evolusi. Mutasi dapat dilakukan dengan rumus atau secara acak.



Gambar 2: Contoh mutasi pada *binary encoding* dan *value encoding*.

6. Crossover

Crossover bertujuan memisahkan individu induk dan menggabungkannya kembali. Titik *crossover* dapat dipilih secara acak untuk meningkatkan keragaman populasi baru.



Gambar 3: Kode genetik orang tua dan anak sebelum dan sesudah crossover [4].

Multi-point crossover merupakan *crossover* sederhana dengan lebih dari satu posisi di mana proses *crossover* akan terjadi.

IV. IMPLEMENTASI

A. Gambaran Umum

Algoritma dapat dijalankan secara terus-menerus membuat serangkaian solusi baru menggunakan algoritma genetika untuk mengembangkan setiap kali dalam sebuah generasi hingga ditemukan solusi yang dapat diterima.

B. Algoritma Pseudo

MULAI

Inisialisasi populasi.

Hitung kebugaran masing-masing individu.

LAKUKAN SAMPAI SOLUSI TERBAIK

Pengkodean setiap individu untuk menghasilkan kromosom.

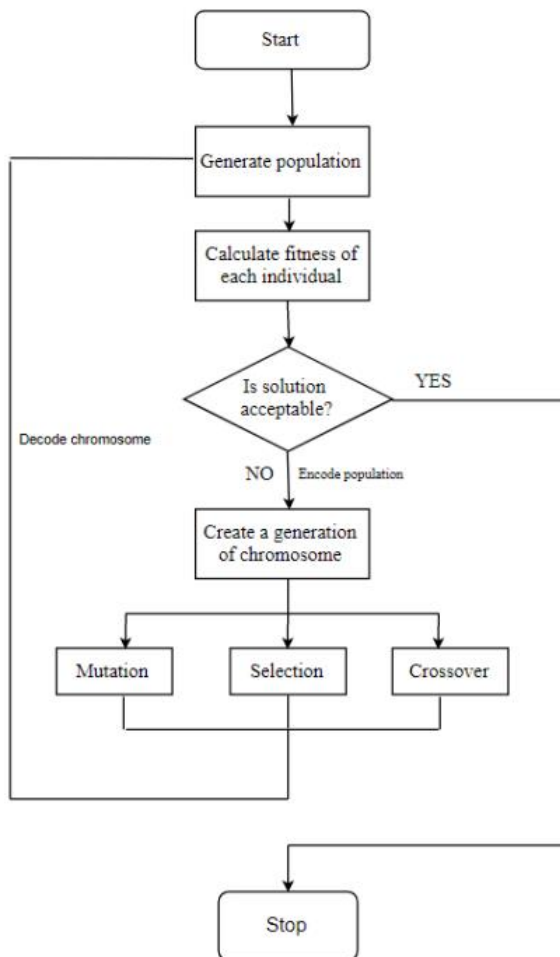
Lakukan operator algoritma genetika pada populasi yang ada.

Dekode populasi baru dan bunuh populasi lama.

ULANGI

SELESAI

C. Flowchart



Gambar 4: Flowchart algoritma genetika

D. Aplikasi dalam Membangun AI untuk Permainan Flappy Bird

1. Gambaran Umum

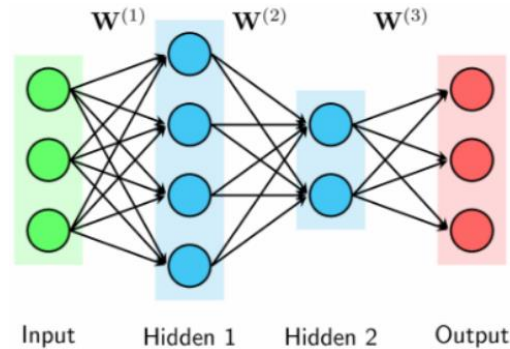
Flappy Bird adalah game ponsel yang dikembangkan oleh developer Vietnam bernama Dong Nguyen. Tujuannya adalah untuk mengarahkan burung terbang, bernama “Faby”, yang bergerak terus menerus ke kanan, di antara set pipa seperti Mario. Jika pemain menyentuh pipa, maka *gameover*. Faby dengan cepat mengepak ke atas setiap kali pemain mengetuk layar; jika layar tidak diketuk, Faby jatuh karena gravitasi; setiap pasang pipa yang dilewati menghasilkan satu poin bagi pemain.

2. Model Pelatihan

Pada bagian ini, model multilayer perceptron akan dibahas dan diimplementasikan pada game Flappy Bird.

2.1. Jaringan Saraf Tiruan

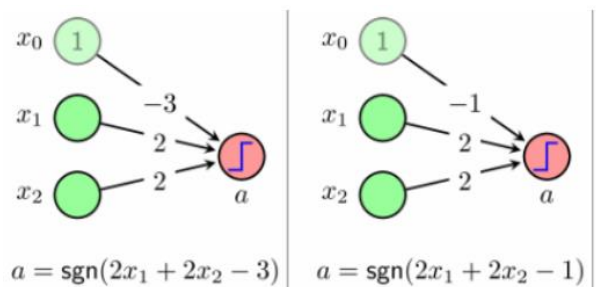
2.1.1. *Layer*: satu set neuron (sebuah lingkaran yang mengandung angka). Selain itu layer input dan output, satu Multilayer Perceptron dapat memiliki satu atau lebih layer tersembunyi.



Gambar 5: Contoh jaringan perceptron multilayer dengan 2 lapisan tersembunyi

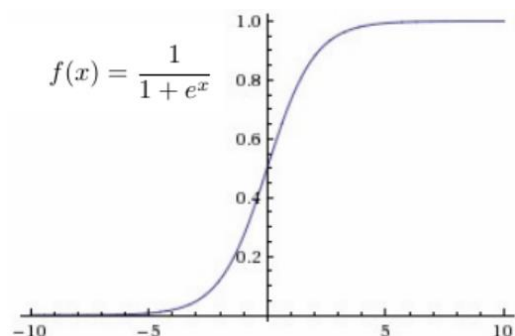
2.1.2. *Unit*: Satu simpul (lingkaran pada Gambar 6) disebut satu unit. Input dari setiap unit dilambangkan sebagai z dan output dari setiap unit dilambangkan sebagai a (a singkatan dari aktivasi, unit input di lapisan berikutnya)

2.1.3 *Bobot dan Bias*: Pada gambar 7, angka yang menghubungkan 2 simpul dalam 2 lapisan disebut Bobot, angka tersebut menentukan seberapa besar pengaruh suatu simpul pada unit input berikutnya di lapisan berikutnya, bias adalah simpul x_0 pada gambar 7, nilainya biasanya konstan yaitu 1. Bias menambahkan fleksibilitas ke jaringan.

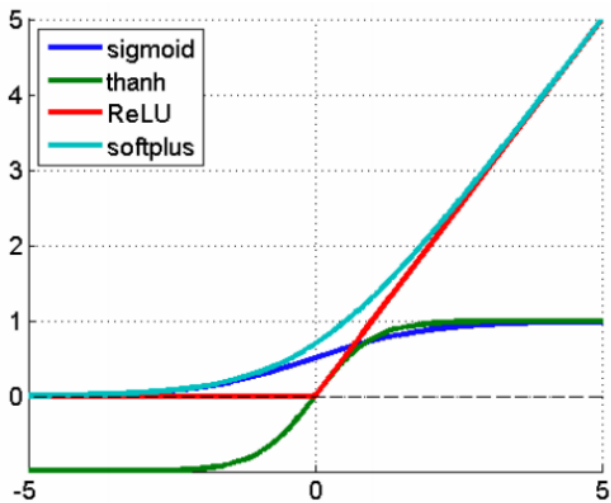


Gambar 6: Proses *feed forward* dengan aktivator Sigmoid

2.1.4. *Aktivator, fungsi aktivasi*: Ketika berbicara tentang aktivator, berarti fungsi yang berlaku pada node untuk menghasilkan unit output. Tujuan dari fungsi aktivasi adalah untuk menekan nilai setelah mengalikan nilai node dan bobot dalam menghasilkan angka dengan rentang yang ditentukan.



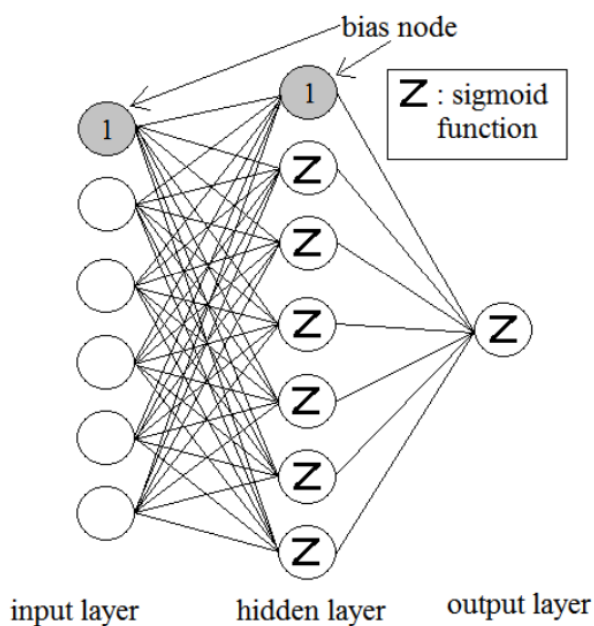
Gambar 7: Sigmoid menekan angka ke nilai 0 ketika x menuju tak terbatas dan 1 ketika x menuju tak terbatas



Gambar 8: Berbagai fungsi aktivasi

2.1. Jaringan Saraf Tiruan untuk AI permainan Flappy Bird

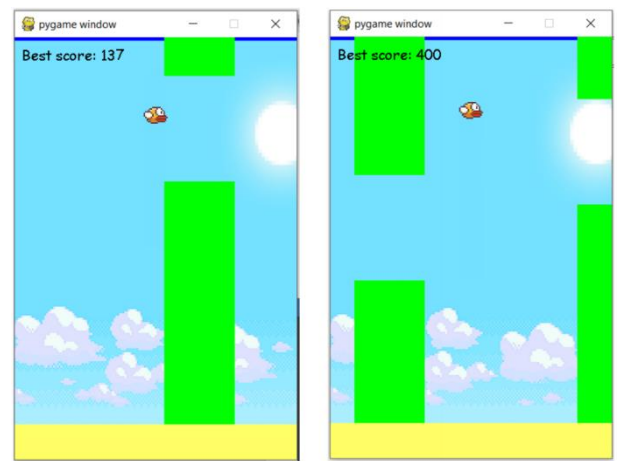
Dalam percobaan ini, model sederhana *multilayer perceptron* [5] dipilih. Jaringan ini mencakup satu lapisan tersembunyi dengan 6 simpul, satu lapisan keluaran dan satu lapisan masukan. Satu simpul bias ditambahkan pada lapisan input dan lapisan tersembunyi. Node bias ini memastikan variabel konstan dapat memiliki efek pada solusi. Lapisan tersembunyi dan lapisan keluaran menggunakan fungsi sigmoid sebagai aktivator. Output dari jaringan saraf adalah angka dalam kisaran 0 dan 1. Ambang batas diatur ke 0,5. Jika output $> 0,5$ maka burung akan mengepak.



Gambar 9 Arsitektur jaringan saraf tiruan

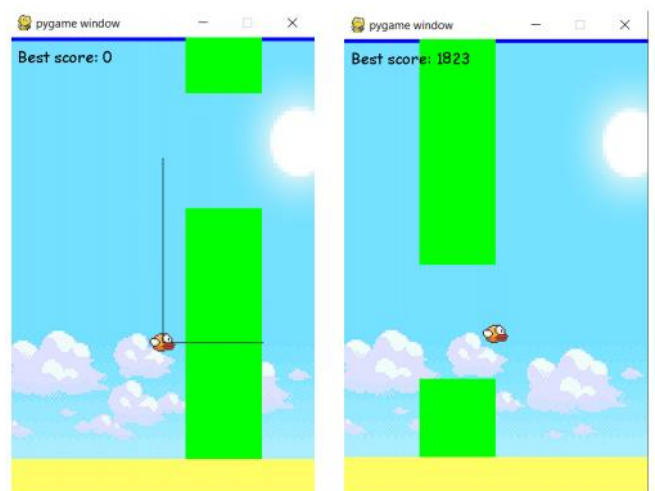
3. Masukan untuk model pelatihan

Dalam jaringan saraf, setiap input harus terkait dengan solusi. Ada 5 input seperti yang dijelaskan pada gambar 10.



(a) Generasi = 300, (b) Generasi = 350, Best score - 137 Best score - 400

Gambar 10: Hasil percobaan 1



(a) Dua sensor burung: garis horizontal dan garis vertikal (b) Skor kinerja terbaik 1823

Gambar 11: Hasil percobaan 2

Input 1: Jarak horizontal antara burung dan tabung (garis horizontal pada gambar 12).

Input 2: Jarak vertikal antara burung dan bagian tengah dari dua tabung (Garis vertikal pada gambar 12).

Input 3: Lebar burung.

Input 4: Tinggi burung.

Input 5: Lebar tabung.

4. Implementasi Algoritma genetika

4.1. *Encode*: Ada 49 bobot dalam jaringan saraf pada gambar 10. Semua bobot ini ditambahkan ke dalam *float array* ukuran 49. Array ini akan membawa semua informasi dari seluruh jaringan. Maka, setiap larik dianggap sebagai kromosom dengan metode pengkodean nilai (dijelaskan pada bagian III-1). Setiap bobot akan menjadi gen dalam kromosom.

4.2. Penerapan operator genetika:

Selection: Metode pemilihan kebugaran terbaik (III-2.2)

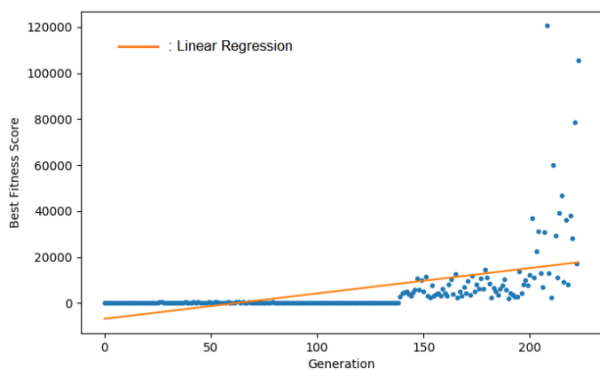
Mutation: Mengubah bobot secara acak dengan memberikan penomoran baru. (III-4)

Crossover: Pertukaran dua bobot dengan posibilitas tetap (III-3)

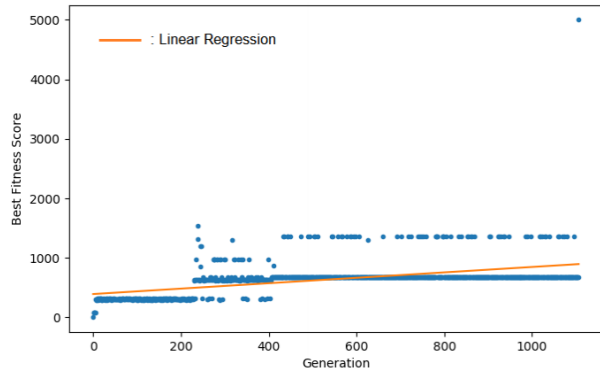
Paramater	Nilai
Populasi	50
Tingkat Seleksi	0.1
Tingkat Mutasi	0.7
Tingkat Crossover	0.6
Presentase Mutasi	0.6
Presentase Crossover	0.1

4.3. *Fitness Function*: Dalam percobaan ini, fungsi kebugaran hanyalah waktu bertahan hidup seekor burung, satuannya adalah jumlah *refresh frame* setelah burung mati.

V. EVALUASI



Gambar 12: Hasil percobaan 1



Gambar 13: Hasil percobaan 2

Dua grafik di atas menggambarkan korelasi antara jumlah generasi dan skor kebugaran terbaik untuk mengetahui apakah algoritma genetika benar-benar membantu populasi berkembang dari waktu ke waktu atau tidak. Dalam kedua percobaan, solusi terbaik dapat ditemukan. Namun, percobaan pertama memakan waktu 230 generasi sedangkan percobaan kedua mengambil lebih dari 1000 generasi untuk menemukan solusi terbaik. Koefisien korelasi (*Pearson*) dilakukan untuk menguji apakah ada korelasi antara jumlah generasi dan skor kebugaran terbaik.

Percobaan	pearson-r	p-value
Percobaan 1	0.5	0.001
Percobaan 2	0.5	0.001

Kedua percobaan menghasilkan hasil yang sama, terdapat hubungan linear yang kuat dan positif antara jumlah generasi, skor kebugaran terbaik, dan korelasi Pearson menunjukkan bahwa hubungan ini signifikan, $r = .5, p < .001$.

Dapat disimpulkan bahwa algoritma genetika mampu meningkatkan optimasi secara berkala. Namun, kompleksitas waktu tidak dapat diukur dengan tepat karena tingginya aspek arbitrer.

VI. KESIMPULAN

Dapat diketahui bahwa tidak banyak dasar ilmu matematika murni untuk algoritma genetika. Algoritma tersebut secara murni berasal dari biologi. Kinerja algoritma genetika juga tidak dapat dihitung atau diperkirakan, hal ini menjadi alasan mengapa banyak orang cenderung memiliki pendapat negatif tentang algoritma ini. Namun, penulis sangat percaya bahwa suatu hari, akan ada peneliti menemukan dasar yang kuat dan membuat peningkatan dari algoritma genetika. Sebagai contohnya adalah neural network, ide tersebut berasal dari bidang yang tidak berhubungan, yaitu *neuroscience*, node-node diilustrasikan sebagai hubungan antara jutaan neuron di otak manusia. Sering kali, peneliti dan pengembang meningkatkan kinerja jaringan saraf, beberapa teknik pelatihan seperti backpropagation diperkenalkan dan memastikan konvergensi masalah optimasi, banyak juga variasi telah dibuat untuk jaringan saraf, seperti convolutional neural network (CNN), recurrent neural network (RNN) dan long short-term memory (LSTM). Sekali lagi, penulis sangat yakin bahwa akan ada pertumbuhan skala penelitian yang membahas pada algoritma genetika suatu hari nanti.

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat-Nya sehingga penulis dapat menyelesaikan artikel ini. Penulis juga mengucapkan teima kasih kepada Bapak Noor Akhmad Setiawan yang telah banyak memberikan bimbingan perkuliahan selama ini. Tidak lupa ucapan terima kasih kepada Tuan Dung Lai yang telah memberikan inspirasi atas terselesaikannya penulisan artikel ini.

DAFTAR PUSTAKA

- [1] A. Jafari, M. Zekri, S. Sadri, and A. R. Mallahzadeh, "Design of Analog Integrated Circuits by Using Genetic Algorithm," pp. 2–5, 2010.
- [2] H. Tao, G. Liao, and L. Wang, "Space-Borne Antenna Adaptive Side-Lobe Nulling Algorithm Based on Gradient-Genetic Algorithm," pp. 733–736, 2004.
- [3] E. Eiben, "Genetic algorithms with multi-parent recombination," pp. 78–87, 1994.
- [4] E. Schultz, J. Mellander, C. Endorf, "Intrusion Detection and Prevention - A basic genetic algorithm," 2008.
- [5] N. Andrew, "Machine learning," Stanford University Online, lecture notes week 4, [online]: <https://www.coursera.org/learn/machine-learning>