

Optimasi Pemilihan Komponen Terbaik untuk Perakitan *Skateboard* dengan Algoritma Genetika Berbasis Java pada Toko DSRuntul

Reni Utami^{1*}

Universitas Dian Nusantara Teknik dan Informatika, Kampus 1 Tanjung Duren
E-Mail: reni.utami@dosen.undira.ac.id

Irfan Nurdiansyah²

Universitas Dian Nusantara Teknik dan Informatika, Kampus 1 Tanjung Duren
E-Mail: irfan.nurdiansyah@dosen.undira.ac

ABSTRACT

Along with the development of skateboarding and advances in information technology, consumers are now smarter in purchasing. Consumers prefer assembled skateboards because consumers can get skateboard components according to their wishes. The problem faced by the Dsruntul Store is that it still sells complete skateboards. This can cause consumers to be unable to choose the desired skateboard components. As for the assembly according to consumer demand, the assembly still uses the conventional method of selecting each skateboard component one by one. This problem can be overcome by the existence of an application that is able to search for the selection of skateboard component combinations from each type of skateboard component according to the consumer's budget and the desired specific component criteria effectively and efficiently. In this application, the Genetic Algorithm method will be applied, which is an optimization method that can provide alternative solutions to a problem that is adapted to the genetic process of biological organisms based on Charles Darwin's theory of evolution. The coding technique used in determining skateboard components is integer representation, the selection used is roulette wheel selection, the crossover used is one point crossover and the mutation used is random mutation.

Keywords: Genetic Algorithm, Roulette Wheel Selection, One Point Crossover, Random Mutation, Skateboard Components

PENDAHULUAN

Seiring dengan perkembangan olahraga *skateboard* dan kemajuan teknologi informasi, kini para konsumen lebih cerdas dalam membeli. Konsumen lebih menginginkan *skateboard* rakitan, sebab dari segi kualitas dan kuantitas konsumen merasa dipuaskan. Selain itu konsumen juga bisa mendapatkan komponen *skateboard* sesuai dengan keinginannya. Permasalahan yang dihadapi Toko Dsruntul yaitu belum adanya aplikasi yang mendukung proses perakitan *skateboard* tersebut. Pihak toko masih melakukan transaksi penjualan *skateboard* utuh, dimana toko merakit *skateboard* sesuai dengan keinginan toko. Hal ini dapat menyebabkan konsumen tidak dapat memilih komponen *skateboard* yang diinginkan. Adapun perakitan yang sesuai dengan permintaan konsumen, namun perakitan tersebut masih menggunakan cara konvensional yaitu memilih satu-satu dari tiap komponen *skateboard*. Permasalahan tersebut mempengaruhi penulis untuk membangun suatu aplikasi yang mampu melakukan pencarian untuk pemilihan kombinasi komponen *skateboard* terbaik dari tiap jenis komponen *skateboard* sesuai dengan anggaran yang dimiliki konsumen dan kriteria komponen tertentu yang diinginkan secara efektif dan efisien menggunakan Algoritma Genetika. Diharapkan dengan adanya aplikasi ini nantinya dapat meningkatkan produktifitas, kualitas dan kesesuaian harga serta memudahkan konsumen

dalam merakit *skateboard* sesuai yang diinginkan secara efektif dan efisien.

Algoritma Genetika merupakan algoritma yang digunakan dalam teknik pencarian *evolutionary computation* yang menemukan perkiraan solusi untuk optimasi dan masalah pencarian. Algoritma Genetika termasuk algoritma *heuristik* (solusi yang mendekati optimum dengan waktu yang relatif cepat) dan *stochastic* (menerapkan metode pengacakan dalam mencari himpunan solusi) (Mahmudy, 2013).

Berdasarkan latar belakang masalah di atas, maka dapat dirumuskan masalah sebagai berikut:

- Bagaimana cara mendapatkan kombinasi komponen *skateboard* menggunakan Algoritma Genetika?
- Bagaimana membangun aplikasi yang dapat melakukan pencarian kombinasi komponen *skateboard* sesuai dengan anggaran yang dimiliki *user* dan kriteria komponen tertentu yang diinginkan secara efektif dan efisien?

BAHAN DAN METODE

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan penerapan Algoritma Genetika (Genetic Algorithm/GA) untuk mengoptimalkan pemilihan kombinasi komponen terbaik dalam perakitan *skateboard*. Penelitian dilaksanakan secara terapan (applied research) dengan studi

kasus di Toko DSRUNTUL, yang menjadi mitra penelitian.

Tujuan utama metode ini adalah menghasilkan sistem berbasis Java yang dapat memberikan rekomendasi komponen skateboard optimal berdasarkan sejumlah kriteria seperti harga, durabilitas, berat, kompatibilitas, dan rating pelanggan.

Penelitian dibagi menjadi enam tahapan utama:

1. Pengumpulan dan praproses data,
2. Perancangan dan implementasi algoritma,
3. Validasi dan eksperimen,
4. Integrasi dan uji lapangan,
5. Optimasi dan evaluasi,
6. Dokumentasi dan pelaporan akhir.

Tahapan dan Proses Penelitian

Pengumpulan dan Praproses Data

Proses:

- a. Mengumpulkan data komponen skateboard (deck, truck, wheel, bearing, hardware) dari inventori Toko DSRUNTUL.
- b. Setiap komponen dicatat berdasarkan atribut: *harga, berat, durabilitas, rating pengguna, kompatibilitas, dan stok*.
- c. Data kemudian dinormalisasi agar dapat diolah secara matematis oleh algoritma genetika.
- d. Dilakukan wawancara dengan pemilik dan teknisi toko untuk menentukan bobot awal kriteria.

Luaran:

Dataset terstruktur (format CSV/SQL) dan tabel bobot kriteria.

Indikator capaian:

Tersedianya data lengkap minimal 50 variasi komponen dari 5 kategori utama dengan validitas >90%.

Penanggung jawab:

Mahasiswa peneliti (developer), didampingi dosen pembimbing dan mitra Toko DSRUNTUL.

Perancangan dan Implementasi Algoritma Genetika

Proses:

- a. Mendesain encoding kromosom, di mana setiap kromosom mewakili satu kombinasi komponen skateboard lengkap.
- b. Menentukan fungsi fitness berdasarkan kombinasi bobot kriteria:
- c.
$$F(x) = \sum_{i=1}^n w_i \times f_i(x)$$
$$F(x) = \sum_{i=1}^n w_i \times f_i(x)$$

- d. di mana $f_i(x)$ adalah skor setiap kriteria dan w_i bobotnya.
- e. Mengimplementasikan proses seleksi, crossover, mutasi, dan elitism menggunakan bahasa pemrograman Java.
- f. Melakukan tuning parameter seperti ukuran populasi, jumlah generasi, dan probabilitas crossover/mutasi.

Luaran:

Modul sistem rekomendasi berbasis Java dengan implementasi Algoritma Genetika yang berfungsi penuh.

Indikator capaian:

Program berhasil dieksekusi dan menghasilkan solusi dengan *fitness value* meningkat secara konsisten selama iterasi.

Penanggung jawab:

Mahasiswa pengembang (coding), dengan supervisi dosen pembimbing di bidang komputasi evolusioner.

Validasi dan Eksperimen

Proses:

- a. Melakukan pengujian sistem menggunakan data uji (test set).
- b. Membandingkan hasil rekomendasi sistem dengan metode manual (dipilih oleh ahli perakitan toko).
- c. Mengukur performa algoritma berdasarkan akurasi solusi, stabilitas konvergensi, dan waktu komputasi.
- d. Melakukan analisis statistik sederhana terhadap hasil eksperimen.

Luaran:

Laporan hasil uji sistem dan analisis performa algoritma.

Indikator capaian:

- a. Nilai fitness sistem meningkat minimal 15% dibanding baseline manual.
- b. Waktu komputasi per solusi <10 detik.

Penanggung jawab:

Mahasiswa peneliti dan dosen pembimbing; validasi dilakukan bersama mitra toko.

Integrasi dan Uji Lapangan

Proses:

- a. Mengintegrasikan modul rekomendasi ke dalam sistem sederhana berbasis Java GUI.
- b. Melakukan uji coba di lingkungan nyata (Toko DSRUNTUL) untuk menilai efisiensi dan kepuasan pengguna.
- c. Mengumpulkan umpan balik kualitatif dari staf toko dan pelanggan yang mencoba rekomendasi sistem.

Luaran:

Sistem prototipe siap digunakan dan laporan hasil uji lapangan.

Indikator capaian:

- a. Sistem dapat dioperasikan oleh staf toko tanpa error selama 3 hari uji coba.
- b. Tingkat kepuasan pengguna $\geq 80\%$ (berdasarkan survei sederhana).

Penanggung jawab:

Mahasiswa peneliti (operator), mitra toko (penguji lapangan).

Optimasi Parameter dan Evaluasi Akhir

Proses:

- a. Melakukan penyetelan ulang parameter algoritma berdasarkan hasil uji lapangan.
- b. Menganalisis sensitivitas terhadap perubahan bobot kriteria (harga, performa, estetika).
- c. Melakukan perbandingan antar konfigurasi untuk menemukan parameter terbaik.

Luaran:

Versi final sistem dengan parameter optimal dan laporan evaluasi.

Indikator capaian:

Algoritma mencapai hasil stabil dengan deviasi fitness $< 5\%$ antar uji.

HASIL DAN PEMBAHASAN

Definisi Optimasi

Optimasi adalah suatu pendekatan normatif untuk mengidentifikasi penyelesaian terbaik dalam pengambilan keputusan dari suatu permasalahan. Penyelesaian permasalahan dalam teknik optimasi diarahkan untuk mendapatkan titik maksimum dan titik minimum dari fungsi yang dioptimalkan. Tujuan dari optimasi adalah untuk meminimumkan usaha yang diperlukan atau biaya operasional dan memaksimalkan hasil yang diinginkan. Jika usaha yang diperlukan atau hasil yang diharapkan dapat dinyatakan sebagai fungsi dari sebuah keputusan, maka optimasi dapat didefinisikan sebagai proses pencapaian kondisi maksimum dan minimum dari fungsi tersebut (Witarty dkk., 2009).

Algoritma Genetika

Algoritma Genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma Genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (Kusumadewi, 2003). Tujuan utama dari Algoritma ini adalah untuk menghapus dan untuk menyaring individu yang berkualitas rendah, untuk mendapatkan nilai individu yang optimal, untuk melakukan analisis statistik untuk memverifikasi relevansi biologis individu dan untuk memilih individu yang memiliki tingkat reproduksi dan tingkat survival yang lebih tinggi dibandingkan dengan individu yang kurang *fit* (kuat). Pada suatu waktu tertentu (atau lebih sering dikenal dengan istilah generasi), populasi

secara keseluruhan akan lebih banyak memuat organisme yang *fit*.

Sifat algoritma genetika adalah mencari kemungkinan – kemungkinan dari calon solusi untuk mendapatkan yang optimal bagi penyelesaian masalah. Ruang cakupan dari semua solusi yang layak (*Feasible*), yaitu obyek – obyek di antara solusi yang sesuai, dinamakan ruang pencarian (Search Space). Tiap titik dalam ruang pencarian merepresentasikan satu solusi yang layak. Tiap solusi yang layak dapat ditandai dengan nilai *Fitness*-nya bagi masalah. Solusi yang dicari dalam algoritma genetika adalah titik (satu atau lebih) diantara solusi yang layak dalam ruang pencarian. Sifat pencarian inilah yang menyebabkan algoritma genetika baik untuk di terapkan dalam menyelesaikan masalah optimasi (Desiana dan Arhami, 2005).

Algoritma Genetika diilhami oleh ilmu genetika, oleh karena itu istilah yang digunakan dalam Algoritma Genetika banyak diadaptasi dari ilmu tersebut. Pada Algoritma Genetika teknik pencarian dilakukan sekaligus atas sejumlah solusi yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom, kromosom inilah yang merepresentasikan sebuah solusi permasalahan. Kromosom terdiri atas gen yang tersusun secara linier dan alel sebagai nilai yang terdapat dalam gen (Kusumadewi, 2003).

Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut fungsi *fitness*. Fungsi *fitness* dapat berupa fungsi matematika atau fungsi lainnya dengan melihat permasalahan yang hendak diselesaikan. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dari populasi tersebut. Kromosom yang berkualitas baik mempunyai kemungkinan yang lebih besar untuk terpilih menjadi induk (*parent*). Generasi berikutnya dikenal dengan istilah anak (*offspring*) yang terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk dengan menggunakan operator penyilangan (*crossover*). Selain penyilangan, suatu kromosom dapat juga dimodifikasi dengan cara mutasi. Kondisi berhenti dari Algoritma Genetika apabila solusi yang diberikan telah konvergen atau jumlah generasi yang diminta telah tercapai (Mawaddah, 2015).

Perbedaan antara Algoritma Genetika dengan teknik optimasi yang biasa digunakan yakni (Desiana dan Arhami, 2005):

- a. Algoritma Genetika bekerja dengan pengkodean dari himpunan solusi permasalahan berdasarkan parameter yang telah ditetapkan dan bukan parameter itu sendiri.
- b. Algoritma Genetika melakukan pencarian pada sebuah populasi dari sejumlah individu-

individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.

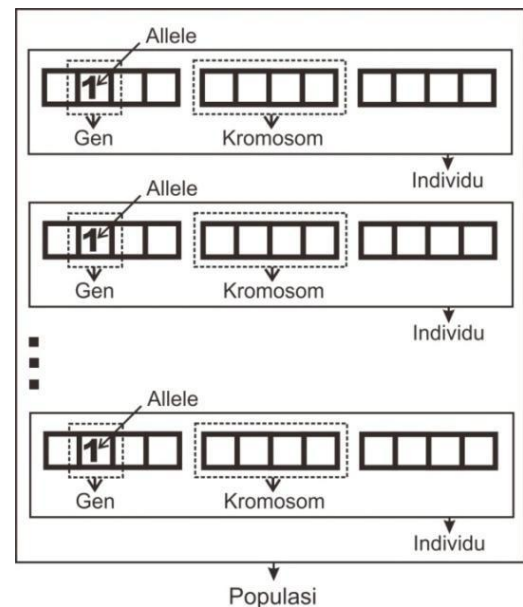
- c. Algoritma Genetika merupakan informasi fungsi objektif (*fitness*), sebagai cara untuk mengevaluasi individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.
- d. Algoritma Genetika menggunakan aturan-aturan transisi peluang, bukan *deterministic*.
- e. Algoritma Genetika sangat berguna dan efisien untuk masalah dengan karakteristik sebagai berikut (Firmansyah dkk., 2012):
 - 1) Ruang masalah sangat besar, kompleks, dan sulit dipahami.
 - 2) Kurang atau bahkan tidak ada pengetahuan yang memadai untuk merepresentasikan masalah ke dalam ruang pencarian yang lebih sempit.
 - 3) Tidak tersedianya analisis matematika yang memadai.

Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi. Solusi yang diharapkan tidak harus paling optimal, tetapi cukup “bagus” atau bisa diterima. Terdapat batasan waktu, misalnya dalam real time system atau sistem waktu nyata.

Definisi Dalam Algoritma Genetika

Dalam konsep biologi dikenal adanya istilah sel. Sel merupakan bagian kecil yang menyusun sebuah organisme. Secara umum organisme terdiri dari sel-sel penyusun. Sebuah sel tersusun dari beberapa gen. Gen merupakan sekumpulan DNA (*Deoxyribo Nuclead Acid*). Konsep biologi semacam ini yang di adaptasi dalam Algoritma Genetika. Dalam Algoritma Genetika, kromosom merupakan sebuah alternatif solusi dari sebuah permasalahan. Kromosom dapat mempresentasikan beberapa bentuk sesuai dengan jenis pengkodean yang digunakan. Beberapa definisi yang perlu diketahui dalam Algoritma Genetika (Supardi, 2014):

- a. *Genotype* (gen) adalah sebuah nilai yang menyatakan suatu dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Representasi di dalam Algoritma Genetika berupa nilai *biner*, *float*, *integer*, karakter, kombinatorial.
- b. *Allele*, merupakan nilai dari suatu gen.
- c. Kromosom, gabungan dari gen yang mempresentasikan nilai tertentu dan menyatakan solusi yang mungkin dari suatu permasalahan.
- d. Individu, menyatakan sebuah nilai atau sebuah solusi yang mungkin dari permasalahan yang ada.
- e. Populasi, merupakan sekumpulan dari beberapa individu yang akan mengalami proses evolusi.



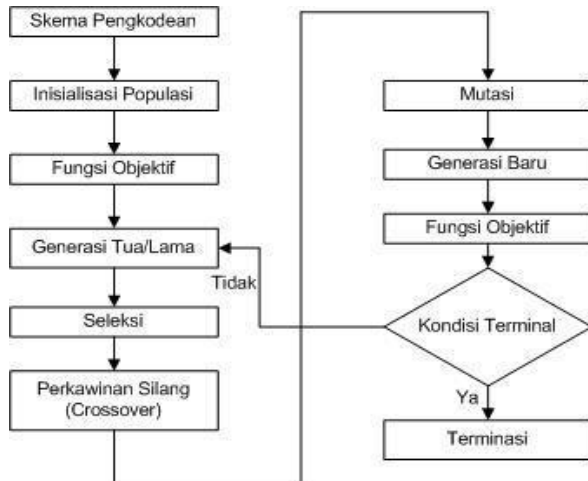
Gambar 1. Ilustrasi Representasi Dalam Algoritma Genetika

Sumber: (Supardi, 2014)

Proses Umum Algoritma Genetika

Algoritma Genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti yang terjadi pada makhluk hidup. Dengan menirukan proses genetika dan seleksi alami maka algoritma genetika akan menghasilkan kromosom setelah melewati sekian generasi. kromosom ini harus diuraikan (*decoding*) menjadi sebuah solusi yang diharapkan mendekati optimum.

Proses dalam algoritma genetika diawali dengan inisialisasi populasi awal dilakukan untuk menghasilkan solusi awal dari suatu permasalahan Algoritma Genetika. Inisialisasi ini dilakukan secara acak sebanyak jumlah kromosom/populasi yang diinginkan. Selanjutnya dihitung nilai *fitness* setelah itu dilakukan seleksi dengan menggunakan metode *roulette wheel*. Selanjutnya dilakukan perkawinan silang (*crossover*) dan mutasi. Setelah melewati sekian iterasi (generasi) akan didapatkan individu terbaik. Individu terbaik ini mempunyai susunan kromosom yang bisa dikonversi menjadi solusi yang terbaik (paling tidak mendekati optimum). Dari sini bisa disimpulkan bahwa algoritma genetika menghasilkan suatu solusi optimum dengan melakukan pencarian di antara sejumlah alternatif titik optimum berdasarkan fungsi probabilistic. Diagram alir Algoritma Genetika dapat dilihat pada gambar 2 berikut ini (Desiana dan Arhami, 2005):



Gambar 1.2 : Diagram alir Algoritma Genetika

- Skema Pengkodean**
 Pengkodean adalah bagaimana mengkodekan gen dari kromosom, gen merupakan bagian dari kromosom. Satu gen akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetik, maka alternatif solusi tersebut harus dikodekan terlebih dahulu kedalam bentuk kromosom.
- Inisialisasi Populasi Awal**
 Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal. Teknik dalam pembangkitan populasi awal ini ada beberapa cara, diantaranya adalah random generator, pendekatan tertentu, dan permutasi gen.
- Fungsi Objektif (Fitness)**
 Suatu individu atau kromosom dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performasinya. Fungsi objektif adalah formula yang dibuat untuk mengukur derajat kualitas individu. Formula ini yang akan dibuat untuk menghitung angka nilai *fitness*. Nilai *fitness* yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Algoritma Genetika bertujuan mencari individu dengan nilai *fitness* yang paling tinggi.
- Seleksi**
 Merupakan proses pemilihan individu sebagai orang tua yang biasanya dilakukan secara proporsional berdasarkan nilai-nilai *fitness*nya.
- Perkawinan Silang (crossover)**
 Mengkombinasikan dua kromosom orangtua berdasar nilai probabilitas *crossover*nya untuk menghasilkan *offspring*.

- Mutasi**
 Suatu proses berubahnya satu atau lebih nilai gen dalam kromosom dengan suatu nilai acak
- Update Generasi**
 Memperbarui pada Algoritma Genetika berjenis generational replacement, sejumlah individu pada suatu generasi digantikan sejumlah individu baru hasil pindah silang dan mutasi.
- Pengecekan Faktor Pemberhenti (Terminasi)**
 Iterasi/ generasi di Algoritma Genetika akan terus diulang dan berhenti jika memenuhi dari salah satu dari kondisi untuk berhenti. Proses iterasi bisa dihentikan berdasarkan beberapa kondisi, misalnya ketika:
 - Iterasi telah mencapai jumlah generasi maksimum yang diizinkan Iterasi berhenti sampai generasi n . Nilai n ditentukan sebelumnya berdasarkan beberapa eksperimen pendahuluan. Semakin tinggi ukuran dan kompleksitas masalah maka nilai n semakin besar. Nilai n ditentukan sedemikian rupa sehingga konvergensi populasi tercapai dan akan sulit didapatkan solusi yang lebih baik setelah n iterasi.
 - Terdapat suatu individu yang telah memiliki *fitness* tertentu yang diharapkan.
 - Keberagaman populasi telah mencapai tingkat minimum yang diizinkan.
 - Dalam beberapa generasi tertentu, tidak ada peningkatan nilai *fitness* yang diharapkan.

Skema Pengkodean

Pengkodean merupakan bagian yang terpenting dalam menyelesaikan suatu permasalahan menggunakan Algoritma Genetika. Pengkodean adalah pengkonversian masalah dalam dunia nyata menjadi bentuk yang dapat diolah dengan menggunakan Algoritma Genetika. Pengkodean yang tepat sangat menentukan keberhasilan menggunakan Algoritma Genetika dalam menyelesaikan masalah. Pengkodean yang tepat juga menentukan tingkat efisiensi komputasi yang digunakan.

Pada pengkodean *integer*, setiap gen bisa bernilai bilangan bulat (*integer*). Bilangan bulat bisa digunakan untuk merepresentasikan nomor urut, posisi, jenis, atau kualitas obyek. Dengan representasi ini, ukuran kromosom menjadi lebih sederhana (Suyanto, 2008).

Tabel 1. Representasi Kromosom dalam Pengkodean *Integer*

Parent 1	1	3	5	6	2	8	4	7
Parent 2	6	4	7	1	8	3	2	5

Fungsi Objektif (Fungsi *Fitness*)

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran nilai *fitness*nya. Dalam proses evolusi, individu yang survive dari

proses seleksi alam akan mempunyai kesempatan untuk bereproduksi kembali. Dengan demikian kemampuan individu untuk dapat beradaptasi dan bertahan hidup untuk selamat sangat penting. Dalam terminologi algoritma genetika kemampuan suatu individu atau kromosom untuk bertahan dapat diukur berdasarkan nilai *fitness*-nya. Semakin baik nilai *fitness* suatu kromosom maka akan semakin baik peluang kromosom tersebut untuk bertahan hidup dan ikut dalam proses reproduksi. Nilai *fitness* suatu kromosom dapat dihitung dengan menggunakan fungsi objektif (Desiana dan Arhami, 2005).

Seleksi

Seleksi merupakan proses menentukan individu mana saja yang akan dipilih untuk dilakukan rekombinasi. Individu yang terpilih atau terseleksi adalah individu dengan nilai *fitness* terbaik karena diharapkan dari individu yang baik akan didapatkan keturunan yang baik bahkan lebih baik dari individu awal (*parents*) sesuai dengan prinsip evolusi biologi. Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif (*Fitness*) dirinya sendiri terhadap nilai objektif (*Fitness*) dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya (Kusumadewi, 2003).

Pada *Roulette Wheel Selection*, kromosom akan terpilih secara acak ditentukan dengan memperhitungkan nilai *fitness* masing-masing kromosom. Semakin besar nilai *fitness* suatu kromosom, semakin besar pula peluang kromosom tersebut untuk terpilih sebagai orangtua. Metode seleksi *Roulette Wheel* dapat dianalogikan seperti permainan roda putar. Pada permainan roda putar, lingkaran roda dibagi menjadi beberapa wilayah. Pada *Roulette Wheel Selection*, lebar suatu wilayah kromosom ditentukan menurut nilai *fitness*-nya. Semakin besar nilai *fitness*-nya maka semakin besar wilayahnya, dan semakin besar pula peluang kromosom tersebut untuk terpilih. Berikut ini adalah langkah-langkah melakukan seleksi *Roulette Wheel*:

$$P_i = \text{fitness}_i / \sum \text{fitness}_j$$

- 1) Hitung *fitness* suatu individu.
- 2) Hitung Probabilitas *fitness* individu dengan membagi nilai *fitness* dengan total *fitness* populasi.
- 3) Pilih bilangan acak 0-1.
- 4) Pilih individu sebagai *parent* berdasarkan nilai random pada (tahap 3) berada.
- 5) Sebagai contoh terdapat 10 individu seperti terlihat pada gambar di bawah ini:

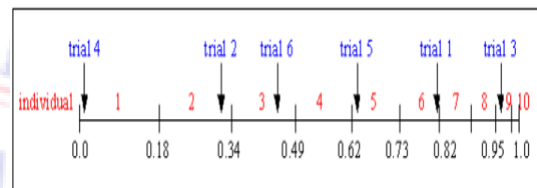
Tabel 2. Probabilitas *Fitness* untuk *Roulette Wheel Selection*

No Individu	1	2	3	4	5	6	7	8	9	10
<i>Fitness</i>	20	18	16	14	12	10	8	6	4	2
Probabilitas (P)	0.18	0.16	0.14	0.12	0.10	0.08	0.06	0.04	0.02	0.01

Selanjutnya dilakukan pemilihan bilangan acak 0-1 sejumlah individu yang ada, yaitu 10 individu sebagai berikut:

0.81 0.32 0.96 0.01 0.65 0.42

Hasil yang didapatkan setelah melalui proses seleksi



Gambar 1. Pointer *Roulette Wheel Selection*

Maka individu yang terpilih adalah individu 1, 2, 3, 5, 6, 9.

Perkawinan Silang (Crossover)

Perkawinan silang (*crossover*) adalah operator dalam Algoritma Genetika yang melibatkan dua orangtua untuk membentuk sebuah kromosom baru. Kromosom baru hasil penyilangan disebut anak atau *offspring*. Prinsip dari perkawinan silang ini adalah melakukan operasi pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses perkawinan silang ini dipengaruhi oleh probabilitas perkawinan silang (*pc*). Semakin besar nilai *Pc*, maka kemungkinan variasi individu juga akan semakin besar. Operator perkawinan silang ini bergantung pada representasi kromosom yang digunakan. Prinsip dari kawin silang ini adalah melakukan operasi pertukaran pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses kawin silang dilakukan pada setiap individu dengan probabilitas *crossover* yang telah ditentukan.

Operator kawin silang ini bergantung pada representasi pengkodean kromosom yang dilakukan. Pada pengkodean *integer* tidak mengharuskan untuk memperhatikan konsistensi urutan nilai pada kromosomnya sehingga banyak variasi operasi perkawinan silang yang dapat dilakukan, diantaranya (Suyanto, 2008):

Perkawinan Silang Satu Titik (One Point Crossover)

One point crossover menggunakan satu titik *p* yang dijadikan sebagai titik *crossover*. Titik *p*

merupakan sembarang titik di dalam kromosom yang dipilih secara acak. Jika jumlah gen pada kromosom *parent* adalah n , maka gen pada kromosom *offspring* adalah gabungan dari gen ke-1 sampai gen ke- p pada *parent* 1 digabung dengan gen ke- $p + 1$ sampai gen ke- n dari *parent* 2.

Tabel 3. Perkawinan Silang Satu Titik Representasi *Integer*

	crossover point							
Parent 1	1	3	5	6	2	8	4	7
Parent 2	6	4	7	1	8	3	2	5
Offspring 1	1	3	5	6	8	3	2	5
Offspring 2	6	4	7	1	2	8	4	7

Perkawinan Silang Dua Titik (*Two Point Crossover*)

Two point crossover hampir sama dengan *single point crossover*. Hanya saja pada *two point crossover* digunakan dua buah titik *crossover*. Proses persilangan dua titik dimulai dengan memilih dua titik perkawinan silang secara acak pada barisan bit kromosom orang tua. Kromosom baru hasil perkawinan silang akan terbentuk dengan menyalin barisan bit orang tuanya mulai dari barisan bit pertama sampai dengan titik perkawinan silang ke dua sampai dengan bit terakhir. Sisanya yaitu dari titik persilangan pertama sampai dengan titik persilangan ke dua disalin dari orang tua ke dua.

Tabel 4. Perkawinan Silang Dua Titik Representasi *Integer*

	crossover point 1				crossover point 2			
Parent 1	1	3	5	6	2	8	4	7
Parent 2	6	4	7	1	8	3	2	5
Offspring 1	1	3	7	1	8	3	4	7
Offspring 2	6	4	5	6	2	8	2	5

Perkawinan Silang Seragam (*Uniform Crossover*)

Uniform crossover sering disebut juga dengan *many point crossover*. Metode ini hampir sama dengan dua metode sebelumnya, hanya saja pada *uniform crossover* titik *crossover* yang digunakan tidak ditentukan. Titik *crossover* dipilih secara acak dan jumlah tidak ditentukan.

Tabel 5. Perkawinan Silang Seragam Representasi *Integer*

	crossover point 1			crossover point 2		crossover point 3		
Parent 1	1	3	5	6	2	8	4	7
Parent 2	6	4	7	1	8	3	2	5
Offspring 1	1	4	7	1	2	8	2	5
Offspring 2	1	3	5	6	2	8	4	7

Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom.

operator ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Tujuannya untuk menghindari konvergensi prematur, yaitu pencapaian suatu nilai/hasil yang belum atau bukan maksimal.

Proses mutasi dalam sistem biologi berlangsung dengan mengubah isi *allele* gen pada suatu *locus* dengan *allele* yang lain. Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan *fitness* yang lebih baik. Namun, mutasi ini tetap memberikan harapan agar kromosom yang diperoleh mempunyai *fitness* yang lebih baik dibandingkan sebelumnya. Operasi mutasi digunakan dengan probabilitas yang rendah agar tidak mengganggu kromosom baik yang telah diperoleh (Desiana dan Arhami, 2005).

Mutasi untuk representasi pengkodean *integer* dapat dilakukan dengan cara *random mutation* dan membalikan nilai (Suyanto, 2008). Pada *Random Mutation* gen yang terpilih secara acak nilainya diganti dengan menjadi nilai acak (*random*). Misalkan nilai yang terpilih 4 dan nilai randomnya adalah 8. Maka 4 diganti dengan 8. Contoh mutasi representasi *integer* menggunakan cara *random mutation* dapat dilihat pada Tabel 6.

Tabel 6. Mutasi Representasi *Integer* Menggunakan *Random Mutation*

Kromosom sebelum mutasi	1	2	3	4	6	8	5	7
Kromosom setelah mutasi	1	2	3	8	6	4	5	7

Pada cara membalikan nilai jika nilai gen berada pada interval $[1, 8]$, maka gen bernilai 1 dibalik menjadi 8, gen bernilai 2 menjadi 7, gen bernilai 3 menjadi 6, dan seterusnya. Contoh mutasi representasi *integer* menggunakan cara membalik nilai dapat dilihat pada Tabel 7.

Tabel 7. Mutasi Representasi *Integer* Menggunakan Pembalikan Nilai

Kromosom sebelum mutasi	1	2	3	4	6	8	5	7
Kromosom setelah mutasi	8	2	3	4	6	8	5	7

Update Generasi

Setelah proses seleksi, perkawinan silang dan mutasi yang dilakukan terhadap suatu populasi, proses berikutnya adalah pemilihan kromosom untuk membentuk generasi baru yang disebut *update generasi*. Dalam Algoritma Genetika Holland, proses *update* yang dilakukan adalah *update* secara generasi (*generational update*), yaitu menggantikan keseluruhan kromosom pada generasi sebelumnya dengan kromosom-kromosom yang didapat dari hasil seleksi, perkawinan silang dan mutasi.

Selain itu terdapat skema *update* secara kontinu (*continuous update*) yang mengizinkan *parent*

dan *offspring* untuk bercampur dalam satu generasi dengan memilih secara acak baik itu dari gen *parent* maupun *offspring*.

Suatu strategi metode yang baik untuk digunakan dengan skema *continous update* adalah *elitism*, yang berarti mempertahankan kromosom kromosom baik dalam populasi lama dengan menyertakannya lagi pada populasi baru. Sejumlah kromosom terbaik pada populasi lama dapat hilang karena proses perkawinan silang dan mutasi. Secara teori, *elitism* dapat meningkatkan kemampuan dari Algoritma Genetika karena mempertahankan kromosom yang baik dari populasi lama (Nugraha, 2011).

Parameter dalam Algoritma Genetika

Tidak ada metode pasti untuk menentukan nilai parameter dalam algoritma gentika. Kombinasi nilai yang tepat untuk parameter tersebut sangat dipengaruhi oleh permasalahan yang akan diselesaikan. Dalam penelitian optimasi menggunakan Algoritma Genetika, serangkaian pengujian pendahuluan diperlukan untuk mendapatkan kombinasi nilai parameter yang sesuai. Berikut parameter dalam Algoritma genetika (Syarif, 2014):

a. Ukuran Populasi

Yakni banyak kromosom yang mendiami suatu populasi dalam satu generasi. Semakin banyak dan beragam individu dalam populasi tersebut maka peluangnya akan semakin besar untuk mendapatkan individu yang mendekati sempurna. Apabila ukuran populasi yang terlalu besar juga tidak terlalu baik, dikarenakan akan membutuhkan waktu yang lebih lama untuk menyelesaikan masalah.

b. Jumlah Generasi

Menunjukkan banyaknya generasi yang akan dibangkitkan selama proses Algoritma Genetika. Paramater ini juga dijadikan sebagai salah satu kriteria berhenti dalam Algoritma Genetika. Jumlah generasi juga mempunyai andil yang besar dalam menentukan individu yang lebih baik, meskipun tidak berarti semakin besar jumlah generasi maka individu yang dihasilkan selalu baik. Hal ini disebabkan pada suatu saat dimana nilai *fitnes* semua individu akan sama atau konvergen.

c. Probabilitas Crossover (Pc)

Probabilitas kawin silang mempunyai nilai 0-1. Angka ini menunjukkan seberapa sering akan dilakukan perkawinan silang dalam satu generasi. Biasanya nilai Pc berkisar antara 0–1. Jika Pc = 100% berarti semua *offspring* dihasilkan dari proses *crossover*.

d. Probabilitas Mutasi (Pm)

Menunjukkan rasio perbandingan banyaknya gen dalam setiap kromosom yang akan bermutasi. Untuk menghasilkan individu yang lebih baik, nilai Pm harusnya diset sekecil mungkin agar tidak mengganggu kromosom-

koromsom dengan nilai *fitness* baik yang telah diperoleh

KESIMPULAN

Setelah melewati tahap perancangan dan implementasi program, maka dapat disimpulkan bahwa:

- Algoritma Genetika dapat digunakan untuk melakukan pencarian komponen *skateboard* berdasarkan anggaran dan kriteria tertentu secara efektif dan efisien.
- Hasil komponen *skateboard* dan kecepatan eksekusi program dipengaruhi oleh ukuran populasi dan jumlah generasi.
- Ukuran populasi dan jumlah generasi yang lebih besar menghasilkan nilai *fitness* yang lebih baik dikarenakan terbentuknya kombinasi kromosom yang lebih banyak, namun proses menjadi lebih lama.
- Hasil yang diberikan belum tentu sama dengan percobaan sebelumnya, dikarenakan Algoritma Gentika mencari himpunan solusi secara random.
- Hasil komponen *skateboard* yang dihasilkan hanyalah sebagai saran kepada calon pembeli, tidak mutlak harus dibeli. Keputusan tetap ditentukan oleh calon pembeli.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, bantuan, dan kontribusi dalam pelaksanaan penelitian yang berjudul “Optimasi Pemilihan Komponen Terbaik untuk Perakitan Skateboard dengan Algoritma Genetika Berbasis Java pada Toko DSRuntul.”

Ucapan terima kasih penulis sampaikan kepada Universitas Dian Nusantara sebagai institusi yang telah memberikan fasilitas, ilmu pengetahuan, serta lingkungan akademik yang mendukung terselenggaranya penelitian ini. Penulis juga mengucapkan terima kasih kepada Lembaga Riset dan Pengabdian Masyarakat (LRPM) atas dukungan, arahan, serta fasilitasi dalam pelaksanaan dan pengembangan penelitian ini.

Selain itu, penulis mengucapkan terima kasih kepada dosen pembimbing dan dosen pengajar Universitas Dian Nusantara atas bimbingan, arahan, serta ilmu yang telah diberikan. Ucapan terima kasih juga disampaikan kepada pihak Toko DSRuntul yang telah memberikan izin, data, dan informasi yang diperlukan selama proses penelitian.

DAFTAR PUSTAKA

- [1] D. Ariyanto and H. Prasetyo, “Implementasi algoritma genetika untuk optimasi pemilihan komponen komputer berbasis Java,” *Jurnal Teknologi dan Sistem Informasi*, vol. 7, no. 2, pp. 85–94, 2021.

- [2] A. Bali and S. Kusumadewi, "Penerapan algoritma genetika untuk sistem rekomendasi produk berbasis multi-kriteria," *Jurnal Ilmu Komputer dan Aplikasi*, vol. 8, no. 1, pp. 12–20, 2020.
- [3] T. R. Nugraha and R. Wibowo, "Sistem rekomendasi berbasis algoritma genetika untuk pemilihan komponen kendaraan listrik," *Jurnal Sistem Cerdas Indonesia*, vol. 10, no. 3, pp. 233–242, 2022.
- [4] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1998.
- [5] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994, doi: 10.1007/BF00175354.
- [6] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York: McGraw-Hill Education, 2020.
- [7] S. Kusumadewi and H. Purnomo, *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu, 2013.
- [8] Learning. Reading, MA: Addison-Wesley, 1989.
- [9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine*

