

Penggantian Resistor Rusak dengan Optimasi Susunan Paralel-Seri-Paralel Menggunakan Algoritma Genetika

Replacement of Faulty Resistors Using Optimization of Parallel-Series-Parallel Configuration Based on Genetic Algorithm

F.X. Wisnu Yudo Untoro*
Program Studi Informatika
Universitas Wijaya Kusuma Surabaya
Surabaya, Indonesia
wisnuyudo@uwks.ac.id

Tjatarsari Widiartin
Program Studi Informatika
Universitas Wijaya Kusuma Surabaya
Surabaya, Indonesia
widiartin@uwks.ac.id

Nonot Wisnu Karyanto
Program Studi Informatika
Universitas Wijaya Kusuma Surabaya
Surabaya, Indonesia
nonotwk@uwks.ac.id

Diterima : Mei 2025
Disetujui : Juni 2025
Dipublikasi : Juli 2025

Abstrak—Kerusakan resistor merupakan masalah umum yang sering terjadi dan menghambat perbaikan perangkat elektronik, terutama saat komponen pengganti dengan nilai resistansi yang sesuai dan spesifik tidak tersedia di pasaran. Penelitian ini mengusulkan penggunaan algoritma genetika untuk mengoptimalkan kombinasi susunan resistor paralel-seri-paralel sebagai solusi pengganti resistor yang rusak secara efektif. Algoritma genetika dipilih karena kemampuannya yang unggul dalam menemukan solusi optimal pada kondisi data yang terbatas. Dalam konteks ini, probabilitas *crossover* (*Pco*) merujuk pada peluang dua solusi (konfigurasi resistor) bertukar sebagian 'materi genetiknya' untuk menciptakan solusi baru, sementara mutasi (*Pmu*) adalah perubahan acak pada sebuah solusi yang membantu memperkenalkan keberagaman solusi dan mencegah algoritma terjebak pada lokal optimum. Simulasi dilakukan menggunakan bahasa pemrograman C++ untuk menghasilkan konfigurasi efisien dalam penggantian resistor rusak. Dengan memanfaatkan lima nilai resistansi yang tersedia (10Ω, 20Ω, 30Ω, 43Ω, dan 51Ω), hasil uji menunjukkan bahwa dengan menerapkan *Pco* 80–90% dan *Pmu* 10%, algoritma genetika mampu menghasilkan konfigurasi yang mendekati atau identik dengan resistansi target yang diinginkan dalam 2 hingga 7 generasi. Keberhasilan ini menegaskan bahwa metode yang diusulkan sangat aplikatif dan efisien dalam konteks perbaikan elektronik. Penelitian ini memberikan kontribusi signifikan dengan menawarkan solusi praktis dan tepat guna bagi teknisi dan penyedia jasa servis elektronik, terutama saat menghadapi kendala keterbatasan komponen, sehingga memungkinkan penggantian resistor yang cepat dan akurat di lapangan.

Kata Kunci—algoritma genetika; resistor rusak; susunan paralel-seri-paralel; optimasi resistansi; keterbatasan komponen

Abstract—Resistor damage is a common problem that frequently hinders electronic device repair, especially when suitable and specific replacement components with the correct resistance values are unavailable in the market. This research proposes the use of genetic algorithms to optimize the combination of parallel-series-parallel resistor arrangements as a solution for replacing

damaged resistors effectively. Genetic algorithms are chosen for their superior ability to find optimal solutions under limited data conditions. In this context, crossover probability (*Pco*) refers to the chance of two solutions (resistor configurations) exchanging parts of their 'genetic material' to create new solutions, while mutation (*Pmu*) is a random change in a solution that helps introduce solution diversity and prevents the algorithm from getting trapped in local optima. Simulations were conducted using the C programming language to generate efficient configurations for damaged resistor replacement. By utilizing five available resistance values (10Ω, 20Ω, 30Ω, 43Ω, and 51Ω), test results show that by applying *Pco* of 80–90% and *Pmu* of 10%, the genetic algorithm is capable of producing configurations that approximate or are identical to the target resistance desired within 2 to 7 generations. This success confirms that the proposed method is highly applicable and efficient in the context of electronic repairs. This research makes a significant contribution by offering a practical and effective solution for technicians and electronic service providers, especially when facing component limitations, thereby enabling quick and accurate resistor replacement in the field.

Keywords— genetic algorithm; damaged resistor; parallel-series-parallel configuration; resistance optimization; component availability limitation

I. PENDAHULUAN

Resistor merupakan salah satu komponen fundamental dalam elektronika. Fungsi utamanya adalah mengatur aliran arus listrik dalam suatu rangkaian. Namun, resistor rentan terhadap kerusakan. Kerusakan ini dapat terjadi akibat panas berlebih, tegangan yang melampaui batas spesifikasi, atau faktor lingkungan yang tidak mendukung, sebagaimana dijelaskan dalam [1]. Kerusakan resistor seringkali menghambat perbaikan perangkat elektronik, terutama jika nilai resistansi pengganti yang sesuai tidak tersedia di pasaran. Penelitian ini mengusulkan penggunaan algoritma

genetika untuk mengoptimalkan kombinasi susunan resistor paralel-seri-paralel sebagai solusi pengganti resistor rusak.

Dalam upaya memahami konfigurasi dasar resistor, hubungan seri dan paralel resistansi dijelaskan secara rinci melalui prinsip Hukum Ohm [2-3]. Pemahaman mengenai perbedaan perilaku arus dan resistansi dalam konfigurasi seri dan paralel juga telah banyak dianalisis dalam literatur, termasuk dalam studi terbaru oleh [4], yang membandingkan karakteristik arus dan resistansi pada kedua jenis konfigurasi tersebut. Struktur jaringan resistor dalam susunan paralel-seri-paralel turut dianalisis dari perspektif matematis, seperti melalui formula "read once" untuk jaringan seri-paralel [5], yang memberikan dasar teoritis penting dalam pengembangan konfigurasi resistansi alternatif.

Proses manual untuk menentukan kombinasi optimal resistor dalam konfigurasi paralel-seri-paralel menjadi semakin sulit dan rentan kesalahan seiring bertambahnya jumlah kombinasi yang mungkin. Kesulitan ini timbul karena ruang pencarian solusi yang sangat luas, di mana setiap penambahan resistor atau perubahan posisi dalam konfigurasi akan secara eksponensial meningkatkan jumlah kombinasi unik yang harus dievaluasi. Oleh karena itu, metode optimasi berbasis algoritma genetika menjadi salah satu solusi potensial. Algoritma genetika mengemulasi prinsip seleksi alam untuk mencari solusi optimal dan telah banyak digunakan dalam optimasi kombinasi sirkuit elektronik [6-7].

Algoritma genetika (AG) bekerja dengan menciptakan "populasi" awal dari solusi-solusi yang mungkin (dalam kasus ini, konfigurasi resistor). Setiap solusi dievaluasi berdasarkan seberapa baik kinerjanya (disebut sebagai *fitness* atau kebugaran), yaitu seberapa dekat resistansi yang dihasilkan dengan resistansi target. Solusi yang lebih baik memiliki peluang lebih besar untuk "bertahan hidup" dan "bereproduksi". Melalui proses seleksi ini, diikuti dengan operator genetik seperti *crossover* (penggabungan bagian dari dua solusi "induk" untuk menciptakan solusi "anak" baru) dan mutasi (perubahan acak kecil pada solusi), algoritma secara iteratif menghasilkan populasi baru yang diharapkan semakin mendekati solusi optimal dari waktu ke waktu.

Dalam beberapa penelitian, penggunaan algoritma genetika telah terbukti efektif dalam berbagai skenario optimasi teknis, seperti kompensasi daya reaktif dalam sistem distribusi [8], optimasi penempatan kapasitor bank pada jaringan distribusi [25], optimasi penempatan dan ukuran distributed generation pada sistem distribusi radial [26], prediksi produksi panas pada perangkat elektronik [9], dan pengembangan berbagai teknik eksplorasi dan eksploitasi ruang solusi [10-11]. Pendekatan inisialisasi populasi secara strategis untuk memperluas pencarian solusi juga telah diteliti dalam literatur terkini [12]. Selain algoritma genetika, terdapat juga pendekatan optimasi berbasis metaheuristik lainnya seperti *Particle Swarm Optimization* (PSO), *Simulated Annealing* (SA), dan *Ant Colony Optimization* (ACO) yang telah diterapkan untuk berbagai masalah optimasi, termasuk dalam desain dan analisis sirkuit. Studi-studi *state-of-the-art* yang relevan telah mengaplikasikan algoritma genetika untuk membangun kombinasi resistansi paralel [23] dan kombinasi paralel-seri [24] yang setara dengan resistansi rusak. Meskipun demikian, sebagian besar fokus penelitian sebelumnya adalah pada desain sirkuit baru, optimasi sistem secara makro, atau konfigurasi resistor yang lebih sederhana. Kesenjangan dalam literatur saat ini adalah kurangnya metode yang spesifik dan efisien untuk mengoptimalkan penggantian resistor yang rusak pada perangkat elektronik

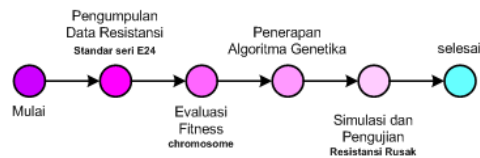
menggunakan kombinasi susunan paralel-seri-paralel secara lebih kompleks dalam kondisi nyata di mana stok komponen pengganti yang tepat sangat terbatas.

Selain itu, prinsip-prinsip dasar dan mekanisme kerja algoritma genetika secara komprehensif dijelaskan dalam literatur klasik oleh Goldberg [13], yang menjadi referensi utama dalam penerapan metode ini. Oleh karena itu, penelitian ini secara eksplisit bertujuan untuk mengembangkan algoritma genetika guna menghasilkan kombinasi konfigurasi resistor paralel-seri-paralel dengan resistansi ekuivalen mendekati target tertentu dalam kondisi data terbatas.

II. METODE

Bagian ini menjelaskan metode yang digunakan dalam penelitian untuk menentukan konfigurasi optimal susunan paralel-seri-paralel sebagai pengganti resistor yang rusak. Proses metodologis dimulai dari pengambilan data resistansi berdasarkan standar seri E24 [14-15], dilanjutkan dengan pemodelan solusi menggunakan algoritma genetika, serta evaluasi *fitness* dan simulasi untuk menilai keakuratan konfigurasi yang dihasilkan.

Rangkaian proses penelitian dari awal hingga diperoleh konfigurasi optimal digambarkan secara komprehensif dalam diagram alur yang ditunjukkan pada Gambar 1.



Gambar 1: Diagram alur proses penelitian untuk pencarian konfigurasi paralel-seri-paralel

Gambar ini menunjukkan alur kerja utama penelitian secara berurutan, dimulai dari Mulai, diikuti oleh Pengumpulan Data Resistansi (berdasarkan standar seri E24), lalu Evaluasi Fitness (pada kromosom), dilanjutkan dengan Penerapan Algoritma Genetika, kemudian Simulasi dan Pengujian (terhadap resistansi rusak), dan diakhiri dengan Selesai.

A. Ketersediaan Data Resistansi

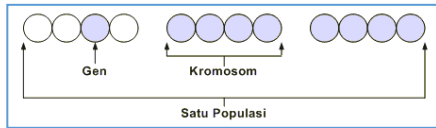
Nilai resistansi yang digunakan berasal dari resistor standar seri E24 yang umum tersedia di pasaran, dipilih untuk merepresentasikan kondisi keterbatasan ketersediaan resistor dalam praktik perbaikan. Data ini disusun dalam berkas teks eksternal bernama `dataR-UWKS.txt`. Tabel 1 berikut adalah nilai resistansi yang tersedia.

TABEL 1. KETERSEDIAAN DATA RESISTANSI

No	Resistansi (Ω)
1	10
2	20
3	30
4	43
5	51

B. Representasi Kromosom

Dalam algoritma genetika, setiap solusi direpresentasikan sebagai sebuah kromosom yang terdiri dari empat gen, seperti diilustrasikan pada Gambar 2. Masing-masing gen merepresentasikan satu nilai resistansi, yang kombinasinya akan disusun dalam konfigurasi paralel-seri-paralel.



Gambar 2: Ilustrasi struktur kromosom yang terdiri dari empat gen, dilengkapi dengan anotasi fungsi masing-masing gen

Gambar ini menunjukkan konsep *kromosom* sebagai sekumpulan *gen* (lingkaran yang berisi nilai resistansi), dan kumpulan kromosom membentuk *satu populasi*. Setiap gen merepresentasikan satu nilai resistansi yang akan membentuk konfigurasi paralel-seri-paralel.

C. Evaluasi Fitness

Penentuan optimalitas kombinasi resistor dalam konfigurasi paralel-seri-paralel memerlukan perhitungan total resistansi ekuivalen (R_{eq}). Konfigurasi ini terdiri dari dua pasangan resistor yang disusun paralel, kemudian digabungkan secara seri.

Berdasarkan konsep dasar Hukum Ohm, resistansi dua resistor yang disusun paralel dihitung dengan rumus:

$$R_{p1} = \frac{R_1 R_2}{R_1 + R_2} \quad (1)$$

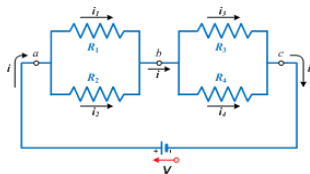
Demikian pula, resistansi dari pasangan kedua dihitung dengan:

$$R'_{p2} = \frac{R_3 R_4}{R_3 + R_4} \quad (2)$$

Persamaan (1) dan (2) tersebut selanjutnya digabungkan dalam konfigurasi seri untuk menghasilkan R_{eq} berikut:

$$R_{eq} = R_{p1} + R'_{p2} \quad (3)$$

Struktur konfigurasi ini digunakan sebagai solusi pengganti resistor yang rusak, di mana pemilihan nilai resistansi berasal dari kumpulan data yang tersedia untuk mendekati nilai resistansi awal perangkat elektronik. Representasi visual dari konfigurasi ini ditunjukkan pada Gambar 3.



Gambar 3. Struktur konfigurasi resistor dalam susunan paralel-seri-paralel.

Gambar 3 menampilkan skematik rangkaian, menunjukkan dua resistor (R_1, R_2) terhubung paralel dan dua resistor lainnya (R_3, R_4) juga terhubung paralel. Kedua blok paralel ini kemudian terhubung secara seri, dengan R_{eq_total} yang diukur dari titik awal 'a' ke titik akhir 'c'. Sumber tegangan 'V' diaplikasikan pada rangkaian.

Setelah memperoleh R_{eq} , setiap individu dalam populasi algoritma genetika dievaluasi menggunakan fungsi *fitness*. Fungsi *fitness* mengukur tingkat kedekatan antara R_{eq_total} dan resistansi target (R_{target}). Proses evaluasi *fitness* ini sangat penting karena secara langsung mencerminkan seberapa baik suatu solusi memenuhi tujuan optimasi, yaitu seberapa dekat resistansi yang dihasilkan dengan resistansi target. Rumus *fitness* yang digunakan adalah sebagai berikut:

$$Fitness = \frac{1}{|R_{target} - R_{eq}| + \epsilon} \quad (4)$$

di mana

- R_{target} adalah resistansi target dari resistor yang rusak
- R_{eq_total} adalah total resistansi ekuivalen dari konfigurasi paralel-seri-paralel
- ϵ adalah nilai kecil untuk menghindari pembagian dengan nol.

Penambahan konstanta kecil ϵ menghindari nilai tak hingga saat $R_{target} = R_{eq_total}$ [16]. Nilai *fitness* yang mendekati $1/\epsilon$ menunjukkan konfigurasi tersebut semakin optimal.

D. Algoritma Genetika

Algoritma genetika merupakan metode pencarian dan optimasi berdasarkan prinsip seleksi alam. Dalam penelitian ini, Algoritma Genetika digunakan untuk mencari konfigurasi resistor paralel-seri-paralel yang memiliki resistansi ekuivalen mendekati resistansi target. Prosesnya meliputi inisialisasi populasi, evaluasi *fitness* untuk setiap individu (solusi potensial), serta iterasi evolusi melalui operator seleksi, *crossover*, dan mutasi untuk menghasilkan generasi yang lebih adaptif [17-18]. Iterasi ini berlanjut hingga ditemukan solusi optimal.

E. Operator Algoritma Genetika

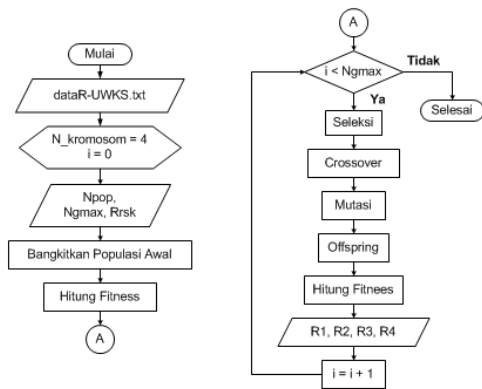
Dalam penelitian ini, algoritma genetika menggunakan empat operator utama untuk evolusi populasi menuju solusi optimal:

1. Seleksi. Untuk memilih individu terbaik dari populasi berdasarkan nilai *fitness*, penelitian ini menggunakan metode Roulette Wheel Selection, di mana probabilitas terpilihnya sebanding dengan *fitness* individu [17].
2. Crossover. Operator ini menggabungkan dua individu (*parent*) untuk menghasilkan *offspring* baru yang mewarisi karakteristik keduanya, mempercepat eksplorasi solusi dan mempertahankan kombinasi baik [18].
3. Mutasi. Mutasi mengubah nilai gen secara acak pada individu terpilih untuk mempertahankan keragaman populasi dan mencegah konvergensi prematur ke solusi lokal. Tingkat mutasi ditetapkan 10% dari total populasi, berdasarkan efektivitas yang ditunjukkan dalam studi [19-20] untuk berbagai masalah optimasi.
4. Pembentukan Generasi Baru. *Offspring* dari *crossover* dan mutasi dievaluasi *fitness*-nya, lalu populasi generasi berikutnya dibentuk dengan mempertahankan individu terbaik dan mengganti individu dengan *fitness* rendah, mendorong evolusi menuju konfigurasi resistor optimal.

F. Algoritma Pencarian Kombinasi Resistansi Optimal Paralel-Seri-Paralel

Penelitian ini merancang algoritma berbasis evolusi menggunakan struktur dasar algoritma genetika untuk menemukan kombinasi resistor paralel-seri-paralel yang optimal.

Alur proses pencarian kombinasi resistansi optimal disajikan secara visual dalam *flowchart* pada Gambar 4, yang menggambarkan langkah-langkah secara detail. Proses ini dimulai dari inisialisasi data resistansi dan pembangkitan populasi awal, dilanjutkan dengan evaluasi *fitness*, seleksi, *crossover*, mutasi, serta evaluasi dan pergantian generasi secara iteratif hingga konvergensi atau batas generasi maksimum tercapai.



Gambar 4: Algoritma pencarian kombinasi paralel-seri-paralel resistansi optimal.

Gambar 4 adalah diagram alir (flowchart) yang menunjukkan langkah-langkah detail dari algoritma pencarian kombinasi resistansi optimal. Alurnya dimulai dari Mulai, diikuti oleh inisialisasi data (`dataR-UWKS.txt`), pengaturan parameter ($N_{kromosom}=4$, N_{pop} , N_{gmax} , R_{rsk}). Kemudian dilanjutkan ke Bangkitkan Populasi Awal, Hitung Fitness. Diikuti oleh loop iterasi ($i < N_{gmax}$), yang di dalamnya terdapat Seleksi, Crossover, Mutasi, Offspring, Hitung Fitness kembali, dan pencatatan hasil (R_1, R_2, R_3, R_4), hingga mencapai kriteria konvergensi atau generasi maksimum, dan diakhiri dengan Selesai.

G. Implementasi Algoritma di Borland C++ 5.02

Algoritma diimplementasikan menggunakan Borland C++ 5.02, dipilih karena kecepatan eksekusi, kemudahan akses berkas eksternal (`dataR-UWKS.txt`), dan fleksibilitas manipulasi struktur data. Program dirancang untuk bekerja secara otomatis, dari membaca data resistansi hingga menjalankan seluruh proses algoritma genetika (inisialisasi, evaluasi *fitness*, seleksi, *crossover*, mutasi) hingga konvergensi atau generasi maksimum tercapai. Hasil konfigurasi optimal disimpan dan ditampilkan secara otomatis.

H. Simulasi dan Pengujian

Simulasi mengevaluasi kinerja algoritma genetika dalam mencari konfigurasi resistansi optimal pada susunan paralel-seri-paralel. Implementasi menggunakan Borland C++ 5.02, dengan input data resistansi eksternal terbatas, resistansi rusak, dan parameter operator algoritma genetika.

Tujuan utama simulasi ini adalah untuk:

1. Mengamati efektivitas algoritma genetika dalam mencapai resistansi ekuivalen mendekati resistansi target.
2. Menganalisis pengaruh variasi probabilitas *crossover* terhadap waktu konvergensi dan kualitas solusi.
3. Memvalidasi kestabilan hasil optimasi terhadap variasi populasi dan parameter mutasi.

Parameter simulasi yang digunakan dalam penelitian ini dirangkum pada Tabel 3.

TABEL 3. PARAMETER SIMULASI ALGORITMA GENETIKA

Parameter	Nilai	Keterangan
N_{pop}	10	Ukuran populasi dalam setiap generasi
P_{co}	75%, 80%, 90%	Tingkat pertukaran gen antar individu
P_{mu}	10%	Tingkat perubahan gen acak
N_{Gmax}	50 iterasi	Batas maksimal iterasi algoritma

Pemilihan ukuran populasi 10 individu didasarkan pada pertimbangan efisiensi komputasi untuk masalah optimasi dengan ruang solusi terbatas [18]. Batas generasi maksimum

50 iterasi dipilih untuk memastikan konvergensi algoritma menuju solusi optimal tanpa membebani sumber daya komputasi secara berlebihan, sesuai praktik umum dalam literatur optimasi iteratif.

Setiap skenario probabilitas *crossover* diuji hingga konvergensi atau mencapai jumlah generasi maksimum, dengan probabilitas mutasi dan ukuran populasi dijaga tetap. Hasil simulasi kemudian dianalisis dalam bagian "Hasil dan Pembahasan" untuk mengidentifikasi konfigurasi resistansi terbaik serta memahami dinamika evolusi.

I. Black Box Testing

Pengujian menggunakan pendekatan Black Box Testing dilakukan untuk memastikan program implementasi algoritma genetika berfungsi sesuai spesifikasi [21-22]. Metode ini memvalidasi fungsionalitas sistem dari perspektif pengguna, tanpa mengacu pada struktur internal kode.

Skenario pengujian meliputi:

1. Program mampu membaca dan memproses data resistansi dari berkas eksternal `dataR-UWKS.txt`.
2. Program dapat membangkitkan populasi awal dengan konfigurasi kromosom yang sesuai.
3. Operator seleksi, *crossover*, dan mutasi berfungsi sesuai parameter yang ditentukan.
4. Nilai *fitness* dihitung dengan benar berdasarkan persamaan yang dirumuskan.
5. Program dapat mendeteksi konvergensi atau menghentikan proses pada generasi maksimum.

Hasil Black Box Testing menunjukkan fungsi-fungsi utama program berjalan sesuai spesifikasi di berbagai skenario input awal. Evaluasi *output* lebih rinci disajikan dalam Hasil dan Pembahasan.

III. HASIL DAN PEMBAHASAN

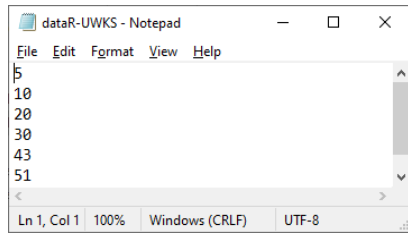
Pada bagian ini, disajikan hasil implementasi dan analisis kinerja algoritma genetika dalam mengoptimalkan konfigurasi resistansi paralel-seri-paralel sebagai solusi pengganti resistor yang rusak. Simulasi dilakukan menggunakan *dataset* resistansi terbatas, sebagaimana disimpan dalam berkas eksternal `dataR-UWKS.txt`. Optimasi dilakukan melalui beberapa tahap, mulai dari validasi fungsi dasar program hingga eksplorasi konfigurasi optimal dengan variasi parameter algoritma genetika.

Setiap tahap pengujian bertujuan untuk memastikan bahwa sistem dapat membaca data resistansi, membangkitkan populasi awal, memilih individu terbaik, serta melakukan proses evolusi secara efektif. Hasil-hasil simulasi dianalisis untuk mengevaluasi hubungan antara probabilitas *crossover*, distribusi resistansi awal, dan kecepatan konvergensi solusi menuju resistansi target. Analisis konvergensi, dalam konteks ini, merujuk pada evaluasi seberapa cepat dan akurat algoritma dapat menemukan solusi optimal yang mendekati resistansi target dari waktu ke waktu, dengan tujuan akhir menemukan konfigurasi resistansi yang optimal meskipun ada keterbatasan data.

A. Uji Pembacaan Ketersediaan Data Resistansi

Tahap awal dalam pengujian adalah memastikan program dapat membaca data resistansi dari berkas eksternal `dataR-UWKS.txt` dengan benar. Berkas ini berisi lima nilai resistansi yang mewakili kondisi keterbatasan ketersediaan resistor di pasaran.

Fungsi `BacaDataR()` diimplementasikan untuk membaca seluruh data resistansi ke dalam *array* internal program. Hasil pembacaan data divisualisasikan melalui tampilan tabel 1, sebagaimana diperlihatkan pada Gambar 5 dan Gambar 6. Tampilan ini menunjukkan output program secara umum, yaitu daftar nilai resistansi yang terbaca dan konfirmasi jumlah data.



Gambar 5. Data resistansi dalam berkas eksternal.

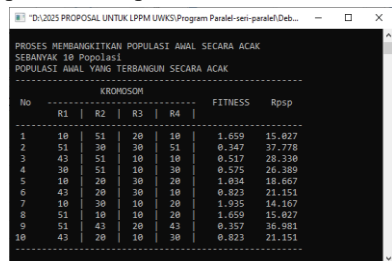


Gambar 6: Verifikasi jumlah data resistansi yang terbaca

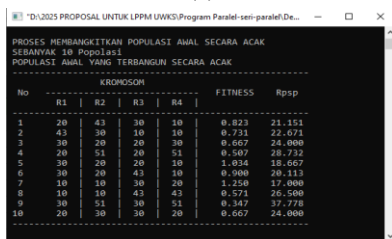
Berdasarkan pengujian, seluruh nilai resistansi berhasil dibaca dengan akurat, dan jumlah data yang terdeteksi sesuai dengan jumlah yang tercatat dalam berkas input. Hal ini menunjukkan bahwa mekanisme pengambilan data eksternal berjalan sesuai spesifikasi dan siap untuk proses selanjutnya

B. Uji Pembangkitan Populasi Awal

Setelah memastikan data resistansi terbaca dengan benar, tahap selanjutnya adalah membangkitkan populasi awal yang akan digunakan dalam proses algoritma genetika. Populasi awal dibentuk dengan mengambil empat nilai resistansi secara acak dari kumpulan data yang tersedia, untuk membentuk satu kromosom individu dalam konfigurasi paralel-seri-paralel.



(a)



(b)

Gambar 7: (a) Daftar individu populasi awal hasil pembangkitan acak, (b) Visualisasi keragaman konfigurasi resistansi dalam populasi awal

Fungsi `BangkitkanPopulasiAwal()` digunakan untuk menghasilkan sepuluh individu awal, sesuai dengan parameter ukuran populasi yang telah ditetapkan. Visualisasi hasil pembangkitan populasi awal ditunjukkan pada Gambar 7a dan Gambar 7b. Tampilan ini merepresentasikan

bagaimana program menghasilkan berbagai kombinasi resistor secara acak sebagai titik awal optimasi.

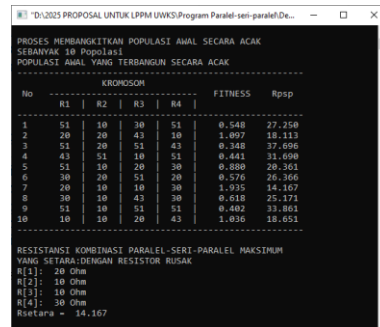
Berdasarkan pengujian, populasi yang dihasilkan menunjukkan variasi kombinasi nilai resistansi yang cukup beragam. Hal ini penting untuk mendukung mekanisme eksplorasi solusi dalam algoritma genetika, karena keberagaman populasi awal berkontribusi dalam memperluas ruang pencarian solusi potensial [19].

C. Uji Pemilihan Individu Maksimum

Tahap berikutnya adalah menguji fungsi pemilihan individu terbaik berdasarkan nilai *fitness* tertinggi. Pemilihan individu terbaik sangat penting dalam algoritma genetika karena menentukan kandidat solusi yang akan diturunkan ke generasi berikutnya..

Fungsi `CariIndividuMaks()` diimplementasikan untuk memilih individu dengan nilai *fitness* tertinggi dari seluruh populasi. *Fitness* dihitung berdasarkan kedekatan R_{eq_total} terhadap R_{target} , sebagaimana dijelaskan dalam bagian metodologi.

Visualisasi hasil pemilihan individu maksimum ditunjukkan pada Gambar 8. Gambar ini menunjukkan keluaran program berupa data individu dengan nilai *fitness* terbaik pada suatu generasi.

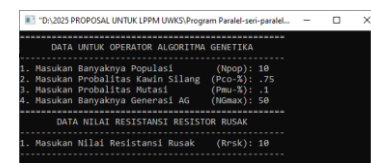


Gambar 8: Tampilan individu dengan *fitness* maksimum dalam populasi

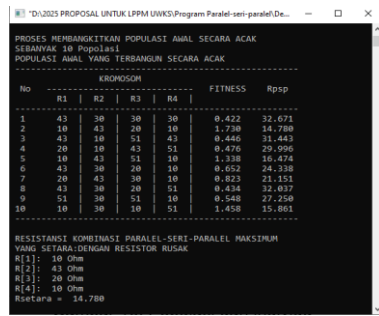
Berdasarkan pengujian, fungsi ini mampu mengidentifikasi individu dengan *fitness* terbaik secara konsisten di berbagai skenario data populasi. Hal ini memastikan bahwa proses seleksi dalam algoritma genetika dapat berjalan efektif, mempertahankan kualitas solusi dari satu generasi ke generasi berikutnya [17].

D. Simulasi Parameter 1: Probabilitas Crossover 75% & Target Resistansi 10Ω

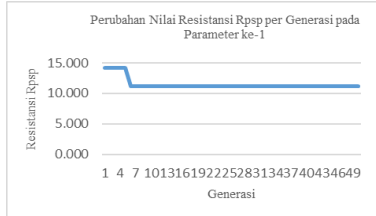
Simulasi pertama menguji kinerja AG dengan N_{pop} 10, P_{co} 75%, P_{mu} 10% dan resistansi target 10Ω. Gambar 9 menampilkan data masukan yang digunakan untuk simulasi ini. Pada populasi awal 10 individu menghasilkan individu terbaik di generasi ke-0 dengan R_{eq} 14.780Ω dan *fitness* 1.730 (Gambar 10). Pada generasi 6 (Gambar 11), ditemukan konfigurasi optimal dengan R_{eq} 11.167Ω dengan *fitness* 4.615 (Tabel 5).



Gambar 9: Data masukan untuk simulasi parameter ke-1



Gambar 10. Populasi awal dan individu maksimum di generasi ke-0



Gambar 11. Evolusi resistansi individu hingga optimum pada generasi ke-6.

Gambar 11 menampilkan grafik yang menunjukkan nilai *fitness* atau resistansi ekuivalen dari individu terbaik berubah seiring bertambahnya generasi, hingga mencapai generasi ke-6 menjadi stabil sebagai solusi optimalnya.

Tabel 5 merangkum hasil evaluasi *fitness* dari individu maksimum pada generasi 0 dan generasi 6.

TABEL 5. HASIL EVALUASI PARAMETER KE-1

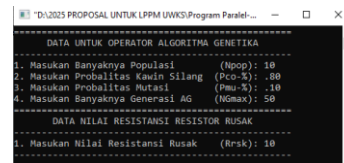
Generasi	Individu Maksimum				Req Ω	Fitness
	R1 Ω	R2 Ω	R3 Ω	R4 Ω		
0	20	43	20	10	14.780	1.730
6	10	10	20	10	11.167	4.615

Analisis Konvergensi dan Temuan Utama

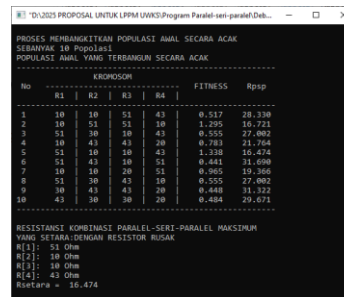
Penerapan Pco sebesar 75% menghasilkan konvergensi yang cenderung lambat karena ruang eksplorasi solusi yang lebih sempit. Distribusi resistansi populasi awal yang dominan pada nilai resistansi besar (30 Ω dan 43 Ω) menyulitkan algoritma untuk menemukan kombinasi resistansi rendah yang lebih dekat ke target 10 Ω . Keberagaman populasi tetap terjaga hingga generasi ke-6, namun tidak cukup untuk menemukan solusi dengan resistansi tepat 10 Ω . Temuan ini mengindikasikan bahwa pengaturan probabilitas *crossover* dan pemilihan data resistansi sangat mempengaruhi kecepatan dan efektivitas optimasi resistansi ekuivalen [19-20]. Untuk mencapai resistansi target 10 Ω secara presisi, diperlukan penyesuaian parameter lebih lanjut atau eksplorasi lebih luas. Hasil ini selaras dengan prinsip dasar algoritma genetika bahwa *crossover* yang lebih rendah membatasi eksplorasi, sehingga memerlukan waktu lebih lama untuk menemukan solusi optimal pada ruang pencarian yang kompleks [18].

E. Simulasi Parameter 2: Probabilitas Crossover 80% & Target Resistansi 10 Ω

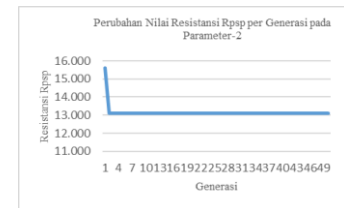
Simulasi kedua menggunakan Npop 10, Pco 80%, Pmu 10%, target 10 Ω , dan batas 50 generasi (Gambar 12). Populasi awal 10 individu menghasilkan individu terbaik generasi ke-0 dengan Req 16.474 Ω dan *fitness* 1.338 (Gambar 13). Pada generasi ke-2 (Gambar 14), ditemukan konfigurasi optimal dengan Req 13.113 Ω dan *fitness* 2.431 (Tabel 6).



Gambar 12. Data masukan simulasi parameter ke-2



Gambar 13. Populasi awal dan individu maksimum generasi ke-0



Gambar 14. Evolusi resistansi individu hingga optimum pada generasi ke-2

Gambar 14 menampilkan grafik yang menunjukkan nilai *fitness* atau resistansi ekuivalen dari individu terbaik berubah seiring bertambahnya generasi, hingga mencapai generasi ke-6 menjadi stabil sebagai solusi optimalnya.

Tabel 6 merangkum hasil evaluasi *fitness* dari individu maksimum pada generasi 0 dan generasi 2.

TABEL 6 HASIL EVALUASI PARAMETER KE-2

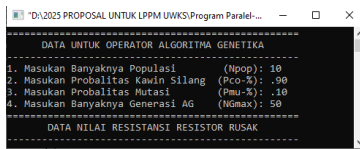
Generasi	Individu Maksimum				Req Ω	Fitness
	R1 Ω	R2 Ω	R3 Ω	R4 Ω		
0	53	10	10	43	16.474	1.338
2	10	10	10	43	13.113	2.431

Analisis Konvergensi dan Temuan Utama

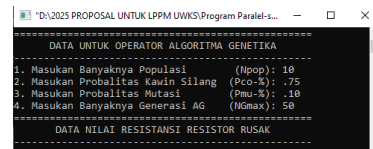
Peningkatan Pco dari 75% menjadi 80% berhasil mempercepat konvergensi solusi. Terbukti, individu dengan *fitness* lebih baik ditemukan pada generasi ke-2, lebih cepat dibanding parameter ke-1 (generasi ke-6). Meskipun demikian, resistansi ekuivalen yang diperoleh masih belum mencapai target 10 Ω secara presisi. Di sini Pmu 10% cukup efektif mempertahankan keberagaman populasi, namun dominasi nilai resistansi tinggi dalam populasi awal tetap menjadi faktor penghambat konvergensi penuh. Temuan ini mengindikasikan bahwa peningkatan Pco perlu dikombinasikan dengan pengendalian distribusi awal resistansi untuk mencapai optimasi lebih efektif [19-20]. Hasil ini mendukung teori bahwa peningkatan *crossover* yang moderat meningkatkan kemampuan eksplorasi algoritma genetika, memungkinkan pencarian solusi yang lebih efisien dibandingkan tingkat *crossover* yang terlalu rendah [18].

F. Simulasi Parameter 3: Probabilitas Crossover 90% & Target Resistansi 10 Ω

Simulasi ketiga menggunakan Npop 10, Pco 90%, Pmu 10%, batas 50 generasi, dan target 10 Ω (Gambar 15). Populasi awal 10 individu menghasilkan individu terbaik generasi ke-0 dengan Req 17.000 Ω dan *fitness* 1.250 (Gambar 16). Pada generasi ke-7 (Gambar 17), ditemukan konfigurasi optimal dengan Req 10.000 Ω dan *fitness* 10.000 (Tabel 7), yang persis sama dengan resistansi target.



Gambar 15. Data masukan simulasi parameter ke-3



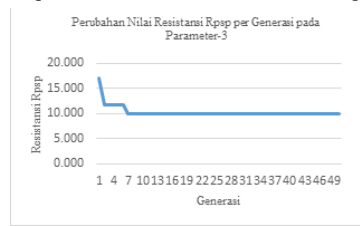
Gambar 18. Data masukan simulasi parameter ke-4.

PROSES HEBANGKETTAN POPULASI AWAL SECARA ACAK
SEBANYAK 10 Populasi
POPULASI AWAL YANG TERBANGUN SECARA ACAK

No	KROMOSOM				FITNESS	Rppp
	R1	R2	R3	R4		
1	51	20	51	51	0.324	39.866
2	43	20	10	43	0.783	21.764
3	51	30	30	43	0.363	36.569
4	51	10	43	20	0.269	22.011
5	20	30	10	20	1.024	18.667
6	20	20	30	43	0.464	29.671
7	10	40	51	43	0.458	30.839
8	10	43	43	20	0.783	21.764
9	30	20	10	10	1.250	17.000
10	30	10	30	51	0.575	26.389

RESISTANSI KOMBINASI PARALEL-SERI-PARALEL MAKSIMUM
YANG SETARA DENGAN RESISTOR RUSAK
R[1]: 30 Ohm
R[2]: 20 Ohm
R[3]: 10 Ohm
R[4]: 10 Ohm
Rsetara = 17.000

Gambar 16. Populasi awal dan individu maksimum generasi ke-0



Gambar 17. Evolusi resistansi individu hingga optimum pada generasi ke-7

Gambar 17 menampilkan grafik yang menunjukkan nilai *fitness* atau resistansi ekuivalen dari individu terbaik berubah seiring bertambahnya generasi, hingga mencapai generasi ke-7 menjadi stabil sebagai solusi optimalnya.

Tabel 7 merangkum hasil evaluasi *fitness* dari individu maksimum pada generasi 0 dan generasi 7.

TABEL 7. HASIL EVALUASI PARAMETER KE-3

Generasi	Individu Maksimum				Req Ω	Fitness
	R1 Ω	R2 Ω	R3 Ω	R4 Ω		
0	30	20	10	10	17.000	1.250
7	10	10	10	10	10.000	10.000

Analisis Konvergensi dan Temuan Utama

Penerapan Pco sebesar 90% meningkatkan eksplorasi ruang solusi secara signifikan, sehingga algoritma berhasil menemukan kombinasi resistansi yang tepat sesuai dengan target 10 Ω . Meskipun membutuhkan lebih banyak generasi dibandingkan simulasi parameter ke-2, hasil akhirnya lebih akurat dan stabil. Temuan ini menguatkan teori bahwa peningkatan Pco memperluas ruang eksplorasi, namun memerlukan pengaturan yang seimbang untuk menjaga stabilitas konvergensi [19-20]. Tingkat *crossover* yang tinggi seperti ini memungkinkan algoritma menjelajahi kombinasi genetik yang lebih luas, konsisten dengan prinsip-prinsip algoritma genetika untuk pencarian global [18].

G. Simulasi Parameter 4: Probabilitas Crossover 75% & Target Resistansi 12.5 Ω

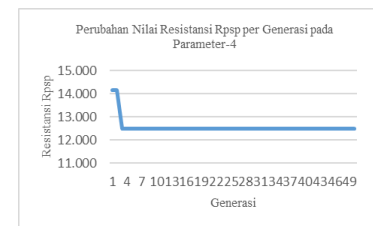
Simulasi keempat mengevaluasi Pco 75% dengan target 12.5 Ω memakai Npop 10, Pmu 10%, dan batas 50 generasi (Gambar 18). Populasi awal 10 individu menghasilkan individu terbaik generasi ke-0 dengan Req 14.167 Ω dan *fitness* 4.688 (Gambar 19). Pada generasi ke-3 (Gambar 20), ditemukan konfigurasi optimal dengan Req 12.500 Ω dan *fitness* 12.500 (Tabel 8), persis sama dengan resistansi target.

PROSES HEBANGKETTAN POPULASI AWAL SECARA ACAK
SEBANYAK 10 Populasi
POPULASI AWAL YANG TERBANGUN SECARA ACAK

No	KROMOSOM				FITNESS	Rppp
	R1	R2	R3	R4		
1	43	20	30	10	1.250	21.151
2	10	20	30	51	0.889	25.556
3	20	30	10	43	1.431	26.113
4	43	20	30	51	0.504	32.540
5	51	51	43	30	0.395	43.171
6	20	10	30	10	4.000	14.167
7	30	30	20	43	0.729	28.051
8	20	20	20	43	1.029	23.051
9	51	51	30	51	0.300	44.100
10	51	20	30	10	1.280	21.806

RESISTANSI KOMBINASI PARALEL-SERI-PARALEL MAKSIMUM
YANG SETARA DENGAN RESISTOR RUSAK
R[1]: 20 Ohm
R[2]: 10 Ohm
R[3]: 30 Ohm
R[4]: 10 Ohm
Rsetara = 14.167

Gambar 19. Populasi awal dan individu maksimum pada generasi ke-0



Gambar 20. Evolusi resistansi individu hingga optimum pada generasi ke-3

Gambar 20 menampilkan grafik yang menunjukkan nilai *fitness* atau resistansi ekuivalen dari individu terbaik berubah seiring bertambahnya generasi, hingga mencapai generasi ke-3 menjadi stabil sebagai solusi optimalnya.

Tabel 8 merangkum hasil evaluasi *fitness* dari individu maksimum pada generasi 0 dan generasi 3.

TABEL 8. HASIL EVALUASI PARAMETER KE-4

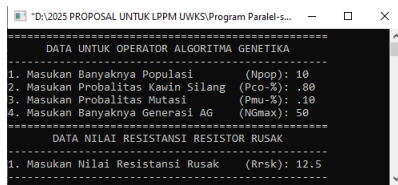
Generasi	Individu Maksimum				Req Ω	Fitness
	R1 Ω	R2 Ω	R3 Ω	R4 Ω		
0	20	10	30	10	14.167	4.688
3	10	10	30	10	12.500	12.500

Analisis Konvergensi dan Temuan Utama

Penerapan Pco 75% terbukti efektif dalam meng eksplorasi solusi optimal tanpa menyebabkan fluktuasi besar alam nilai *fitness* antar generasi. Konvergensi terhadap esistensi target 12.5 Ω tercapai hanya dalam 3 generasi, lebih epat dibandingkan skenario sebelumnya. Di sini, Pmu 10% berperan dalam menjaga keberagaman populasi sekaligus menghindari perubahan ekstrem pada konfigurasi individu. Hasil ini memperkuat pentingnya keseimbangan antara eksplorasi (melalui *crossover*) dan stabilisasi populasi (melalui mutasi) dalam optimasi berbasis algoritma genetika [19-20]. Tingkat *crossover* yang lebih rendah pada target ini dapat memicu konvergensi yang lebih cepat jika ruang solusi optimal tidak terlalu tersebar luas, mendukung prinsip eksploitasi yang lebih kuat [18].

H. Simulasi Parameter 5: Probabilitas Crossover 80% & Target Resistansi 12.5 Ω

Simulasi kelima mengevaluasi Pco 80% dengan target 12.5 Ω , Npop 10, Pmu 10%, dan batas 50 generasi (Gambar 21). Populasi awal 10 individu menghasilkan individu terbaik generasi ke-0 dengan Req 14.167 Ω dan *fitness* 4.688 (Gambar 22). Pada generasi ke-2 (Gambar 23), ditemukan individu optimal dengan Req 12.500 Ω (Tabel 9), persis sama dengan resistansi target.



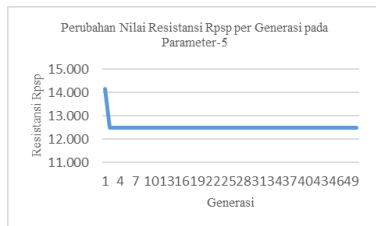
Gambar 21. Data masukan simulasi parameter ke-5

PROSES MEMBANGKITKAN POPULASI AWAL SECARA ACAK
SEBANYAK 10 Populasi
POPULASI AWAL YANG TERBANGUN SECARA ACAK

No	KROMOSOM				FITNESS	Rpsp
	R1	R2	R3	R4		
1	30	43	30	51	0.499	36.569
2	10	10	20	43	1.110	22.071
3	51	43	10	43	0.527	31.443
4	43	30	43	10	0.875	25.784
5	43	30	51	43	0.824	41.001
6	51	10	43	30	0.860	26.032
7	43	20	30	10	1.295	21.151
8	20	30	20	43	0.688	29.672
9	51	51	30	10	0.581	33.000
10	10	20	30	10	4.688	14.167

RESISTANSI KOMBINASI PARALEL-SERI-PARALEL MAKSIMUM
YANG SETARA-DENGAN RESISTOR RUSAK
R[1]: 10 Ohm
R[2]: 20 Ohm
R[3]: 30 Ohm
R[4]: 10 Ohm
Rsetara = 14.167

Gambar 22. Populasi awal dan individu maksimum generasi ke-0



Gambar 23. Evolusi resistansi individu hingga pada generasi ke-2

Gambar 23 menampilkan grafik yang menunjukkan nilai *fitness* atau resistansi ekuivalen dari individu terbaik berubah seiring bertambahnya generasi, hingga mencapai generasi ke-2 menjadi stabil sebagai solusi optimalnya.

Tabel 9 merangkum hasil evaluasi *fitness* dari individu maksimum pada generasi 0 dan generasi 2.

TABEL 9. HASIL EVALUASI PARAMETER KE-5

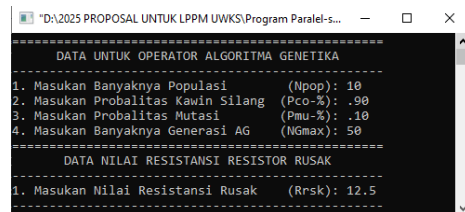
Generasi	Individu Maksimum				Req Ω	Fitness
	R1 Ω	R2 Ω	R3 Ω	R4 Ω		
0	10	20	30	10	14.167	4.688
2	10	10	30	10	12.500	12.500

Analisis Konvergensi dan Temuan Utama

Perapan Pco 80% terbukti mempercepat eksplorasi ruang solusi, dibandingkan simulasi dengan Pco 75%. Waktu konvergensi yang lebih cepat (generasi ke-2) menunjukkan bahwa peningkatan *crossover* memperbaiki adaptasi individu terhadap resistansi target. Di sini terlihat bahwa Pmu 10% efektif dalam menjaga keberagaman populasi, meskipun efeknya terhadap percepatan konvergensi tidak sekuat peran *crossover*. Distribusi resistansi awal yang cukup merata antara nilai rendah dan menengah juga berkontribusi dalam mempercepat pencapaian solusi optimal. Temuan ini mendukung teori bahwa keseimbangan eksplorasi dan eksploitasi dalam algoritma genetika dapat dicapai dengan pengaturan *crossover* yang tepat [19-20]. Efisiensi konvergensi pada tingkat *crossover* ini menunjukkan titik optimal antara pencarian solusi baru dan mempertahankan karakteristik solusi baik yang sudah ada [18].

I. Simulasi Parameter 6: Probabilitas Crossover 90% & Target Resistansi 12.5 Ω

Simulasi keenam mengevaluasi Pco 90% dengan target 12.5 Ω , Npop 10, Pmu 10%, dan batas 50 generasi (Gambar 24). Populasi awal 10 individu menghasilkan individu terbaik generasi ke-0 dengan Req 15.613 Ω dan *fitness* 3.039 (Gambar 25). Pada generasi ke-10 (Gambar 26), ditemukan individu optimal dengan Req 12.500 Ω (Tabel 10), persis sama dengan resistansi target.



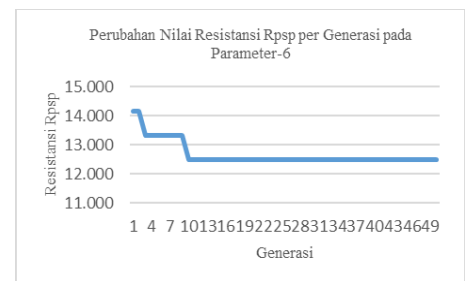
Gambar 24. Data masukan simulasi parameter ke-6

PROSES MEMBANGKITKAN POPULASI AWAL SECARA ACAK
SEBANYAK 10 Populasi
POPULASI AWAL YANG TERBANGUN SECARA ACAK

No	KROMOSOM				FITNESS	Rpsp
	R1	R2	R3	R4		
1	30	20	30	43	0.688	29.671
2	30	30	51	20	0.760	29.366
3	51	51	30	43	0.395	43.171
4	51	30	20	30	0.645	30.889
5	20	43	10	10	1.748	18.651
6	30	10	43	43	0.714	29.080
7	20	43	20	43	0.791	27.302
8	51	51	20	43	0.452	39.151
9	30	20	20	10	1.744	18.667
10	43	10	10	30	3.039	15.613

RESISTANSI KOMBINASI PARALEL-SERI-PARALEL MAKSIMUM
YANG SETARA-DENGAN RESISTOR RUSAK
R[1]: 43 Ohm
R[2]: 10 Ohm
R[3]: 10 Ohm
R[4]: 30 Ohm
Rsetara = 15.613

Gambar 25. Populasi awal dan individu maksimum generasi ke-0



Gambar 26. Evolusi resistansi individu hingga pada generasi ke-10

Tabel 10 merangkum hasil evaluasi *fitness* dari individu maksimum pada generasi 0 dan generasi 10.

TABEL 10. HASIL EVALUASI PARAMETER KE-6

Generasi	Individu Maksimum				Req Ω	Fitness
	R1 Ω	R2 Ω	R3 Ω	R4 Ω		
0	43	10	10	30	15.613	3.039
10	10	10	10	30	12.500	12.500

Analisis Konvergensi dan Temuan Utama

Penerapan Pco sebesar 90% meningkatkan luas eksplorasi solusi, sehingga memperbanyak variasi individu dalam populasi. Namun, eksplorasi yang luas tersebut menyebabkan waktu konvergensi lebih panjang, dengan pencapaian solusi optimal baru terjadi pada generasi ke-10. Probabilitas mutasi sebesar 10% cukup menjaga keberagaman populasi, tetapi untuk mempercepat eksploitasi, disarankan mengeksplorasi peningkatan tingkat mutasi atau menurunkan *crossover* ke kisaran 80–85% dalam penelitian selanjutnya. Temuan ini sejalan dengan teori bahwa peningkatan *crossover* mendukung eksplorasi, tetapi perlu dikombinasikan dengan strategi eksploitasi yang lebih agresif untuk mempercepat konvergensi solusi [19-20]. Pco yang sangat tinggi dapat menyebabkan pencarian kurang terfokus pada solusi baik yang sudah ditemukan, sesuai dengan prinsip *trade-off* eksplorasi-eksploitasi dalam optimasi [11].

J. Ringkasan Hasil dan Pembahasan

Hasil simulasi secara komprehensif menunjukkan bahwa peningkatan probabilitas *crossover* (Pco) dari 75% hingga 90% secara signifikan meningkatkan eksplorasi ruang solusi dalam optimasi konfigurasi resistansi paralel-seri-

paralel. Pada Pco 90%, kombinasi resistansi presisi terhadap target berhasil ditemukan (simulasi ke-3 dan ke-6), meskipun memerlukan lebih banyak generasi untuk konvergensi.

Sebaliknya, Pco yang lebih rendah (75%–80%) mempercepat konvergensi, namun solusi yang ditemukan kadang hanya mendekati tetapi tidak tepat mencapai resistansi target (simulasi ke-1 dan ke-2).

Distribusi nilai resistansi dalam populasi awal juga terbukti memiliki pengaruh krusial; dominasi nilai resistansi rendah (10Ω, 20Ω) mempercepat pencapaian solusi optimal, sementara dominasi nilai resistansi besar (30Ω, 43Ω) cenderung memperlambat konvergensi.

Probabilitas mutasi sebesar 10% terbukti efektif dalam menjaga keberagaman populasi. Namun, eksplorasi lebih lanjut terhadap adaptasi laju mutasi direkomendasikan untuk meningkatkan performa algoritma genetika dalam aplikasi ke depan, terutama dalam mencapai keseimbangan antara eksplorasi dan eksploitasi.

Secara keseluruhan, penelitian ini dengan jelas menunjukkan bahwa algoritma genetika mampu mengoptimalkan kombinasi resistansi dalam konfigurasi paralel-seri-paralel secara efektif meskipun terdapat keterbatasan dalam ketersediaan data resistansi. Hasil ini sangat dipengaruhi oleh pengaturan parameter algoritma dan karakteristik populasi awal, yang menegaskan potensi metode ini sebagai solusi aplikatif untuk penggantian resistor rusak.

TABEL 11. RINGKASAN HASIL SIMULASI PARAMETER ALGORITMA GENETIKA

Simulasi	Pco (%)	Pmu (%)	Target R (Ω)	Avg Req Gen 0	Final Req	Generasi Konvergen	Solusi vs Target
1	75	10	10	14.780	11.167	6	Mendekati
2	80	10	10	16.474	13.113	2	Mendekati
3	90	10	10	17.000	10.000	7	Tepat Sama
4	75	10	12.5	14.167	12.500	3	Tepat Sama
5	80	10	12.5	14.167	12.500	2	Tepat Sama
6	90	10	12.5	15.613	12.500	10	Tepat Sama

IV. KESIMPULAN

Penelitian ini berhasil menunjukkan bahwa algoritma genetika efektif dalam mengoptimalkan konfigurasi resistor paralel-seri-paralel sebagai solusi pengganti resistor rusak, khususnya dalam kondisi keterbatasan ketersediaan komponen. Berikut adalah temuan-temuan penting dari penelitian ini:

1. Probabilitas *crossover* sebesar 90% terbukti paling efektif dalam menemukan solusi resistansi yang presisi dan identik dengan target (misalnya, 10Ω dan 12.5Ω), meskipun terkadang memerlukan jumlah generasi yang lebih banyak.
2. Probabilitas *crossover* yang lebih rendah (75%–80%) mampu mempercepat konvergensi menuju solusi, namun seringkali menghasilkan resistansi ekuivalen yang mendekati target tanpa mencapai presisi sempurna.
3. Distribusi resistansi dalam populasi awal memiliki pengaruh signifikan; populasi dengan variasi resistansi yang lebih merata atau dominasi nilai rendah cenderung mempercepat konvergensi. Keberagaman ini memungkinkan algoritma untuk menjelajahi ruang

solusi yang lebih luas secara efisien, mengurangi risiko terjebak pada lokal optimum, dan mempercepat pencarian kombinasi resistansi yang mendekati target.

4. Probabilitas mutasi 10% secara konsisten menjaga keberagaman populasi, mencegah algoritma terjebak pada lokal optimum dan mendukung pencarian solusi optimal.

Secara praktis, pendekatan ini menawarkan solusi yang relevan dan dapat diterapkan bagi teknisi dan penyedia jasa servis elektronik untuk mengatasi kendala stok komponen, memungkinkan penggantian resistor yang akurat dan tepat waktu di industri elektronika.

Rekomendasi

Untuk penelitian lanjutan, beberapa area yang dapat dieksplorasi meliputi:

1. Eksplorasi metode optimasi lain seperti algoritma koloni semut (Ant Colony Optimization) atau algoritma *particle swarm optimization* (PSO) untuk membandingkan efisiensi dan akurasi dalam permasalahan serupa.
2. Penerapan algoritma genetika pada konfigurasi rangkaian elektronik yang lebih kompleks atau untuk optimasi penggantian jenis komponen lain seperti kapasitor atau induktor.
3. Pengujian algoritma dengan *dataset* resistansi yang lebih luas dan beragam (misalnya, seri E48 atau E96) untuk menguji skalabilitas dan robustnesnya.
4. Pengembangan antarmuka pengguna yang lebih intuitif untuk aplikasi praktis di lapangan, serta mempertimbangkan implementasi pada sistem *hardware* untuk validasi waktu nyata.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Wijaya Kusuma Surabaya atas dukungan pendanaan yang telah diberikan dalam pelaksanaan penelitian ini.

Dukungan tersebut memungkinkan pengembangan dan penerapan algoritma genetika untuk optimasi konfigurasi kombinasi resistansi paralel-seri-paralel, serta berkontribusi terhadap kemajuan riset di bidang optimasi berbasis evolusi dan teknik kelistrikan.

REFERENSI

- [1] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics*, 9th ed. Hoboken, NJ: John Wiley & Sons, Inc., 2011.
- [2] J. E. Parks, "Ohm's Law III -- Resistors in Series and Parallel," Aug. 2007. [Online]. Available: <http://www.phys.utk.edu/labs/Ohms%20Law%20Series%20Parallel%20Resistors.pdf>.
- [3] F. Asadi, "Resistors," in *Electric Circuits Laboratory Manual*. Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-24552-7_2.
- [4] K. A. Imtiyaaza, "Analyzing Ohm's Law: Comparison of Current and Resistance in Series and Parallel Circuits," *Scholar J. Phys. Educ.*, vol. 5, no. 4, pp. 142–149, Dec. 2024.
- [5] D. Mundici, "The read once formula of a series–parallel network," *Discrete Appl. Math.*, vol. 320, pp. 304–310, 2022, doi: 10.1016/j.dam.2022.06.002.
- [6] A. Zemliak, A. Osadchuk, and C. Serrano, "Optimization Process by Generalized Genetic Algorithm," *WSEAS Trans. Circuits Syst.*, vol. 23, pp. 37–45, 2024. [Online]. Available:

- [https://www.wseas.com/journals/cas/2024/a085101-004\(2024\).pdf](https://www.wseas.com/journals/cas/2024/a085101-004(2024).pdf).
- [7] A. Lberni, M. A. Marktani, A. Ahaitouf, and A. Ahaitouf, "Analog circuit sizing based on Evolutionary Algorithms and Deep Learning," *Expert Syst. Appl.*, vol. 237, Part B, pp. 121480–121482, 2024, doi: 10.1016/j.eswa.2023.121480.
 - [8] J. M. Lujano-Rojas et al., "Embedding quasi-static time series within a genetic algorithm for stochastic optimization: the case of reactive power compensation on distribution systems," *J. Comput. Des. Eng.*, vol. 7, no. 2, pp. 177–187, Apr. 2020, doi: 10.1093/jcde/qwaa016.
 - [9] Z. Dong, "Genetic algorithm-based analysis of heat production prediction in electronic devices," *Appl. Math. Nonlinear Sci.*, vol. 9, no. 1, pp. 1–10, Jan. 2024, doi: 10.2478/amns.2023.1.00388.
 - [10] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.
 - [11] A. Hussain and Y. S. Muhammad, "Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator," *Complex Intell. Syst.*, vol. 6, no. 1, pp. 1–14, 2020, doi: 10.1007/s40747-019-0102-7.
 - [12] Q. Li, S.-Y. Liu, and X.-S. Yang, "Influence of Initialization on the Performance of Metaheuristic Optimizers," *arXiv preprint arXiv:2003.03789*, 2020. [Online]. Available: <https://arxiv.org/abs/2003.03789>.
 - [13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
 - [14] IEC 60063:2015, *Preferred number series for resistors and capacitors*. International Electrotechnical Commission, 2015.
 - [15] IEEE Std 315-1975, "IEEE Standard Graphic Symbols for Electrical and Electronics Diagrams." IEEE, 1975.
 - [16] A. Kimuya, "The Modified Ohm's Law and Its Implications for Electrical Circuit Analysis," *Eur. J. Sci. Tech. (EJSET)*, vol. 4, no. 2, pp. 59–70, Dec. 2023, doi: 10.55696/ejset.1373552.
 - [17] A. Raj et al., "Applications of Genetic Algorithm with Integrated Machine Learning," in *2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, Uttar Pradesh, India, 2023, pp. 1–6, doi: 10.1109/ICIPTM57143.2023.10118328.
 - [18] A. Sohail, "Genetic Algorithms in the Fields of Artificial Intelligence and Data Sciences," *Ann. Data Sci.*, vol. 10, pp. 1007–1018, Aug. 2021, doi: 10.1007/s40745-021-00354-9.
 - [19] R. Salman, S. Suprpto, I. Irfandi, and O. Y. Hutajulu, "Optimization of Genetic Algorithm Computation Time with Mutation Probability Variations in Course Scheduling," *Jambura Journal of Electrical and Electronics Engineering*, vol. 5, no. 1, pp. 10–19, 2023.
 - [20] A. Kumar, B. Liu, R. Miikkulainen, and P. Stone, "Effective Mutation Rate Adaptation through Group Elite Selection," *arXiv preprint*, arXiv:2204.04817, 2022. Available: <https://arxiv.org/abs/2204.04817>.
 - [21] GeeksforGeeks, "Software Engineering | Black Box Testing," Available: <https://www.geeksforgeeks.org/software-engineering-black-box-testing/>
 - [22] Z. Aghababaeyan, M. Abdellatif, L. Briand, R. S. and M. Bagherzadeh, "Black-Box Testing of Deep Neural Networks through Test Case Diversity," *IEEE Transactions on Software Engineering*, vol. 49, no. 5, pp. 3182–3204, May 2023. DOI: 10.1109/TSE.2023.3243522.
 - [23] F.X. W. Y. Untoro, "Pemanfaatan Algoritma Genetika Untuk Membangun Kombinasi Paralel Resistansi Yang Setara Dengan Resistansi Rusak", j. nas. pendidik. teknik. inform., vol. 8, no. 2, pp. 94–108, Aug. 2019, doi: 10.23887/janapati.v8i2.17193.
 - [24] F.X. Wisnu Yudo Untoro, "Optimasi Nilai Resistansi pada Rangkaian Resistansi Kombinasi Paralel-Seri yang Setara dengan Nilai Resistansi Rusak Menggunakan Algoritma Genetika", *Journal Technology of Civil, Electrical, Mechanical, Geology, Mining, and Urban Design*, vol. 4, no. 2, pp. 53–60, Dec. 2019, doi: 10.33579/krvtk.v4i2.1562.
 - [25] R. Umar and S. Humena, "Optimasi Penempatan Capacitor Bank pada Jaringan Distribusi 20 kV Menggunakan Algoritma Genetika (Studi Kasus: Feeder Isimu)," *Jambura J. Electr. Electron. Eng.*, vol. 3, no. 2, pp. 119-125, 2021.
 - [26] M. S. Ibrahim, A. M. Nur, and R. Ismail, "Optimasi Penempatan dan Ukuran Distributed Generation Menggunakan Algoritma Genetika pada Sistem Distribusi Radial," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 1-9, 2023.