

PEMANFAATAN ALGORITMA GENETIKA UNTUK MEMBANGUN KOMBINASI PARALEL NILAI RESISTANSI RESISTOR YANG SETARA DENGAN NILAI RESISTANSI RUSAK

F.X. Wisnu Yudo Untoro

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma
Surabaya, Indonesia

e-mail: wisnusakti2410@gmail.com

ABSTRACT

Within a certain period of time electronic items will be damaged. This damage is caused by electronic components that do not work properly. One of the electronic components that can be damaged is the resistor. On the side of electronic goods service providers, the availability of resistor components is quite limited. Limited availability of these resistors can cause problems if resistance from damaged electronic goods is not available. Noting this, the proposed study is the search for damaged resistance through data availability of resistors using algorithm genetics. In this proposal, inputting a broken resistance value will give an output in the form of a combination of parallel resistance which is fair with faulty resistance. The application of the 0.95 crossover probability parameters and the probability of a 0.1 mutation, the genetic algorithm has been able to provide a parallel combination of resistance results that are equivalent to broken resistance.

Keywords: genetic algorithm, combination parallel resistance value.

ABSTRAK

Dalam kurun waktu tertentu barang elektronik akan mengalami kerusakan. Kerusakan ini disebabkan adanya komponen elektronik yang tidak bekerja dengan semestinya. Salah satu komponen elektronik yang bisa rusak ini adalah resistor. Di sisi penyedia jasa service barang elektronik, ketersediaan komponen resistornya cukup terbatas. Terbatasnya ketersediaan resistor ini dapat menimbulkan masalah bila resistansi dari barang elektronik yang rusak tidak tersedia. Memperhatikan hal tersebut usulan penelitian ini adalah pencarian resistansi yang rusak melalui data ketersediaan resistor menggunakan algoritma genetika. Dalam usulan ini, masukan nilai resistansi rusak akan memberikan luaran berupa kombinasi paralel resistansi yang semilirisitas dengan resistansi yang rusak. Penerapan parameter probabilitas crossover 0.95 dan probabilitas mutasi 0.1, algoritma genetika telah mampu memberikan hasil kombinasi paralel resistansi yang setara dengan resistansi rusak.

Kata kunci: algoritma genetika, kombinasi paralel nilai resistansi.

PENDAHULUAN

Barang elektronik merupakan suatu benda yang mempergunakan rangkaian elektronika yang dibentuk atau bekerja untuk menghasilkan sesuatu atas dasar rangkaian elektronika tersebut.

Fakta menunjukkan bahwa barang elektronik yang dapat bekerja sesuai

dengan rangkaian elektronika pembangunannya memerlukan yang disebut catu daya. Catu daya untuk barang elektronik ini dapat berupa daya listrik atau arus bolak-balik (AC) maupun daya baterai atau arus searah (DC) [4].

Seiring dengan berjalannya waktu, barang elektronik akan mengalami

kerusakan. Salah satu komponen pada barang elektronik yang rusak adalah komponen resistor [7]. Agar barang elektronik itu dapat bekerja lagi maka komponen resistor yang rusak perlu diganti dengan resistor yang mempunyai nilai resistansi yang sama.

Memperhatikan keterbatasan ketersediaan komponen resistor yang dimiliki oleh seorang jasa *service* barang elektronik tentunya untuk memperoleh resistor dengan nilai resistansi yang sesuai dengan yang rusak menjadi permasalahan tersendiri. Oleh karena komponen resistor dalam rangkaian elektronik adalah dipergunakan untuk membatasi jumlah arus yang masuk dalam rangkaian elektronika [3], sehingga cukup beresiko bila nilai resistansi resistor yang rusak diganti dengan nilai resistansi yang sembarangan.

Walaupun sudah terbayangkan bahwa cara *trial and error* kombinasi paralel nilai resistansi resistor merupakan cara yang dapat dipergunakan untuk memperoleh nilai resistansi rusak (pengganti/harapan) lebih akurat, namun kelemahan cara tersebut adalah membutuhkan waktu yang lama untuk menemukan kombinasi paralelnya.

Terkait dengan tujuan dari penelitian ini adalah menemukan sebuah solusi berupa susunan kombinasi paralel resistor yang menghasilkan sebuah nilai kesetaraan resistansi dengan nilai resistansinya resistor yang rusak [7] (tidak bekerja dengan semestinya) berdasarkan sejumlah n resistor yang tersedia pada seorang jasa *service* maka diperlukan alternatif lain.

Dalam [10] mengatakan bahwa suatu algoritma untuk pemecahan masalah dapat memberikan hasil pemecahan masalah yang benar maupun salah. Memperhatikan hal tersebut maka dalam penelitian ini permasalahan tentang pencarian susunan kombinasi paralel resistor akan dicoba diselesaikan menggunakan algoritma genetika dengan menggunakan operator kromosom string biner.

Pemilihan kromosom string biner atas dasar bahwa algoritma genetika dapat bekerja cukup baik untuk menyelesaikan masalah TSP [16], optimasi fungsi tanpa kendala [13], pemodelan optimasi [17], deteksi gangguan pada jaringan listrik [9], dan mempunyai analogi antara evolusi

biologis dengan algoritma genetika biner [14], serta sebagai salah satu strategi penggunaan algoritma genetika [12]. Dengan demikian, sangat dimungkinkan penerapan algoritma genetika biner ini untuk menemukan solusi susunan kombinasi paralel resistor.

KAJIAN TEORI

A. Kombinasi

Dalam [15], kombinasi ialah pengambilan satu atau lebih benda tanpa memandang urutannya. Dalam kombinasi ini berlaku dua teorema kombinasi sebagai berikut

Teorema 1 (kombinasi)

Banyaknya kombinasi k benda dari n benda tanpa ulangan adalah

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\cdots(n-k+1)}{1\cdot 2\cdots k} \quad (1)$$

Artinya banyaknya himpunan k benda yang dapat disusun dari n benda, dengan syarat setiap dua himpunan tidak boleh terdiri atas k yang sama.

Teorema 2 (kombinasi)

Banyaknya kombinasi k benda dari n benda dengan ulangan adalah

$$\binom{n+k-1}{k} \quad (2)$$

Artinya banyaknya himpunan k benda yang dapat disusun dari n benda, setiap benda dapat digunakan sesering sekehendak kita.

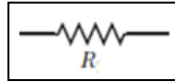
Sebagai ilustrasi, dari tiga huruf a, b, c dapat dapat dibentuk tiga kombinasi yang masing-masing terdiri atas 2 huruf tanpa ulangan, yaitu ab, ac, bc , dan enam kombinasi dengan ulangan, yaitu $ab, ac, bc, aa, bb, dan cc$.

Konsep kombinasi yang dipergunakan dalam penelitian ini mengacu pada teorema 2 ditambah satu unsur susunan posisi, sehingga untuk kasus pada ilustrasi di atas akan memberikan 9 kombinasi, yaitu $ab, ba, ac, ca, bc, cb, aa, bb, dan cc$.

B. Resistor

Pada dasarnya resistor adalah komponen elektronika pasif yang memiliki

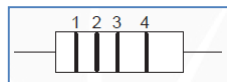
nilai resistansi atau hambatan tertentu yang berfungsi untuk membatasi dan mengatur arus listrik yang mengalir dalam suatu rangkaian elektronika [1][3][4][20].



Gambar 1. Simbol dan lambang pada resistor tetap

Sebuah resistor dapat bernilai tetap dan bernilai tidak tetap/variabel [5]. Simbol untuk resistor dengan nilai tetap memiliki sedikit berbeda dibanding resistor bernilai tidak tetap/variabel. Oleh karena pada penelitian ini terarah pada resistor bernilai tetap maka simbol dan lambang untuk resistor tersebut seperti ditunjukkan pada gambar 1, sedangkan untuk satuan nilai resistansi dari resistor adalah Ohm (Ω)

Nilai resistansi dari suatu resistor tetap dapat diketahui melalui kode warna gelang yang terdapat pada resistor tersebut. Kode warna gelang pada resistor memiliki 3 macam, yaitu 4, 5, dan 6 gelang [6][7]. Tabel 1 adalah tabel yang menunjukkan kode warna gelang untuk resistor dengan 4 gelang seperti yang terlihat pada gambar 2.



Gambar 2. Resistor tetap dengan 4 gelang.

Tabel 1. Kode warna gelang pada resistor

Kode warna	Gelang 1	Gelang 2	Gelang 3	Gelang 4
Hitam	0	0	10^0	-
Coklat	1	1	10^1	-
Merah	2	2	10^2	-
Orange	3	3	10^3	-
Kuning	4	4	10^4	-
Hijau	5	5	10^5	-
Biru	6	6	10^6	-
Ungu	7	7	10^7	-
Abu-abu	8	8	10^8	-
Putih	9	9	10^9	-
Emas	-	-	10^{-1}	5%
Perak	-	-	10^{-2}	10%
Tak berwarna	-	-	-	20%

Nilai resistansi sebuah resistor

Dalam [6][7] mengatakan bahwa nilai resistansi dari suatu resistor dapat ditentukan berdasarkan kode warna gelang yang

terdapat pada resistor tersebut. Sebagai contoh sebuah resistor pada gambar 3 menunjukkan resistor dengan 4 (empat), yaitu gelang ke 1 berwarna coklat, gelang ke 2 berwarna hitam, gelang ke 3 berwarna hitam, dan gelang ke 4 berwarna emas.



Gambar 3. Resistor dengan 4 warna gelang coklat, hitam, hitam dan emas.

Berdasarkan tabel 1 maka nilai resistansi dari resistor pada gambar 3 sebagai berikut

$$10 \times 10^0 = 10 \Omega$$

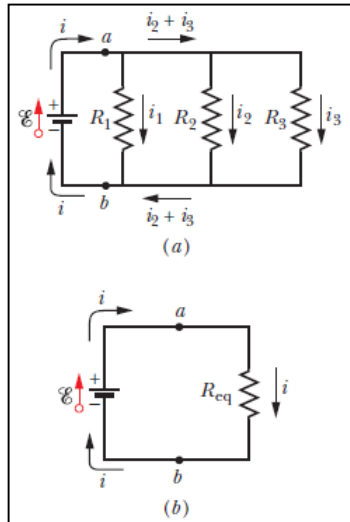
dengan toleransi sebesar 5%. Artinya bahwa resistor tersebut mempunyai nilai resistansi dengan rentang antara 9.5 – 10.5 Ohm.

Dalam [7] juga menyebutkan bahwa jika hasil pengukuran suatu resistor tidak sesuai (bisa lebih kecil atau lebih besar atau tidak ada hambatan sama sekali) dengan nilai yang tertera pada kode warna maka resistor itu dapat dikatakan resistor tersebut rusak.

C. RESISTOR TERHUBUNG PARALEL

Resistor dikatakan terhubung paralel jika dan hanya jika beberapa resistor dirangkai berjajar dalam satu kolom dengan masing-masing ujung resistor satu dengan yang lainnya bertemu dalam satu titik yang sama.

Dalam [4], bentuk resistor terhubung paralel dijelaskan dengan menggunakan gambar 4a, dimana pada gambar 4a tersebut menunjukkan tiga buah resistansi yang terhubung secara paralel dengan baterai yang ideal g . Istilah “paralel” mempunyai arti resistansi dihubungkan secara langsung pada satu sisi dan dihubungkan secara langsung pada satu sisi lainnya, dan beda potensial V diterapkan di seluruh sisi yang terhubung. Dengan demikian, ketiga resistansi itu memiliki beda potensial yang sama, menghasilkan arus melalui masing-masing resistansi. Secara umum dinyatakan bahwa *Ketika beda potensial V diterapkan pada resistansi yang terhubung secara paralel, semua resistansi memiliki beda potensial yang sama dengan V .*



Halliday & Resnick, hal. 741

Gambar 4.(a) Tiga resistor terhubung secara paralel melintasi titik a dan b. (b) rangkaian yang setara, dengan tiga resistor diganti dengan resistansi setara yaitu R_{eq} .

Pada gambar 4a, beda potensial V yang diterapkan dipertahankan oleh baterai. Pada gambar 4b, tiga resistansi paralel telah diganti dengan resistansi yang setara R_{eq} .

Resistansi yang terhubung secara paralel dapat diganti dengan resistansi yang setara, R_{eq} yang memiliki beda potensial yang sama, dan arus total yang sama dengan resistansi aktual.

Untuk mendapatkan ekspresi dari R_{eq} pada gambar 4b, pertama-tama perlu menulis arus pada setiap resistansi aktual pada gambar 4a sebagai

$$i_1 = \frac{V}{R_1}, i_2 = \frac{V}{R_2}, \text{ dan } i_3 = \frac{V}{R_3}$$

di mana V adalah beda potensial antara a dan b. Jika menerapkan aturan arus dalam percabangan pada titik a pada gambar 4a dan kemudian mensubstitusikan nilai-nilai arus itu, memperoleh

$$i = i_1 + i_2 + i_3 = V \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \right) \quad (3)$$

Jika mengganti kombinasi paralel dengan resistansi, R_{eq} , yang setara (gambar 4b), perlu melakukan berikut

$$i = \frac{V}{R_{eq}} \quad (4)$$

Selanjutnya, membandingkan Persamaan.(1) dan (2) memperoleh

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \quad (5)$$

Hasil persamaan (5) dalam kasus resistensi secara umum dapat ditulis dengan persamaan sebagai berikut

$$\frac{1}{R_{eq}} = \sum_{j=1}^n \frac{1}{R_j} \text{ (n resistensi dalam paralel)} \quad (6)$$

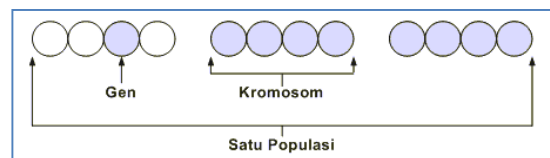
Untuk kasus dua resistensi, resistansi yang setara adalah hasil kali dari dua resistensi (R_1 dan R_2) dibagi dengan jumlah kedua resistensi itu. Secara matematis dituliskan dengan persamaan

$$R_{eq} = \frac{R_1 R_2}{R_1 + R_2} \quad (7)$$

Dalam penelitian ini, persamaan (7) inilah yang disebut sebagai nilai resistansi yang rusak dalam rangkaian elektronik.

D. Algoritma Genetika

Algoritma genetika sebagai cabang Algoritma Evolusi adalah metode adaptive yang digunakan untuk keperluan pencarian data dan optimasi. Algoritma genetika ini didasarkan atas proses genetik makhluk hidup, yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau "Siapa yang kuat, dia yang bertahan (*survive*)".



Gambar 5. Gen, kromosom dan populasi

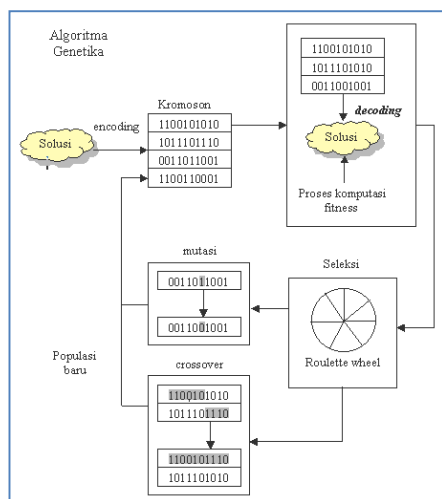
Sebelum Algoritma Genetika dipergunakan sebagai pemecahan masalah, maka sebuah kode yang sesuai untuk persoalan harus dirancang terlebih dahulu. Berkaitan dengan perancangan kode yang sesuai maka titik solusi dalam ruang permasalahan

dikodekan dalam bentuk kromosom; yang terdiri atas komponen genetik terkecil yaitu gen (seperti ditunjukkan pada gambar 5).

Konsep dasar algoritma genetika ini adalah pencarian secara acak, dengan setiap nilai dinyatakan sebagai individu. Untuk memeriksa hasil optimasinya, dibutuhkan sebuah fungsi yang disebut sebagai fungsi fitness. Fungsi fitness ini menggambarkan suatu hasil yang sudahdikodekan.

Selama proses algoritma genetika ini berjalan, induk digunakan untuk reproduksi, sedangkan Rekombinasi dan mutasi digunakan untuk menciptakan keturunan. Dan jika desain algoritma genetika tercipta dengan baik, maka populasi yang tercipta akan mengalami konvergensi dan akan memberikan hasil yang optimum.

Setiap algoritma genetika beroperasi pada populasi kromosom buatan. Populasi kromosom buatan adalah string dalam alfabet terbatas, biasanya berupa string biner [12][14][16][17][10]. Setiap kromosom mewakili solusi untuk masalah dan memiliki kesesuaian, bilangan real yang merupakan ukuran seberapa baik solusi untuk masalah yang dihadapi.



Gambar 6. Mekanisme algoritma genetika.

Dalam pelaksanaannya algoritma genetikamenerapkan operasi diantaranya: **Seleksi**: proses pemilihan individu dari populasi berdasarkan nilai fitnessnya. Salah satu cara untuk proses seleksi ini adalah mempergunakan mesin *Roulette*, **Rekombinasi** (*crossover*): teknik pertukaran gen untuk memperoleh individu

baru. Salah satu teknik dasar untuk Rekombinasi adalah mempergunakan pertukaan gen, **mutasi**: suatu teknik inversi gen atau pergeseran gen. Salah satu teknik dasar untuk mutasi adalah mempergunakan mutasi acak, dan **offspring**: individu baru, dibentuk dengan cara melakukan rekombinasi antar kromosom-kromosom dalam satu generasi.

Secara umum mekanisme algoritma genetika ini seperti yang ditunjukkan pada gambar 6.

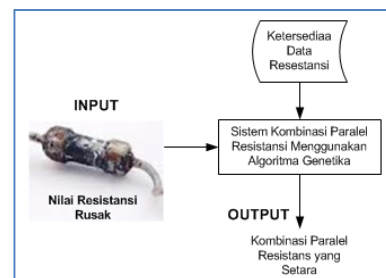
E. Black Box Testing

Setiap *source code* program hasil dari implementasi suatu algoritma sangat perlu dilakukan pengujian. Hal ini dimaksudkan untuk memastikan bahwa semua proses yang ada di dalam program itu sudah dapat berjalan sesuai dengan fungsionalitas yang diharapkan.

Salah satu jenis perangkat yang dapat dipergunakan untuk menguji kode program adalah *Black Box Testing*; yang berfokus pada spesifikasi fungsional dari program. *Tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program [18].

METODE

Penelitian yang dilaksanakan merupakan jenis penelitian simulasi. Penelitian simulasi ini bertujuan untuk menyelesaikan masalah pencarian kombinasi paralel resistansi yang memiliki nilai hitung yang setara dengan nilai dari sebuah resistansi yang dinyatakan rusak.



Gambar 7. Mekanisme dalam memperoleh kombinasi paralel resistansi

Nilai resistensi rusak, dalam penelitian merupakan masukan (*input*) bagi sistem. Seperti yang ditunjukkan pada gambar 7, masukan tersebut selanjutnya, di proses dengan mempergunakan metode algoritma

genetika. Di sini peran algoritma genetika adalah sebagai model pencari solusi secara heuristik. Hasil pemrosesan data masukan oleh algoritma genetika akan memberikan sebuah luaran (*output*). Luaran yang diberikan oleh sistem merupakan solusi optimum bagi permasalahan berupa kombinasi paralel resistansi.






Untuk memperoleh sebuah solusi yang optimal dari permasalahan kombinasi paralel resistansi dengan teknik simulasi ini melalui beberapa tahapan. Adapun tahapan dalam penelitian simulasi ini diantaranya: (1) menetapkan data resistansi rusak, (2) menetapkan ketersediaan data resistansi, (3) inialisasi algoritma genetika, (4) operator algoritma genetika, pelaksanaan implementasi, dan *black box testing* [18].

A. Penetapan Ketersediaan Resistansi

Data ketersediaan resistansi dalam penelitian simulasi pencarian kombinasi paralel resistansi ini merupakan data awal yang penting bagi algoritma genetika. Karena data ketersediaan nilai resistansi dari resistor ini berguna untuk memperoleh data yang disebut sebagai populasi awal. Selain daripada itu data tersebut dipergunakan sebagai kontrol pembuktian dari usulan pemakaian metode algoritma genetika bisa atau tidak bisanya sebagai salah satu algoritma alternatif guna memperoleh kombinasi paralel nilai resistansi.

Simulasi pada penelitian ini memakai data resistansi sebanyak lima buah dengan nilai resistansi berbeda. Adapun kelima data resistansi tersebut seperti ditunjukkan pada tabel 2.

Tabel 2. Daftar resistansi yang tersedia

No.	Resistor	Resistansi
1		10 Ohm
2		20 Ohm
3		30 Ohm
4		40 Ohm
5		50 Ohm

Pada penelitian ini, banyaknya data ketersediaan resistansi dan nilai resistansi tiap resistor disimpan dalam bentuk file

dengan ekstensi txt. Nama file tersebut adalah dataR.txt.

B. Penetapan Resistansi Rusak

Data resistansi rusak merupakan data input bagi sistem. Atas dasar resistansi rusak ini sistem akan melakukan pencarian kesetaraan resistansi melalui persamaan (7) berdasarkan pada ketersediaan nilai resistansi.

Adapun data resistansi rusak yang dipergunakan pada penelitian ini seperti yang ditunjukkan pada tabel 3.

Tabel 3. Daftar resistansi rusak

No.	Resistansi Rusak (untuk simulasi)
1	12.0 Ohm
2	25.0 Ohm

C. Inialisasi Algoritma Genetika

Inialisasi algoritma genetika adalah kegiatan menterjemahkan permasalahan ke dalam sistem alamiah. Hasil inialisasi algoritma genetika ini merupakan model dari individu dan pernyataan *fitness* untuk solusi masalah yang ditangan.

Pada penelitian ini, model individu dan *fitness* didefinisikan secara berurutan sebagai berikut:

Definisi individu

Individu adalah satu pasang resistansi yang tersusun secara paralel. Satu pasang resistansi ini merupakan ilustrasi kombinasi dari resistansi sebagai solusi masalah.



Gambar 8. Definisi individu solusi

Definisi fitness

Berdasarkan konsep dasar susunan resistansi paralel maka dapat definisi nilai *fitness* untuk masalah ini dinyatakan sebagai nilai inversi dari selisih besarnya nilai kombinasi resistansi paralel (R_{eq}) dengan besarnya resistansi rusak (R_{rsk}).

Secara matematis nilai *fitness* untuk penelitian ini dapat dinyatakan dengan persamaan berikut

$$Fitness = \frac{1}{|R_{rsk} - R_{eq}| + 1} \quad (8)$$

D. Operator Algoritma Genetika

Operator-operator yang diterapkan pada algoritma genetika dalam penelitian ini adalah sebagai berikut: (1) Seleksi, (2) Rekombinasi, (3) Mutasi gen.

Seleksi

Proses memilih individu dari populasi berdasarkan nilai fitnessnya. Pada proses ini, individu-individu yang baik dipilih secara acak. Teknik pemilihan individu memakai metode mesin mesin *roulette*.

Rekombinasi

Proses memperoleh individu baru. Pada proses ini menerapkan aritmatika rekombinasi (*crossover*) [19]. Dengan nilai gen-gen yang baru merupakan hasil proses aritmatika rekombinasi.

Implementasi aritmatika rekombinasi pada penelitian ini seperti yang ditunjukkan gambar 8. Untuk simulasi nilai parameter probabilitas Rekombinasi (*probCO*) yang akan digunakan adalah 70%, 85%, dan 95%.

```
void GAResistor::CrossOver(float probCO){
    int i,j,m;
    float p;

    for(i=0;i<Npop;i++){
        anak[i]=induk[i];
    }

    m=int(Npop/2);

    for(i=0;i<m;i++){
        p=(float)rand()/RAND_MAX;
        if(p<probCO){
            j=2*rand()/RAND_MAX;
            anak[2*i].gen[j]=induk[2*i+1].gen[j];
            anak[2*i+1].gen[j]=induk[2*i].gen[j];
            anak[2*i].R[j]=r[anak[2*i].gen[j]];
            anak[2*i+1].R[j]=r[anak[2*i+1].gen[j]];
        }
    }
}
```

Gambar 8. Fragmen program Rekombinasi.

Mutasi

Pada proses mutasi dilakukan dengan cara mengganti satu gen yang terpilih secara acak dengan suatu nilai baru yang diperoleh secara acak pula.

```
void GAResistor::Mutasi(float probMut){
    int i,j;
    float p;

    for(i=0;i<Npop;i++){
        p=(float)rand()/RAND_MAX;
        if(p<probMut){
            j=2*rand()/RAND_MAX;
            anak[i].gen[j]=n*rand()/RAND_MAX;
            anak[i].R[j]=r[anak[i].gen[j]];
        }
    }
}
```

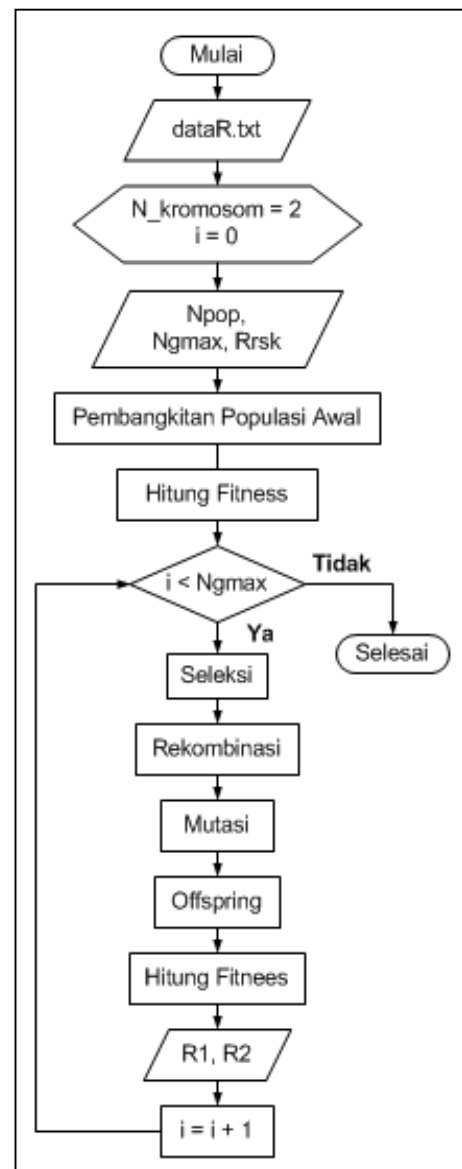
Gambar 9. Fragmen program mutasi

Jumlah individu yang mengalami mutasi dalam satu populasi akan ditentukan oleh parameter probabilitas mutasi (*probMut*).

Implementasi aritmatika Rekombinasi pada penelitian ini seperti yang ditunjukkan gambar 9. Untuk simulasi nilai parameter probabilitas mutasi yang akan digunakan 10%.

E. Algoritma Pencarian Kombinasi Paralel Resistansi

Secara umum, algoritma yang dipakai guna pencarian kombinasi paralel resistansi detailnya seperti ditunjukkan pada gambar 10.



Gambar 10. Flowchart pencarian kombinasi paralel resistansi yang setara dengan resistansi rusak.

F. Implementasi Algoritma Pencarian Kombinasi Paralel Resistansi

Hasil algoritma pencarian kombinasi paralel resistansi yang diperoleh tersebut selanjutnya diimplementasikan memakai bahasa pemrograman C++ pada software Borland C++ 5.02.

HASIL DAN PEMBAHASAN

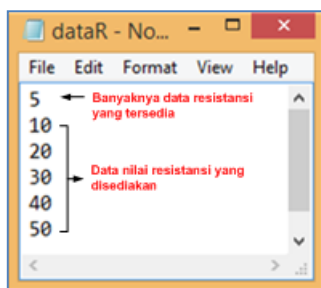
Dalam bagian ini akan menjelaskan tentang hasil simulasi yang telah diperoleh yang di mulai dengan baca data resistansi (asumsi) yang tersedia pada jasa service barang elektronik hingga memberikan hasil luaran berupa kombinasi paralel resistansi.

A. Ketersediaan data resistansi

Gambar 9, menjelaskan bahwa data ketersediaan resistansi untuk simulasi sebanyak lima buah dengan nilai resistansi 10, 20, 30, 40, dan 50. Data ketesediaan resistansi tersebut disimpan pada sebuah file dengan namadataR.txt.

Pada file *dataR.txt*, baris pertama dipergunakan untuk menyatakan banyaknya resistansi yang tersedia, sedangkan untuk baris kedua dan seterusnya dipergunakan untuk menyatakan resistansi dari resistor.

Banyaknya data resistansi yang ditulis harus sama banyak dengan angka yang tertulis pada baris pertama file *dataR.txt*. Bila banyaknya data resistansi lebih kecil dari angka maka resistansi yang lain akan bernilai 0.



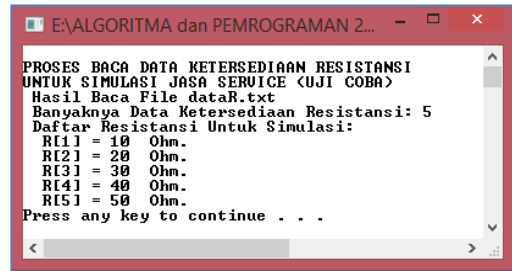
Gambar 11.Data di dalam file dataR.txt.

B. Uji proses baca ketersediaan resistansi

Pengujian baca data ketersediaan resistansi dimaksudkan untuk menyakinkan bahwa fungsi *BacaData()* dapat berjalan sesuai dengan yang diharapkan.

Gambar 12 menunjukkan hasil luaran yang dihasilkan ketika fungsi *BacaData()* dijalankan. Data resistansi yang ditampilkan

sudah sesuai dengan data resistensi di file dataR.txt.

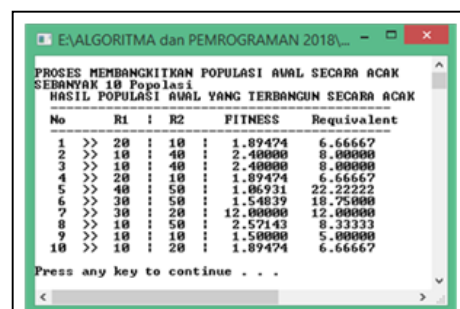


Gambar 12.Hasil baca data ketersediaan resistansi untuk simulasi.

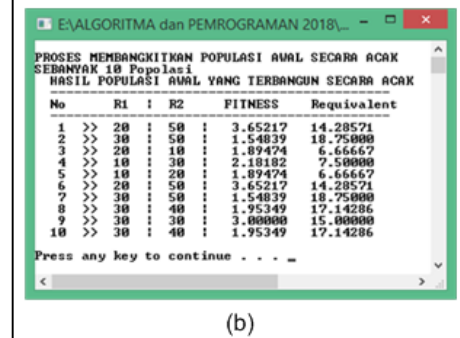
Dengan dapat terbacanya data ketersediaan resistansi ini maka data tersebut dapat dipakai untuk menguji fungsi untuk pembangkitan populasi secara acak, yaitu fungsi *BangkitkanPopulasiAwal()*.

C. Uji proses pembangkitan populasi awal

Gambar 13 menjelaskan tentang hasil luaran fungsi *BangkitkanPopulasiAwal()* setelah dijalankan. Fungsi tersebut adalah fungsi yang bertugas untuk membangkitkan populasi dari sejumlah data ketersediaan resistansi. Fungsi pembangkit populasi ini diberi tugas untuk menciptakan populasi sebanyak 10 individu dari data ketersediaan resistansi yang ditunjukkan di gambar 12.



(a)



(b)

Gambar 13.Hasil pembangkitan populasi awal. (a) pertama, dan (b) kedua.

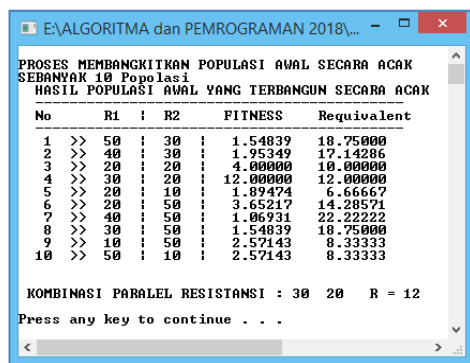
Gambar 13a merupakan hasil luaran dari fungsi pembangkit populasi saat dijalankan untuk yang pertama guna membangun populasi awal, sedangkan gambar 13b merupakan hasil luaran dari fungsi pembangkit populasi saat dijalankan untuk yang kedua kalinya untuk membangun populasi awal.

Hasil populasi yang dibangun oleh fungsi pembangkit populasi awal dijalankan pertama dengan yang kedua memberikan hasil luaran yang berbeda. Hal ini memberi arti bahwa fungsi tersebut telah bekerja sesuai yang diharapkan, yaitu setiap kali fungsi pembangkit populasi awal ini dijalankan harus menciptakan populasi dengan individu-individu baru dan sudah sesuai dengan peran bilangan acak, dimana setiap kali bilangan acak itu dijalankan maka akan memberikan angka-angka yang baru.

D. Uji Proses Cari Individu Maksimum

Pada bagian ini adalah untuk menguji fungsi *CariIndividuMaks()* dalam menemukan individu maksimum pada satu populasi. Individu maksimum ini dapat dikatakan sebagai solusi dari masalah kombinasi paralel resistansi. Selain itu, kemunculan individu maksimum pada populasi awal dapat menimbulkan pencarian terjebak pada lokal optimum.

Gambar 14 menjelaskan hasil luaran dari individu maksimum yang diperoleh dari populasi awal yaitu 30 dan 20. Angka-angka tersebut merupakan gambaran dari kombinasi paralel resistansi dan menunjukkan solusi dari permasalahan. Pemilihan angka-angkaitu dengan dasar nilai fitness pada tiap individu dan dipilih individu dengan nilai fitness terbesar, yaitu 12.



No	R1	R2	FITNESS	Requivalent
1	>> 50	: 30	1.54839	18.75000
2	>> 40	: 30	1.95349	17.14286
3	>> 20	: 20	4.00000	10.00000
4	>> 30	: 20	12.00000	12.00000
5	>> 20	: 10	1.89474	6.66667
6	>> 20	: 50	3.65217	14.28571
7	>> 40	: 50	1.06931	22.22222
8	>> 30	: 50	1.54839	18.75000
9	>> 10	: 50	2.57143	8.33333
10	>> 50	: 10	2.57143	8.33333

KOMBINASI PARALEL RESISTANSI : 30 20 R = 12
Press any key to continue . . .

Gambar 14. Individu maksimum pada populasi awal.

Berdasarkan hasil pengujian fungsi-fungsi untuk keperluan algoritma genetika tersebut dapat diambil simpulan sementara bahwa populasi awal yang terbangun sudah dapat digunakan untuk menguji algoritma genetika dalam membangun kombinasi paralel resistansi yang setara dengan resistansi rusak (input).

E. Kombinasi Paralel Nilai Resistansi Resistor Menggunakan Algoritma Genetika

Untuk melihat kemampuan algoritma genetika dalam menemukan kombinasi paralel nilai resistansi resistor yang setara atau tepat sama dengan nilai resistansi rusak akan diuji menggunakan beberapa parameter.

Adapun parameter-parameter yang akan digunakan untuk menguji sistem pembuat algoritma seperti ditunjukkan pada tabel 4.

Tabel 4. Parameter algoritma genetika

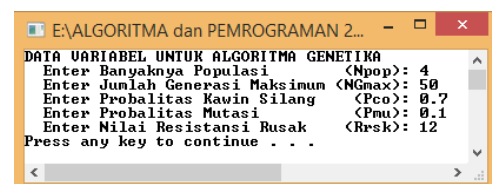
Para	Npop	NGmax	Pco	Pmu	Rrsk
1	4	50	0.70	0.1	12
2	4	50	0.85	0.1	12
3	4	50	0.95	0.1	12
4	4	50	0.70	0.1	25
5	4	50	0.85	0.1	25
6	4	50	0.95	0.1	25
7	10	50	0.95	0.1	12
8	10	50	0.95	0.1	25

Keterangan:

- Para : Parameter ke
- Npop : Banyaknya populasi.
- NGmax: Jumlah generasi maksimum.
- Pco : Probabilitas Rekombinasi.
- Pmu : Probabilitas mutasi.
- Rrsk : Resistansi rusak.

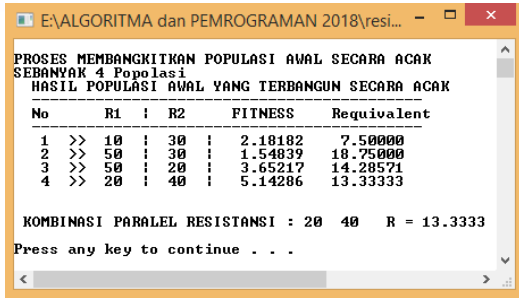
E.1. Parameter ke 1

Masukan (input) data untuk sistem pencarian kombinasi paralel resistansi berdasarkan ketersediaan data resistansi dengan menggunakan algoritma genetika pada resistansi rusak 12 ohm seperti yang ditunjukkan pada gambar 15.



DATA VARIABEL UNTUK ALGORITMA GENETIKA
Enter Banyaknya Populasi (Npop): 4
Enter Jumlah Generasi Maksimum (NGmax): 50
Enter Probabilitas Kawin Silang (Pco): 0.7
Enter Probabilitas Mutasi (Pmu): 0.1
Enter Nilai Resistansi Rusak (Rrsk): 12
Press any key to continue . . .

Gambar 15. Data masukan untuk simulasi parameter ke 1.



No	R1	R2	FITNESS	Requivalent
1	10	30	2.18182	7.50000
2	50	30	1.54839	18.75000
3	50	20	3.65217	14.28571
4	20	40	5.14286	13.33333

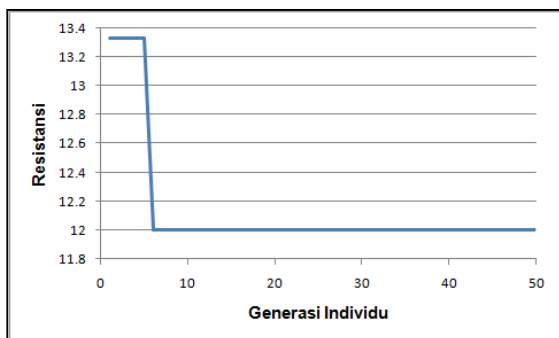
Gambar 16. Populasi dan individu maksimum pada parameter 1.

Dalam gambar 15 menjelaskan bahwa penerapan algoritma genetika mempergunakan 4 individu dalam satu populasi, iterasi guna membentuk individu baru sebanyak 50 kali, 10% untuk probabilitas mutasi, dan 70% untuk probabilitas Rekombinasi serta data resistansi rusak untuk simulasi sebesar 12 Ω .

Masukan data pada gambar 15 tersebut melalui proses bangkitkan populasi awal menghasilkan 4 individu dalam satu populasi dengan individu maksimumnya kombinasi paralel resistansi 20 Ω dan 40 Ω dengan nilai fitness 5.14286 dan resistansi yang dihasilkan adalah 13.3333 Ω seperti yang ditunjukkan pada gambar 16.

Selanjutnya, data individu yang telah diperoleh tersebut (gambar 16) menjadi masukan bagi algoritma genetika untuk mendapatkan kombinasi paralel resistansi melalui generasi.

Gambar 17 merupakan hasil generasi algoritma genetika. Individu baru hasil dari proses generasi adalah kombinasi paralel resistansi 20 Ω dan 30 Ω dengan resistansi sebesar 12 Ω dan sudah setaradengan resistansi rusak pada simulasi. Kombinasi paralel resistansi tersebut diperoleh pada generasi ke 5 (lima).



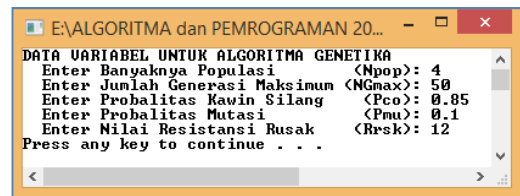
Gambar 17. Nilai resistansi resistor hasil generasi individu pada parameter ke 1.

E.2. Parameter ke 2

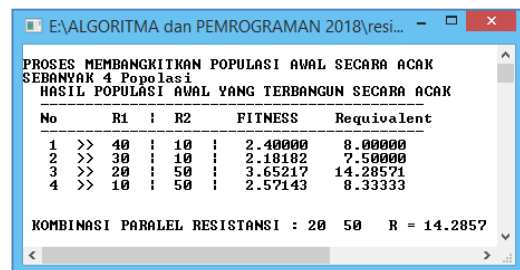
Pada parameter ke 2 ini, masukan untuk sistem seperti yang ditunjukkan pada gambar 18. Data itu sesuai dengan yang tertulis pada tabel 4.

Data tersebut menghasilkan populasi awal dengan empat individu dan individu maksimumnya sebagai kombinasi paralel resistansi 40 Ω dan 20 Ω seperti yang ditunjukkan pada gambar 19.

Selanjutnya, data populasi awal ini menjadi masukan bagi algoritma genetika untuk membangun individu baru melalui generasi. Hasil generasi dan perolehan individu maksimum sebagai solusi pada parameter ke 2 diperoleh pada generasi ke 3 seperti ditunjukkan pada gambar 20 dengan kombinasi paralel resistansi 20 Ω dan 30 Ω . Kombinasi paralel resistansi yang dihasilkan ini sama dengan yang dihasilkan oleh parameter ke 1.

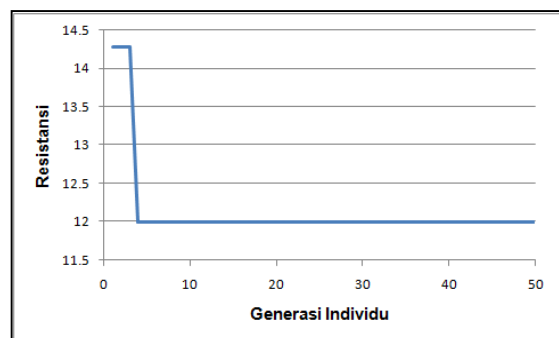


Gambar 18. Data masukan simulasi pada parameter ke 2.



No	R1	R2	FITNESS	Requivalent
1	40	10	2.40000	8.00000
2	30	10	2.18182	7.50000
3	20	50	3.65217	14.28571
4	10	50	2.57143	8.33333

Gambar 19. Populasi awal hasil simulasi pada parameter ke 2.



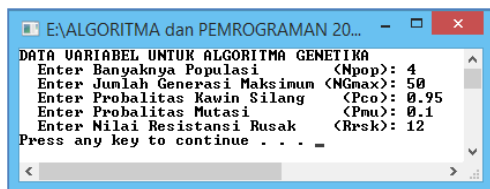
Gambar 20. Nilai resistansi resistor hasil generasi individu. pada parameter ke 2.

E.3. Parameter ke 3

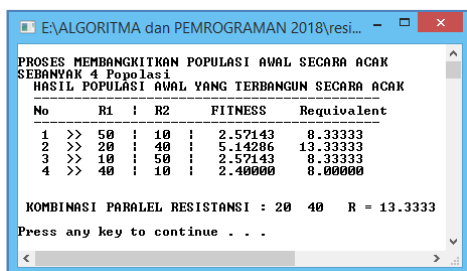
Data masukan untuk parameter ke 3 mempergunakan data seperti ditunjukkan pada gambar 21.

Data tersebut menghasilkan populasi awal dengan empat individu dan individu maksimumnya sebagai kombinasi paralel resistansi 20 Ω dan 40 Ω dengan resistansi sebesar 13.3333 Ω seperti yang ditunjukkan pada gambar 22.

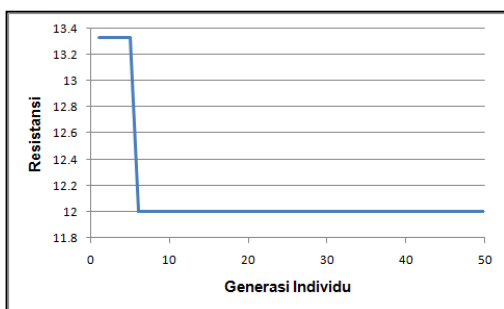
Selanjutnya, data populasi awal ini menjadi masukan bagi algoritma genetika untuk membangun individu baru melalui generasi. Hasil dari proses generasi ini menunjukkan bahwa individu maksimum yang diperoleh sebagai solusi parameter ke 3 adalah pada generasi ke 3 seperti yang ditunjukkan pada gambar 20 dengan kombinasi paralel resistansi 20 Ω dan 30 Ω yang luga merupakan hasil kombinasi paralel resistansi yang sama dengan pada parameter ke 1 maupun parameter ke 2.



Gambar 21. Data masukan simulasi pada parameter ke 3.



Gambar 22. Populasi awal hasil simulasi pada parameter ke 3.



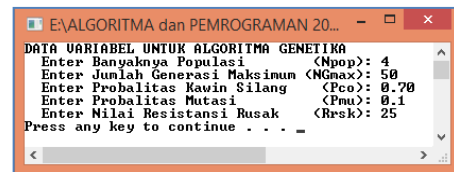
Gambar 23. Nilai resistansi resistor hasil generasi individu pada parameter ke 3.

E.4. Parameter ke 4

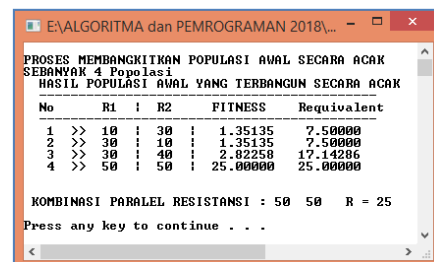
Data masukan untuk parameter ke 4 mempergunakan data seperti ditunjukkan pada gambar 24.

Data tersebut menghasilkan populasi awal dengan empat individu dan individu maksimumnya sebagai kombinasi paralel resistansi 50 Ω dan 50 Ω dengan resistansi sebesar 25 Ω seperti yang ditunjukkan pada gambar 25.

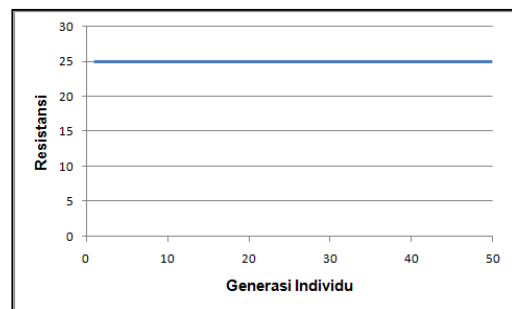
Pada kondisi ini maka pencarian akan terjebak pada optimasi lokal. Hal ini bisa dilihat pada gambar 26 yang menunjukkan individu baru yang dibangun oleh algoritma genetika adalah tetap mulai dari generasi ke 1 hingga ke 50 yaitu kombinasi paralel resistansi 50 Ω dan 50 Ω yang diperoleh sejak generasi pertama.



Gambar 24. Data masukan simulasi pada parameter ke 4.



Gambar 25. Populasi awal hasil simulasi pada parameter ke 4.



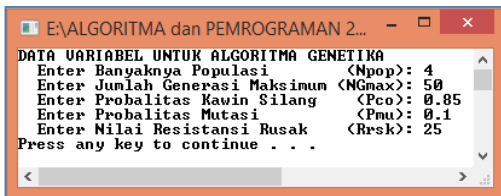
Gambar 26. Nilai resistansi resistor hasil generasi individu pada parameter ke 4.

E.5. Parameter ke 5

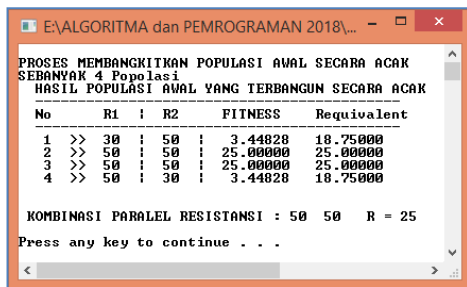
Walaupun hanya berbeda pada angka probabilitasnya dengan parameter ke 4, parameter ke 5 dengan probabilitas

Rekombinasi seperti ditunjukkan pada gambar 27 telah menghasilkan populasi awal empat individu dan sudah memperoleh individu maksimum sebagai kombinasi paralel resistansi 50 Ω dan 50 Ω dengan resistansi sebesar 25 Ω seperti yang ditunjukkan pada gambar 28.

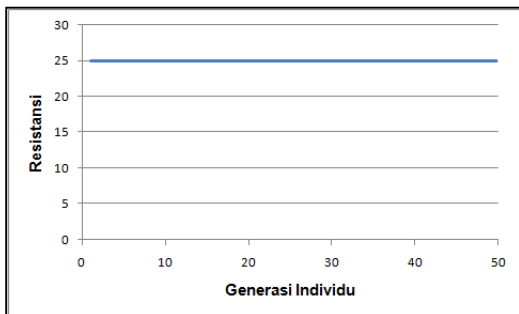
Oleh karena resistansi pada individu maksimum adalah setara dengan resistansi rusak maka pencarian kombinasi paralel resistansi ini terjebak pada optimasi lokal seperti hasil uji parameter ke 4 -gambar 29.



Gambar 27.Data masukan simulasi pada parameter ke 5.



Gambar 28.Populasi awal hasil simulasi pada parameter ke 5.

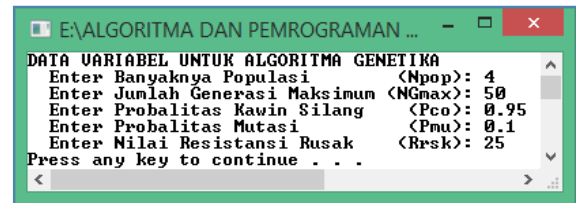


Gambar 29.Nilai resistansi resistor hasil generasi individu pada parameter ke 5.

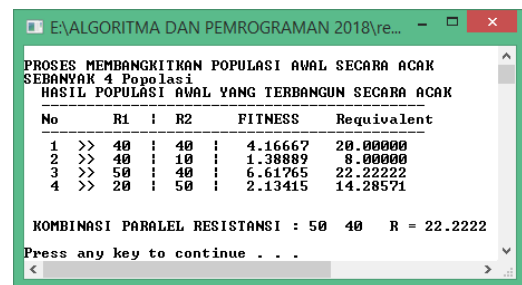
E.6. Parameter ke 6

Perubahan probabilitas Rekombinasi pada parameter ke 6 seperti ditunjukkan gambar 30 telah menghasilkan populasi awal empat individu dan individu maksimum sebagai kombinasi paralel resistansi 50 Ω dan 40 Ω dengan resistansi sebesar 22.22 Ω seperti yang ditunjukkan gambar 31.

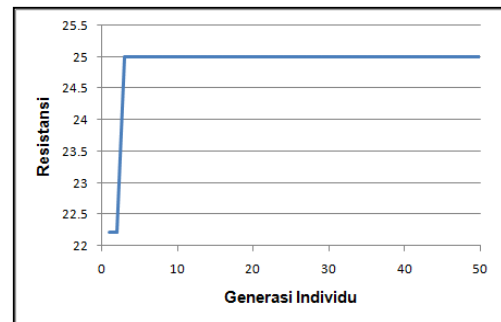
Setelah dilakukan proses generasi menggunakan algoritma genetika diperoleh individu maksimum sebagai kombinasi paralel resistansi 50 Ω dan 50 Ω dengan resistansi 25 Ω pada generasi ke 3 seperti yang ditunjukkan gambar 32.



Gambar 30.Data masukan simulasi pada parameter ke 6.



Gambar 31.Populasi awal hasil simulasi pada parameter ke 6.

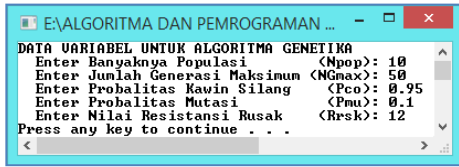


Gambar 32.Nilai resistansi resistor hasil generasi individu pada parameter ke 6.

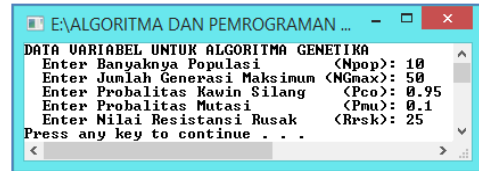
E.7. Parameter ke 7

Pada parameter ke 7 ini pencarian kombinasi paralel resistansi menggunakan populasi yang terbangun atas 10 individu, dengan mempergunakan probabilitas Rekombinasi sebesar 95% untuk memperoleh kombinasi paralel resistansi sebesar 12 Ω seperti ditunjukkan oleh gambar 33.

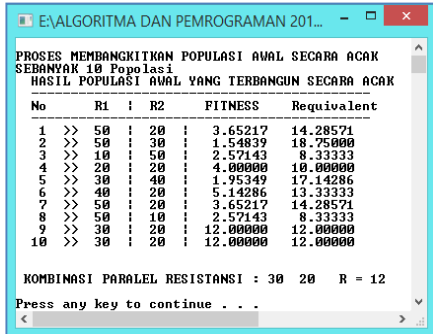
Hasil pembentukan populasi awal ini menghasilkan individu maksimum sebagai kombinasi paralel resistansi 30 Ω dan 20 Ω dengan resistansi sebesar 12 Ω (lihat pada gambar 34).



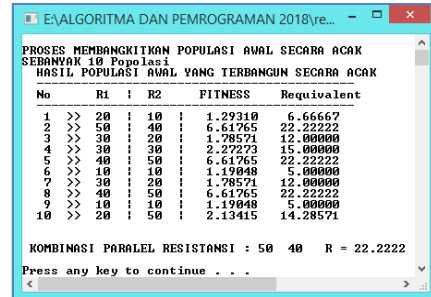
Gambar 33. Data masukan simulasi pada parameter ke 7.



Gambar 36. Data masukan simulasi pada parameter ke 8.

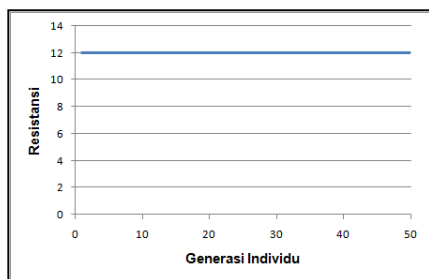


Gambar 34. Populasi awal hasil simulasi pada parameter ke 7.



Gambar 37. Populasi awal hasil simulasi pada parameter ke 8.

Nilai resistansi yang diperoleh pada populasi awal tersebut sesuai dengan nilai resistansi rusak (resistansi pada simulasi). Dengan kondisi demikian, proses pencarian kombinasi paralel resistansi melalui individu maksimum pada setiap generasi dapat dipastikan terjebak dalam optimasi lokal, yakni dari generasi ke 1 sampai ke 50 tidak berubah individu maksimumnya seperti ditunjukkan gambar 35.



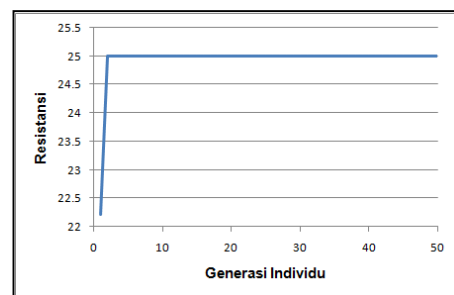
Gambar 35. Nilai resistansi resistor hasil generasi individu pada parameter ke 7.

E.8. Parameter ke 8

Pada parameter ke 8 tidak jauh beda dengan yang dilakukan oleh parameter ke 7, yaitu melakukan pencarian kombinasi paralel resistansi menggunakan populasi yang terbangun atas 10 individu, dengan mempergunakan probabilitas Rekombinasi sebesar 95%. Pada parameter ke 8 ini digunakan untuk memperoleh kombinasi paralel resistansi sebesar 25 Ω seperti ditunjukkan oleh gambar 36.

Kesepuluh individu hasil pembangkitan populasi awal dan individu maksimum sebagai kombinasi paralel resistansi 50 Ω dan 40 Ω dengan resistansi sebesar 22.222 Ω seperti ditunjukkan dalam gambar 37.

Selanjutnya, populasi awal menjadi masukan bagi algoritma genetika untuk melakukan generasi (pembentukan individu baru). Pada proses generasi ini diperoleh individu maksimum sebagai kombinasi paralel resistansi 50 Ω dan 50 Ω dengan resistansi sebesar 25 Ω pada generasi ke 2 seperti ditunjukkan dalam gambar 37



Gambar 38. Nilai resistansi resistor hasil generasi individu pada parameter ke 8.

Secara umum, pengujian algoritma pencarian kombinasi paralel resistansi berdasarkan ketersediaan resistansi menggunakan algoritma genetika dapat dinyatakan sebagai algoritma alternatif untuk bisa memperoleh kombinasi paralel resistansi yang setara atau sama dengan resistansi rusak.

Tabel 5. Hasil Kombinasi Paralel Resistansi

Parameter ke	Kombinasi Paralel		Generasi ke
	Nilai Resistansi Resistor		
	R1	R2	
1	20	30	5
2	20	30	3
3	20	30	5
4	50	50	1
5	50	50	1
6	50	50	3
7	30	20	1
8	50	50	2

Hal ini dapat dilihat pada tabel 5 yang menunjukkan bahwa masalah resistansi rusak yang dipakai untuk simulasi pada parameter ke 1 hingga parameter ke 8 kombinasi paralel resistansi yang dihasilkan sangat baik.

SIMPULAN

Berdasarkan hasil simulasi yang telah dilaksanakan maka dapat diambil simpulan:

1. Algoritma genetika dapat dimanfaatkan untuk mendapatkan kombinasi paralel nilai resistansi resistor yang setara atau pun sama dengan nilai resistansi dari resistor yang rusak berdasarkan data ketersediaan nilai resistansi resistor.
2. Penggunaan probabilitas mutasi 10% dan probabilitas rekombinasi (crossover): 70%, 85%, dan 95% dengan individu sebanyak 4 maupun 10 bisa memperoleh kombinasi paralel nilai resistansi resistor yang setara ataupun sama dengan nilai resistansi dari resistor yang rusak.
3. Bila pada populasi awal sudah diperoleh kombinasi paralel nilai resistansi resistor yang setara ataupun sama dengan nilai resistansi resistor rusak pencarian terjebak pada lokal optimal.
4. Data ketersediaan nilai resistansi resistor memiliki pengaruh terhadap perolehan ketepatan kombinasi paralel resistansi yang setara dengan resistansi rusak.

REFERENSI

- [1] Ana Sofiana, Ian Yulianti dan Sujarwata. (2017). Identifikasi Nilai Hambat Jenis Arang Tempurung Kelapa dan Arang

Kayu Mangrove sebagai Bahan Alternatif Pengganti Resistor Film Karbon. Unnes Physics Journal. 2017; 6(1): 1-6.

- [2] Maisyaroh, Ridwan Septianto. (2016). Sistem Pakar Dianogsis Kerusakan Pada Televisi (TV) Tabung Menggunakan Metode Forward Chaining. Jurnal Techno Nusa Mandiri, 2016; 13(2): 52-62.
- [3] Endri Maulana, Rachmat Adi Purnama. (2017). Pemanfaatan Layanan SMS Telepon Seluler Berbasis Mikrokontroler Atmega328p Sebagai Sistem Kontrol Lampu Rumah. Jurnal Teknik Komputer AMIK BSI, 2017; 3(1): 93-99.
- [4] Halliday, Resnick, Jearl Walker. (2011). Fundamentals of physics. 9th Edition. John Wiley&Sons, Inc ISBN 978-0-470-46908-8.
- [5] Toibah Umi Kalsum, Rosdiana. (2011). Alat Penghapus Whiteboard Otomatis Menggunakan Motor Stepper. Jurnal Media Infotama, 2011; 7(1): 38-56
- [6] Sopian. (2018). Tabel Kode Warna Rresistor dan CaraMenghitung Nilainya. <https://www.sopian.id/2018/08/tabel-kode-warna-resistor-dan-cara.html>. Diakses tanggal 10 Maret 2019. Jam 14.10 WIB.
- [7] Suswono. (2016). Cara Mengukur dan Mengetahui Kerusakan Komponen Resistor. <http://suswono28.blogspot.com/2016/03/cara-mengukur-dan-mengetahui-kerusakan.html>. Diakses pada tanggal 10 Maret 2019 pukul 13.48 WIB.
- [8] Toibah Umi Kalsum, Rosdiana. (2011). Alat Penghapus Whiteboard Otomatis Menggunakan Motor Stepper. Jurnal Media Infotama, 20011; 7(1): 38-56
- [9] Untoro, Wisnu Yudo, F.X. (2006). Deteksi Lokasi Gangguan Listrik Pada Sistem Jaringan Distribusi Listrik Berdasarkan Laporan Gangguan Pelanggan Mengguna-kan Algoritma Genetika. Jurnal Matematika dan Komputer Indoesia, 2006; 1(1): 7-14.
- [10] Untoro, Wisnu Yudo, F.X. (2010). Algoritma & Pemrograman dengan Bahasa Java. Edisi Pertama. Yogyakarta. Graha Ilmu. 2010: 11-27.
- [11] Basuki, Achmad. (2003). Algoritma Genetika Suatu Alternatif Penyelesaian

- Permasalahan Searching, Optimasi dan Machine Learning. <http://basuki.lecturer.pens.ac.id/lecture/AlgoritmaGenetika.pdf>. Diakses pada tanggal 10 Maret 2019 pukul 14.45 WIB.
- [12] Basuki, Achmad. (2003). Strategi Menggunakan Algoritma Genetika. <http://basuki.lecturer.pens.ac.id/lecture/StrategiAlgoritmaGenetika.pdf>. Diakses pada tanggal 10 Maret 2019 pukul 14.55 WIB.
- [13] Mahmudy, Wayan Firdaus. (2008). Optimasi Fungsi Tanpa Kendala Menggunakan Algoritma Genetika Dengan Kromosom Biner dan Perbaikan Kromosom Hill-Climbing. *Kursor*, 2008; 4(1): 23-29.
- [14] Randy L. Haupt and Sue Ellen Haupt. (2004). Chapter 2: The Binary Genetic Algorithm. *Practical Genetic Algorithms*, Second Edition, 2004. John Wiley & Sons, Inc. ISBN 0-471-45565-2. 27-49.
- [15] Kreyzig, Erwin: alihbahasa, Bambang Sumantri. (1993). *Matematika Teknik Lanjutan Buku 2*. Edisi ke 6. Gramedia Pustaka Utama, Jakarta. Hal: 548-552.
- [16] Nitasha Soni, Dr Tapas Kumar; Nitasha Soni et al . (2014). Study of Various Mutation Operators in Genetic Algorithms. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 2014; Vol. 5 (3) : 4519-.
- [17] John McCall . (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 2005; 184 (2005): 205–222.
- [18] Mustaqbal M. Sidi, Roeri Fajri Firdaus, Hendra Rahmadi. (2015). PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 2015; 1(3): 31-36.
- [19] Rohaeni Lani, Deni Saepudin, Aniq Atiqi Rohmawati. (2016). Penerapan Algoritma Genetika Untuk Optimasi Debit Air Pada Pembangkit Listrik Tenaga Air, Studi Kasus : Waduk Situ Cileunca, Jawa Barat. *e-Proceeding of Engineering*, 2016; .3(2): 3964-3971.
- [20] Ruri Hartika Zain, S. Adhista Ricky Yatra. (2012). Aplikasi Pagar Elektrik Pada Keamanan Fasilitas Lembaga Perumahan Dilengkapi Alarm Deteksi Pemutusan Arus Listrik Dan Sensor Menggunakan Jaringan Komputer. *Jurnal Momentum*, 2012: 13(2): 81-97.