

Implementasi Algoritma Genetika pada Mikrokontroler MCS51 Untuk Mencari Rute Terpendek

Thiang, Ronald Kurniawan, Hany Ferdinando

Jurusan Teknik Elektro, Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya

Telp: (031) 8439040, 8494830-31 ext. 1354, 1361 Fax: (031) 8436418

Email: thiang@petra.ac.id, hanyf@petra.ac.id

Abstrak

Dalam makalah ini akan dipaparkan tentang implementasi algoritma genetika dengan menggunakan mikrokontroler MCS51. Aplikasi yang dipilih untuk percobaan adalah mencari rute terpendek yang menghubungkan suatu tempat asal dan tujuan. Semua proses algoritma genetika seperti membangkitkan populasi awal, evaluasi, seleksi, crossover dan mutasi, dilakukan oleh mikrokontroler MCS51. Jenis kromosom yang digunakan dalam sistem ini adalah bit string. Metode seleksi yang digunakan adalah roulette wheel dan elitism. Sebuah program bantuan yang beroperasi di komputer, digunakan untuk membuat peta jalan, men-download ke mikrokontroler MCS51 semua informasi yang dibutuhkan.

Pengujian telah dilakukan dengan simulasi pada program dan dengan menerapkan pada mikrokontroler untuk menentukan rute terpendek dari suatu lokasi menuju ke lokasi lain. Dari hasil pengujian terlihat bahwa implementasi algoritma genetika pada mikrokontroler dapat berjalan dengan baik dan mampu menentukan rute terpendek yang dapat ditempuh dengan bervariasi model peta.

KATA KUNCI: algoritma genetika, mikrokontroler MCS51

1. Pendahuluan

Saat ini perkembangan algoritma genetika demikian pesat. Sebagian besar implementasi algoritma genetika dilakukan pada komputer dengan menggunakan bahasa pemrograman tingkat tinggi seperti pascal, C++ dan lain-lain. Bagaimana bila algoritma genetika diimplementasikan pada mikrokontroler?

Karena itu dalam makalah ini akan dipaparkan penelitian atau percobaan yang telah dilakukan yaitu implementasi algoritma genetika pada mikrokontroler. Dalam hal ini dipilih mikrokontroler MCS51 karena mikrokontroler ini sangat populer di Indonesia. Semua proses algoritma genetika seperti crossover, mutasi, membangkitkan generasi awal, seleksi dilakukan oleh mikrokontroler.

Karena algoritma genetika bersifat stokastik atau bekerja berdasarkan random, maka ini merupakan tantangan tersendiri dalam mengimplementasikan algoritma genetika pada mikrokontroler. Hal ini disebabkan karena keterbatasan instruksi-instruksi dalam bahasa pemrograman assembly untuk mikrokontroler. Karena itu, sangat diharapkan, penelitian atau percobaan ini dapat memberikan kontribusi dalam perkembangan ilmu algoritma genetika.

2. Algoritma Genetika

Algoritma genetika adalah suatu algoritma pencarian yang bersifat stokastik, berbasis pada mekanisme dari seleksi alam dan genetika. Algoritma genetika sangat tepat digunakan untuk penyelesaian

masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode yang konvensional.

Suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator [1] yaitu operator reproduksi, operator crossover dan operator mutasi. Struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

1. **Membangkitkan populasi awal**
Populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang merepresentasikan solusi yang diinginkan.
2. **Membentuk generasi baru**
Dalam membentuk digunakan tiga operator yang telah disebut di atas yaitu operator reproduksi/seleksi, crossover dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru.
3. **Evaluasi solusi**
Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai fitness setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti yang sering digunakan antara lain:
 - Berhenti pada generasi tertentu.
 - Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness tertinggi tidak berubah.

- Berhenti bila dalam n generasi berikut tidak didapatkan nilai fitness yang lebih tinggi.

Sebelum algoritma genetika dilakukan, ada dua hal penting yang harus dilakukan yaitu pendefinisian kromosom dan fungsi fitness atau fungsi obyektif. Dua hal ini juga memegang peranan penting dalam algoritma genetika untuk menyelesaikan suatu masalah.

3. Implementasi pada Mikrokontroler

Dalam penelitian ini, algoritma genetika diimplementasikan pada mikrokontroler 89C51 yang merupakan salah satu keluarga mikrokontroler MCS51.

Beberapa prosedur standar telah dikembangkan untuk menunjang implementasi algoritma genetika pada mikrokontroler khususnya mikrokontroler MCS51. Prosedur tersebut adalah:

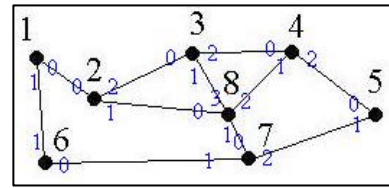
- Prosedur *Random_8* dan *Random_16*
 Prosedur ini disediakan untuk membangkitkan bilangan random. Jenis bilangan random yang dihasilkan adalah *pseudo random*. Prosedur *random_8* berfungsi untuk membangkitkan bilangan random 8 bit dan prosedur *random_16* untuk membangkitkan bilangan random 16 bit.
- Prosedur *Create_1st_Generation*
 Prosedur ini disediakan untuk membangkitkan populasi awal.
- Prosedur *Crossover*
 Prosedur ini disediakan melakukan proses crossover dari dua kromosom dan menghasilkan dua kromosom baru. Jenis crossover yang digunakan adalah one-cut point crossover.
- Prosedur *Mutation*
 Prosedur ini disediakan untuk melakukan proses mutasi. Jenis mutasi yang digunakan adalah mutasi bit.
- Prosedur *Roulette_Wheel_Selection*
 Prosedur ini disediakan untuk melakukan proses seleksi dengan menggunakan metode roulette wheel.
- Prosedur *Elitism_Selection*
 Prosedur ini disediakan untuk melakukan proses seleksi dengan menggunakan metode elitism dimana satu kromosom terbaik akan terpilih.

Beberapa spesifikasi yang perlu diperhatikan adalah jumlah kromosom per populasi maksimum 5 kromosom, jenis kromosom adalah binary bit string dan panjang maksimumnya 16 bit

4. Aplikasi Mencari Rute Terpendek

Untuk mencoba dan menguji sistem maka diambil aplikasi sederhana yaitu mencari rute terpendek dari suatu lokasi ke lokasi lain. Dalam aplikasi ini didisain juga suatu program yang berjalan di PC dengan tujuan untuk membuat peta jalan dan mendownload ke mikrokontroler semua informasi yang dibutuhkan

antara lain informasi peta jalan beserta jaraknya, titik asal dan titik tujuan. Gambar 1 menunjukkan contoh peta jalan yang telah didisain.



Gambar 1. Contoh peta jalan yang diimplementasikan

Titik-titik pada peta menunjukkan lokasi-lokasi yang ada. Setiap titik mempunyai cabang-cabang yang menghubungkan titik tersebut dengan titik yang lain. Jarak antar titik telah didefinisikan terdahulu. Dalam sistem ini dibatasi setiap titik maksimum mempunyai 4 cabang. Setiap cabang dalam satu titik akan diberi nomor. Sebagai contoh pada peta dalam gambar 1, titik 1 mempunyai 2 cabang dan diberi nomor 0 dan 1. Cabang 0 menghubungkan titik 1 dengan titik 2 sedangkan cabang 1 menghubungkan titik 1 dengan titik 6.

4.1. Representasi Kromosom

Jenis kromosom yang digunakan adalah binary bit string dengan jumlah bit bergantung pada:

- Jumlah titik.
 - Jumlah percabangan terbanyak untuk satu titik.
- Panjang kromosom (dalam bit) dapat dihitung dengan menggunakan persamaan berikut:

$$P = C \times (T - 1) \quad (1)$$

Dimana C adalah jumlah bit binary dari jumlah percabangan terbanyak dan T adalah jumlah titik. Pada gambar 1 ada 8 titik dan jumlah percabangan terbanyak adalah 4 cabang (membutuhkan 2 bit binary) maka dengan menggunakan persamaan 1 didapatkan panjang kromosomnya 14 bit.

Setiap kromosom mengandung informasi rute dari satu titik ke titik yang lain. Bit-bit dalam kromosom menunjukkan nomor cabang pada titik-titik yang ada. Bila nomor cabang tidak ada maka dilakukan perulangan. Untuk lebih jelas lihat contoh berikut yang menggunakan peta seperti pada gambar 1.

Start awal = titik 1
 Kromosom = 01 00 11 10 00 11 10
 Rute = 1 - 6 - 7 - 8 - 4 - 3 - 2 - 3

4.2. Fungsi Fitness

Dalam sistem ini, permasalahan optimasi adalah mencari rute terpendek. Karena itu fungsi fitness yang digunakan dapat didefinisikan sebagai jarak total dari rute yang ada. Fungsi fitness tersebut dapat direpresentasikan oleh persamaan berikut:

$$F = \sum_{Rute} \text{Jarak antartitik} \quad (2)$$

4.3. Operator Algoritma Genetika

Dalam aplikasi ini metode seleksi yang digunakan adalah roulette wheel dan elitism. Dalam seleksi roulette wheel, peluang terpilihnya suatu kromosom sebanding dengan besar nilai fitness kromosom tersebut. Karena permasalahan optimasi mencari rute terpendek (minimum), maka pada seleksi roulette wheel, nilai fitness masing-masing kromosom diubah dengan menggunakan persamaan berikut:

$$F_n' = F_{\max} + F_{\min} - F_n \quad (3)$$

Dalam aplikasi ini, jenis crossover yang digunakan adalah one-cut point crossover sedangkan jenis mutasi yang mutasi bit.

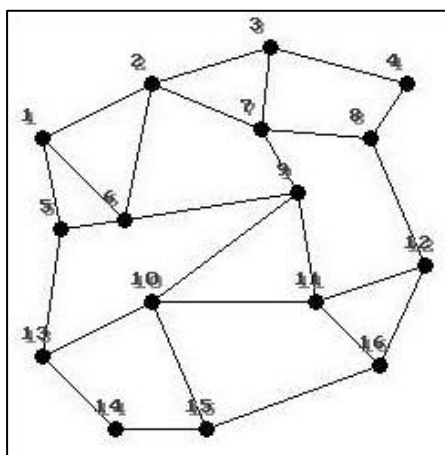
4.4. Evaluasi dan Kriteria Berhenti

Proses evaluasi dilakukan dengan menghitung nilai fitness dari setiap kromosom dalam suatu generasi. Bila ada kromosom yang tidak mempunyai informasi titik tujuan maka kromosom ini dianggap mempunyai nilai fitness terbesar dan dikatakan tidak valid.

Dalam aplikasi ini, proses algoritma genetika akan berhenti bila dalam 500 generasi ke depan tidak ada kromosom yang mempunyai nilai fitness yang lebih baik.

4.5. Hasil Pengujian

Pengujian telah dilakukan dengan beberapa macam peta, beberapa kombinasi metode seleksi dan juga beberapa variasi nilai crossover rate dan mutation rate. Berikut adalah beberapa hasil pengujian dengan peta seperti pada gambar 2.



Gambar 2. Peta Jalan yang Diujikan

Tabel 1. Hasil Pengujian Rute Titik 1 ke Titik 16

No.	Metode Seleksi	Pc	Pm	Waktu (detik)	Fitness Terbaik	Pada Generasi
1	E	0.9	0.01	22.5	254	128
2	E	0.8	0.05	18.95	254	38
3	E	0.6	0.1	18.29	254	12
4	E	0.4	0.2	20.33	254	70
5	RW	0.9	0.01	25.71	254	45
6	RW	0.8	0.05	27.63	254	94
7	RW	0.6	0.1	29.58	254	131
8	RW	0.4	0.2	24.54	254	17
9	Both	0.9	0.01	27.65	254	95
10	Both	0.8	0.05	23.41	254	7
11	Both	0.6	0.1	25.45	254	53
12	Both	0.4	0.2	34.72	254	247

Tabel 2. Hasil Pengujian Rute Titik 3 ke Titik 14

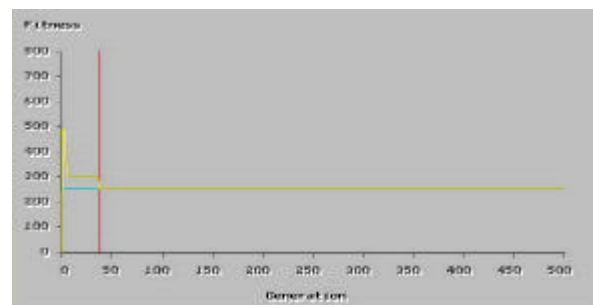
No.	Metode Seleksi	Pc	Pm	Waktu (detik)	Fitness Terbaik	Pada Generasi
1	E	0.9	0.01	22.56	298	132
2	E	0.8	0.05	19.64	298	40
3	E	0.6	0.1	17.69	298	10
4	E	0.4	0.2	17.99	298	9
5	RW	0.9	0.01	23.77	298	6
6	RW	0.8	0.05	23.3	298	2
7	RW	0.6	0.1	24.48	298	25
8	RW	0.4	0.2	25.89	298	48
9	Both	0.9	0.01	22.9	298	1
10	Both	0.8	0.05	22.94	298	3
11	Both	0.6	0.1	23.17	298	7
12	Both	0.4	0.2	25.7	298	62

Keterangan:

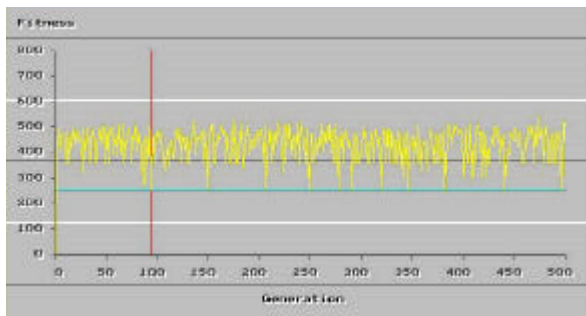
E : Elitism

RW : Roulette Wheel

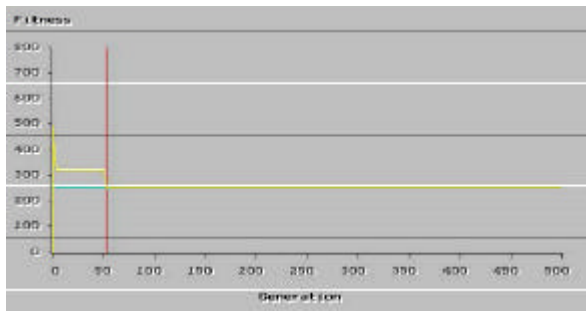
Both: Elitism dan Roulette Wheel



Gambar 3. Grafik Fungsi Fitness (Elitism)



Gambar 4. Grafik Fungsi Fitness (Roulette Whell)



Gambar 5. Grafik Fungsi Fitness (Roulette Whell)

Dari tabel 1 dan 2 terlihat bahwa crossover rate yang terbaik antara 0,6 sampai 0,8 dan mutation rate terbaik antara 0,05 sampai 0,1. gambar 3, 4 dan 5 menunjukkan perjalanan fungsi fitness untuk masing-masing metode seleksi dan kombinasinya. Terlihat bahwa dalam metode seleksi elitism, fungsi fitness terbaik generasi sekarang selalu lebih baik dari generasi sebelumnya sedangkan metode seleksi roulette wheel lebih bersifat random.

5. Kesimpulan

Dari percobaan yang telah dilakukan dapat diambil kesimpulan bahwa implementasi algoritma genetika pada mikrokontroler sudah dapat berjalan dengan baik walaupun dengan beberapa batasan antara lain jumlah kromosom per populasi, jenis kromosom binary bit string. Pengembangan selanjutnya dapat dilakukan dengan menambah memori sehingga batasan-batasan tadi dapat diatasi.

6. Daftar Pustaka

- [1] Davis, Lawrence. "Handbook Of Genetic Algorithms". New York: Van Nostrand Reinhold,1991.
- [2] Gen, Mitsuo,and Cheng, Runwei "Genetic Algorithms And Engineering Design". Edited by Hamid R.Parsaei. United States Of America: John Wiley and Sons,1997.
- [3] Goldberg, David Edward. "Genetic Algorithms in Search, Optimization, and Machine Learning". United States of America: Addison-Westley,1989.

- [4] Man,K.F, et al. "Genetic Algorithms For Control And Signal Processing". London: Springer,1997.
- [5] Advanced Micro Device, "Microcontrollers Handbook", California : Advanced Micro Device, 1988.