

211-6  
Pen

LAPORAN PENELITIAN

MILIK PERPUSTAKAAN  
UNIMED



PEMBUATAN PROGRAM KOMPUTER UNTUK OPTIMALISASI  
FILTER DIGITAL MENGGUNAKAN ALGORITMA GENETIK

Oleh :

- 1. Alkhafi Maas Siregar, S.Si, M. Si (Ketua Peneliti)
- 2. Drs. Abu Bakar (Anggota)
- 3. Drs. Juniar Hutahaean (Anggota)
- 4. Drs. Khairul Amdani, M. Si (Anggota)
- 5. Dra. Betty M. Turnip, M.Pd (Anggota)

DIBIYAI DANA RUTIN UNIVERSITAS NEGERI MEDAN  
TAHUN ANGGARAN 2005  
SK REKTOR UNIMED NOMOR : 01444A/J39.10/LK/2005  
TANGGAL : 24 AGUSTUS 2005

TGL TERIMA	
NOMOR	
REVISI	
NO. INDOX	
	07/012

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI MEDAN  
NOPEMBER, 2005

**HALAMAN PENGESAHAN  
LAPORAN PENELITIAN DANA RUTIN**

1. a. Judul penelitian : Pembuatan Program Komputer untuk Optimalisasi Pembelajaran Digital Menggunakan Algoritma Genetik
- b. Bidang Ilmu : MIPA
- c. Kategori Penelitian : I
2. Ketua Peneliti  
a. Nama Lengkap dan Gelar : Alkhafi Maas Siregar, S.Si, M.Si  
b. Jenis Kelamin : Laki-laki  
c. Golongan, Pangkat dan NIP : III-b, Penata Muda Tk. 1 / 132125659  
d. Jabatan Fungsional : Asisten Ahli  
e. Fakultas/ Jurusan : MIPA/Fisika  
f. Pusat Penelitian : Lembaga Penelitian UNIMED
3. Alamat Ketua Peneliti  
a. Alamat Kantor : Jl. Willem Iskandar Psr. V Medan  
b. Alamat Rumah : Jl. H.M.Yamin Gg.Lurah No. 13 Medan
4. Jumlah Anggota Peneliti : 4 orang  
a. Nama Anggota Peneliti I : Drs. Abu Bakar  
b. Nama Anggota Peneliti II : Drs. Juniar Hutahaean, M.Si  
c. Nama Anggota Peneliti III : Drs. Khairul Amdani, M. Si  
d. Nama Anggota Peneliti IV : Dra. Betty M. Turnip, M. Pd
5. Lokasi Penelitian : Laboratorium Fisika FMIPA UNIMED
6. Lama Penelitian : 8 bulan
7. Biaya yang Diperlukan  
a. Sumber Dana Rutin : Rp. 3.000.000,-  
(Tiga Juta Rupiah)

Mengetahui  
Dekan Fakultas MIPA UNIMED

Medan, Nopember 2005  
Ketua Peneliti,

Prof. Drs. M. Situmorang, M.Sc., Ph.D  
NIP. 131572430

Alkhafi Maas Siregar, S.Si, M.Si  
NIP. 132125659

Menyetujui,  
Ketua Lembaga Penelitian UNIMED

(Prof. Dr. Abdul Muin Sibuea, M.Pd)  
NIP. 130935473

## RINGKASAN

### PEMBUATAN PROGRAM KOMPUTER UNTUK OPTIMALISASI FILTER DIGITAL MENGUNAKAN ALGORITMA GENETIK

Telah dilakukan diimplementasikan algoritma genetik dalam optimalisasi filter digital menggunakan bahasa pemrograman MATLAB. Filter digital yang digunakan adalah Lowpass Filter, Highpass Filter, Bandpass Filter, dan Bandstop Filter. Program yang dirancang terlebih dahulu diuji menggunakan suatu fungsi yang memiliki banyak minimum lokal dan maksimum lokal. Hasil pengujian pada fungsi uji menunjukkan bahwa algoritma yang diterapkan telah bekerja dengan baik.

Hasil pengujian program pada berbagai filter yang digunakan menunjukkan bahwa program dapat diimplementasikan. Filter yang dihasilkan telah memperlihatkan karakter sebagaimana layaknya sebuah filter.

Kata kunci: algoritma genetik, filter

COMPUTER PROGRAM FOR DIGITAL FILTER OPTIMALIZATION  
USING GENETIC ALGORITHM

The implementation of genetic algorithm for optimal digital filter with MATLAB language program has been carried out. The digital filter are Low pass Filter, High pass Filter, Band pass Filter, and Band stop Filter. The program was tested using a function that has many local minimum and maximum. Testing result show that the algorithm was well done work.

Testing result, various filters reveal that the program was well implemented. The filter result show has a character as a filter.

Key words: genetic algorithm, filter



## DAFTAR ISI

Halaman Pengesahan	Halaman
Ringkasan	i
Summary	ii
Daftar isi	iii
Daftar Gambar	iv
	v
<b>BAB I</b>	
<b>PENDAHULUAN</b>	
<b>A. LATAR BELAKANG</b>	1
<b>B. PERUMUSAN MASALAH</b>	2
<b>BAB II</b>	
<b>TINJAUAN PUSTAKA</b>	3
<b>A. TINJAUAN UMUM FILTER DIGITAL</b>	3
<b>B. ALGORITMA GENETIK</b>	5
<b>C. OPERATOR GENETIK</b>	6
<b>D. BAHASA PEMROGRAMAN MATLAB</b>	7
<b>BAB III</b>	
<b>TUJUAN DAN MANFAAT PENELITIAN</b>	11
<b>A. TUJUAN YANG INGIN DICAPAI</b>	11
<b>B. MANFAAT PENELITIAN</b>	11
<b>BAB IV</b>	
<b>METODE PENELITIAN</b>	12
<b>A. TEMPAT DAN WAKTU PENELITIAN</b>	12
<b>B. ALAT DAN BAHAN</b>	12
<b>C. RANCANGAN PENELITIAN</b>	12
<b>D. PENGUJIAN PROGRAM</b>	12
<b>BAB V</b>	
<b>HASIL DAN PEMBAHASAN</b>	13
<b>A. HASIL</b>	13
<b>1. LISTING PROGRAM DALAM MATLAB</b>	13
<b>2. HASIL EKSEKUSI</b>	24
<b>B. PEMBAHASAN</b>	29
<b>BAB VI</b>	
<b>KESIMPULAN DAN SARAN</b>	30
<b>A. KESIMPULAN</b>	30
<b>B. SARAN</b>	30
<b>DAFTAR PUSTAKA</b>	31
<b>Lampiran</b>	
<b>A. PERSONALIA PENELITIAN</b>	32
<b>B. SURAT PERINTAH KERJA</b>	33

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Filter digital dan pencuplikan sinyal analog (kontinu).	3
Gambar 2.2 Penjumlah, pengali, serta tundaan negatif dan positif.	4
Gambar 2.3 Siklus algoritma genetik (Man, 1997).	5
Gambar 2.4. Bagan alir algoritma genetik.	6
Gambar 5.1 Grafik fungsi untuk pengujian.	24
Gambar 5.2 Sebaran nilai hasil pencarian nilai optimum menggunakan algoritma genetik.	25
Gambar 5.3 Lowpass Filter hasil pencarian algoritma genetik.	27
Gambar 5.4 Highpass Filter hasil pencarian algoritma genetik.	27
Gambar 5.5 Bandpass Filter hasil pencarian algoritma genetik.	28
Gambar 5.6 Bandstop Filter hasil pencarian algoritma genetik.	28





# BAB I

## PENDAHULUAN

### A. LATAR BELAKANG

Pendekatan tradisional untuk perancangan filter menyangkut transformasi dari analog ke digital pada spesifikasi yang ditentukan, secara umum menggunakan transformasi bilinear. Masalahnya adalah hasil yang diperoleh tidak optimal karena transformasi bilinear menggunakan pendekatan fungsi dalam penyelesaiannya. Karena itu optimalisasi filter sebagai komponen utama pemrosesan sinyal menjadi sangat dibutuhkan (Koonin, 1990).

Fungsi perancangan filter yang ada saat ini (Butterworth, Chebyshev tipe I dan II, serta *Elliptic*) pada umumnya didasarkan pada metode pendekatan (*approximation method*) untuk mengimplementasikan selektif frekuensi analog analog pada filter seperti LP, HP, BP, dan BS. Sementara perancangan filter dengan unjuk kerja yang lebih optimal serta memenuhi kriteria jelas amat diperlukan. Secara umum perancangan harus memenuhi tiga kriteria (Widrow, 1985), yaitu:

1. Orde filter rendah
2. Filter harus stabil
3. Memenuhi penetapan toleransi

Pemakaian algoritma genetik untuk pemecahan masalah, khususnya teknik optimasi telah merambah ke berbagai bidang ilmu. Algoritma genetik bekerja dengan prinsip seleksi alami, yakni yang terkuatlah yang akan bertahan hidup (*survival of the fittest*). Rumusan permainannya adalah bahwa yang tertangguhlah yang akan menang (*the fittest will win*). Algoritma genetik bukanlah pemecah masalah yang tertuntun secara matematis (*mathematicallly guided problem solver*), seperti yang terjadi pada tipe gradien tradisional dengan formulasi matematis yang sangat sulit. Algoritma genetik adalah suatu *event* yang memiliki kebebasan dan fleksibilitas intrinsik untuk memilih solusi yang diinginkan menurut spesifikasi rancangan dan mempunyai sifat spesifik tersendiri, yakni: *non linier, stochastic, constrained, discrete, dan multimodal*. Nilai optimal yang diperoleh merupakan produk akhir hasil perkembangan dari generasi ke generasi

dengan individu-individu terbaik dalam jumlah populasi tertentu. Algoritma ini tidak rentan terhadap masalah minimum lokal yang muncul pada permukaan-permukaan *multimodal*.

Algoritma genetik merupakan suatu wujud pencarian random (*stochastic*) yang menirukan prinsip proses evolusi biologi alami guna mencari solusi optimal untuk suatu permasalahan kompleks. Algoritma ini dimulai dengan suatu kumpulan parameter yang disebut kromosom atau *string* dan masing-masing dievaluasi tingkat ketangguhannya (*fitness*) oleh fungsi objektif yang telah ditentukan. Melalui proses seleksi alami atas operator genetik, gen-gen dari dua kromosom (disebut *parent*) diharapkan akan menghasilkan kromosom baru dengan tingkat *fitness* yang lebih tinggi sebagai basis generasi baru atau keturunan (*offspring*) berikutnya. Setiap kromosom atau *string* dalam satu populasi menjadi kandidat solusi terhadap fungsi objektif.

Perancangan filter yang optimal karenanya sangat dimungkinkan untuk direalisasikan dengan formulasi algoritma genetik tanpa pembatasan pada jenis rancangan filter. Setiap jenis filter dapat dengan bebas diperoleh optimasinya hingga orde terendah, serta diperoleh suatu fungsi multiobjektif secara simultan. Jadi, pemecahan masalah parameter untuk filter digital menggunakan persamaan fungsi alih untuk setiap jenis filter dapat dilakukan.

## B. PERUMUSAN MASALAH

Berdasarkan uraian yang telah diberikan pada pendahuluan, maka yang menjadi rumusan masalah dalam penelitian ini adalah

1. Bagaimanakah bentuk rancangan program komputer yang akan digunakan untuk optimalisasi filter digital menggunakan algoritma genetik ?
2. Bagaimanakah mengimplementasikan algoritma genetik untuk optimalisasi filter digital?



## BAB II

### TINJAUAN PUSTAKA

#### A. Tinjauan Umum Filter Digital

Secara umum filter atau tapis digital didefinisikan sebagai suatu prosedur numerik atau algoritma untuk mengubah sejumlah runtun (sinyal) yang diberikan ke dalam bentuk runtun dengan sifat sebagaimana yang diinginkan, seperti mengurangi derau, distorsi, maupun interferensi (Kuc, 1982). Misalnya di dalam penerapan radar, filter digital digunakan untuk memperbaiki pendeteksian pesawat udara. Dalam pemrosesan tutur (*speech*) filter digunakan untuk mengurangi kelebihan dalam sinyal tutur serta memenuhi transmisi yang lebih efisien di samping untuk mengenali tutur itu sendiri.



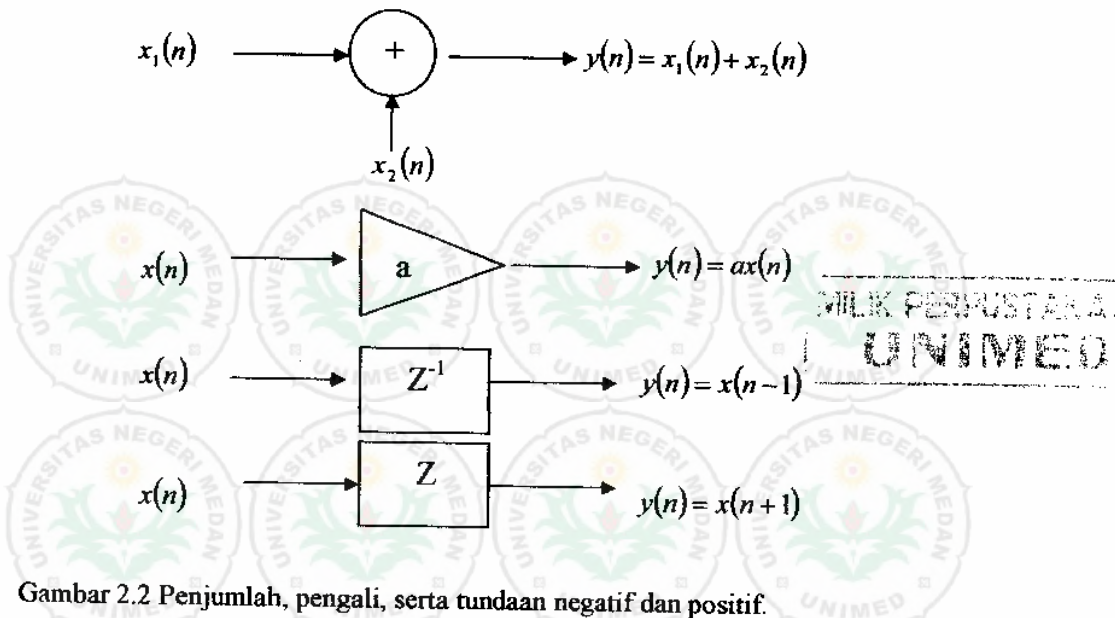
Gambar 2.1 Filter digital dan pencuplikan sinyal analog (kontinu).

Runtun masukan dapat diperoleh dengan berbagai cara diantaranya adalah yang paling umum digunakan seperti pada gambar (2.1), yakni dengan pencuplikan sinyal waktu kontinu pada selang waktu tertentu yang berjarak sama, atau dapat dirumuskan:

$$x(nT_s) = x(t) \Big|_{t=nT_s} \quad (2.1)$$

dengan  $T_s$  merupakan jarak atau periode pencuplikan. Jadi pada hakikatnya filter digital hanya mengolah sinyal dalam bentuk runtun waktu diskrit.

Berdasarkan anatominya filter digital terdiri atas interkoneksi tiga elemen penting yaitu penjumlah (*adders*), pengali (*multipliers*), dan tundaan (*delays*). Penjumlah dan pengali merupakan komponen sederhana yang dapat diimplementasikan terutama di dalam unit aritmatik dan logika suatu alat penghitung. Sedangkan tundaan adalah bagian yang memberikan akses ke nilai yang akan datang atau nilai yang telah lewat atas suatu runtun sinyal. Ketiga komponen inilah yang digunakan untuk menggambarkan suatu diagram filter digital secara utuh dan dalam bentuk pemodelan manapun (Proakis, 1992).



Gambar 2.2 Penjumlah, pengali, serta tundaan negatif dan positif.

Unjuk kerja filter digital dalam kawasan waktu digambarkan dengan faktor runtun tanggapan cuplik satuan (*unit-sample response*) filter tersebut  $[h(n)]$ . Runtun ini merupakan analogi tanggapan impuls pada filter analog. Persamaan konvolusional dapat digunakan untuk menentukan runtun masukan (*input*), luaran (*output*), dan tanggapan (*response*) cuplik satuan itu serta untuk menilai apakah suatu filter dalam keadaan stabil. Disisi lain, persamaan differensial linier juga memberikan alternatif untuk mengimplementasikan struktur filter digital dalam deskripsi kawasan waktu.

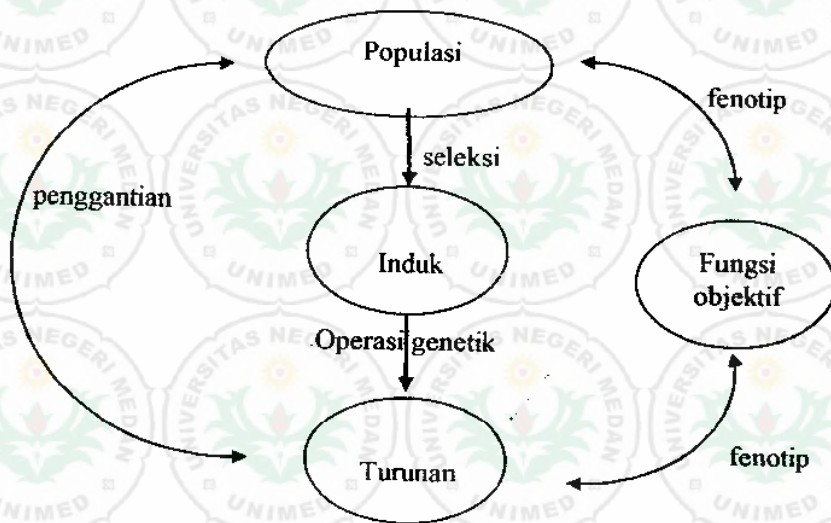
Filter digital dapat dibedakan atas dua spesifikasi ditinjau dari sudut perancangan dan implementasi serta durasi cuplik satuannya, yakni: filter tanggapan impuls takhingga (*Infinite duration Impuls Response, IIR*) dan filter tanggapan impuls berhingga (*Finite duration Impuls Response, FIR*). Khusus untuk kasus filter IIR, pendekatan tradisional untuk perancangannya adalah dengan transformasi dari filter analog, dalam hal ini memakai transformasi bilinier, atau *impulse invariant*. Pada umumnya teknik yang digunakan untuk implementasi filter IIR analog selektif frekuensi (*lowpass, highpass, bandpass, bandstop filter*) adalah dengan metode pendekatan fungsi Butterworth, Chebyshev

tipe I dan II, serta *Ellptic*. Teknik perancangan ini dapat ditemukan dalam MATLAB *toolbox*.

Penggunaan algoritma genetik akan menggantikan nilai-nilai pendekatan, serta struktur filter yang dihasilkan dari perancangan benar-benar memenuhi kriteria yang telah ditetapkan.

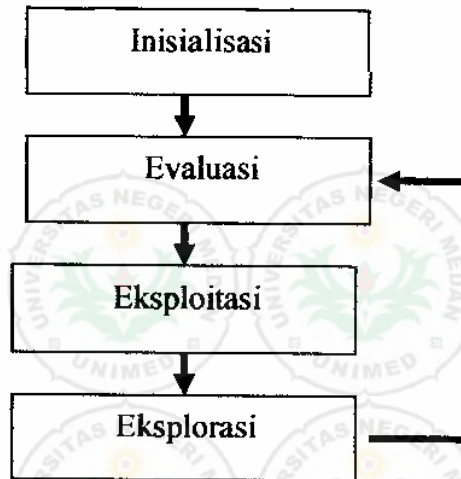
## B. Algoritma Genetik

Algoritma genetik merupakan metode pencarian yang meniru perumpamaan evolusi biologis alami untuk menentukan struktur atau individu berkualitas tinggi dalam suatu *domain* yang disebut populasi. Proses pemilihan suatu individu dari suatu populasi disesuaikan dengan tingkat *fitness* yang ditetapkan berdasarkan fungsi objektif untuk masalah yang dioptimasi. Individu atau yang disebut juga kromosom dengan nilai *fitness* yang lebih tinggi akan terpilih (bertahan hidup) dan menjadi induk atau *mating pool* untuk memperoleh keturunan atau generasi berikutnya. Kromosom tersebut sekaligus menjadi kandidat solusi pada setiap siklus iteratif yang disebut dengan generasi.



Gambar 2.3 Siklus algoritma genetik (Man, 1997).

Proses evolusi algoritma genetik terdiri dari atas empat tahap dasar sebagaimana gambar (2.4) berikut.



Gambar 2.4. Bagan alir algoritma genetik.

### C. Operator Genetik

Operator-operator genetik berguna untuk memperkenalkan kromosom atau string baru dalam populasi. Operator ini terdiri atas operator genetik dasar dan operator genetik hasil modifikasi.

#### 1. Reproduksi

Proses reproduksi terdiri dari dua tahap, yakni proses seleksi dan rekombinasi. Pada proses seleksi, setiap *string* dalam populasi akan diseleksi berdasarkan nilai *fitness*-nya. Probabilitas terpilihnya satu *string* dinyatakan dengan

$$P_i = \frac{f_i}{\sum f_i} \quad (2.2)$$

dengan  $P_i$  adalah peluang string  $i$  terpilih dan  $f_i$  adalah nilai fitness string.

#### 2. Mutasi

Tujuan mutasi adalah untuk mengatasi gangguan terhadap parameter dan menjamin bahwa seluruh ruang lingkup dalam penelitian dapat dicapai. Umumnya, bila peluang yang diperoleh besar, maka konvergensi akan cepat diperoleh. Proses mutasi secara acak mengubah gen dari 0 ke 1 atau dari 1 ke 0 dengan nilai probabilitas ( $P_m$ ) tipikal lebih kecil dari 0,1.

#### 3. Inversi



Inversi berguna untuk membalikkan urutan gen antara dua posisi acak yang dipilih dalam satu kromosom. Hal ini diperlukan untuk pengurutan kembali posisi gen yang secara potensial lebih baik (Goldberg, 1989).

Agar hasil inversi mempunyai informasi seperti induknya, maka diperlukan operator penataan kembali (*reordering*). Operator ini merupakan modifikasi atas operator genetik dasar terutama operator *crossover*. Karena operator ini merupakan operator terpenting pada algoritma genetik dalam menghasilkan *string* baru.

#### 4. Pengkodean kembali nol

Jika nilai ideal suatu parameter adalah negatif dan nilai yang diharapkan positif, ada kemungkinan parameter tersebut akan tetap di sekitar nol sepanjang runtunan generasi. Hal ini dapat diatasi dengan pengkodean kembali nol, yaitu jika bit-bit (karakter) *string* suatu parameter adalah 00...00, dikode kembali sebagai 11...11 atau sebaliknya. Ini diperbolehkan karena 1111...=0000.... dalam salah satu komplemen aritmatik (Yao dan Sethares, 1984).

Operator-operator genetik dalam pengendaliannya memerlukan ukuran parameter. Pemilihan ukuran parameter genetik menentukan kinerja algoritma genetik dalam memecahkan suatu masalah. Ukuran parameter yang sering digunakan terdiri atas ukuran populasi, probabilitas *crossover*, ( $P_C$ ), dan probabilitas mutasi ( $P_M$ ). Ukuran parameter populasi tidak memiliki standar nyata. Pada umumnya semakin besar ukuran populasi, semakin cepat mencapai konvergensi dan akan membutuhkan komputasi lebih banyak dengan waktu yang panjang untuk setiap populasi.

#### D. Bahasa pemrograman MATLAB

Pemilihan bahasa pemrograman MATLAB didasarkan pada kemudahan di dalam mengimplementasikan komponen-komponen Algoritma Genetik (AG) yang banyak menggunakan operasi matriks. Walaupun, jika dibandingkan dengan bahasa C, JAVA, ataupun DELPHI, bahasa pemrograman MATLAB lebih lambat dalam proses komputasi. Hal ini disebabkan banyaknya fungsi ataupun prosedur yang dibuat umum sehingga banyak redundansi.



Sesuai dengan namanya, "MATrix LABoratory", MATLAB adalah suatu bahasa pemrograman untuk komputasi numerik yang berorientasi pada matriks.

### 1. Lingkungan MATLAB

Lingkungan MATLAB terdiri dari **Command Window**, untuk mengetikkan baris-baris perintah, dan **Workspace**, yang menunjukkan variabel-variabel yang sedang aktif pada direktori yang ditunjukkan oleh **Current Directory**.

### 2. Pemrograman MATLAB

Dalam lingkungan MATLAB setiap peubah (*variable*) dipandang sebagai matriks. Peubah tidak perlu dideklarasikan karena sudah ditangani oleh MATLAB. Ukuran sebuah matriks dapat dengan mudah diperbesar maupun diperkecil. Hal ini membuat sebuah peubah menjadi sangat fleksibel. Contohnya: dengan menuliskan  $A=[]$  akan diperoleh sebuah matriks kosong. Bila kemudian bila dituliskan  $A=[1\ 3]$  akan diperoleh matriks  $[1\ 3]$ , sedangkan bila kemudian dituliskan  $A=[A;A]$  akan diperoleh sebuah matriks  $2 \times 2$ .

Pembuatan sebuah program atau fungsi dilakukan di dalam suatu *file* berekstensi *.m*. *File* ini dapat dieksekusi melalui *Command Window* dengan cara mengetikkan nama file atau fungsi terkait. Agar dapat dieksekusi penamaan *file* harus sesuai dengan nama fungsinya. Tempat untuk mengetikkan program yang akan dieksekusi berupa *editor*.

MATLAB membedakan huruf besar dan huruf kecil (*case sensitive*). Artinya, peubah **A** dan **a** dikenali sebagai dua buah peubah yang berbeda. Di dalam MATLAB, *i* dan *j* menunjukkan bagian imajiner dari suatu bilangan kompleks. Karena itu untuk menghindari kesalahan semantik, sebaiknya tidak menggunakan *i* dan *j* sebagai nama sebuah peubah. Sebagai alternatif, mungkin dapat digunakan *ii* dan *jj*.

### 3. Konsep Penting dan Perintah-Perintah MATLAB

#### a. Matriks

Matriks adalah kumpulan angka berbentuk persegi. Dalam MATLAB, setiap peubah dipandang sebagai matriks. Oleh karena itu sebuah bilangan

skalar dinyatakan dalam matriks  $1 \times 1$ . Sedangkan vektor dinyatakan oleh matriks baris atau matriks kolom. Tanda `' >>'`, menyatakan *Command Prompt* pada *Command Window*. Untuk membuat sebuah matriks  $3 \times 3$ , misalnya, dapat dituliskan `>>A=[2 3 5; 4 1 7;6 3 9]`; .Dimana tanda `' ;'`, di dalam kurung siku menyatakan pemisah antar baris, sedangkan tanda `' ;'`, pada akhir perintah menyatakan bahwa peubah A tidak ditampilkan di *Command Window*. Jika tanda `' ;'`, pada akhir perintah dihilangkan, maka isi peubah A akan ditampilkan di *Command Window*.

b. Indeks matriks

Penunjukan elemen matriks dapat dilakukan berdasarkan indeksinya. Penunjukan dilakukan dengan menggunakan aturan **A(baris,kolom)**. Kedua indeks, dalam hal ini baris dan kolom, adalah bilangan bulat positif. Misalnya dibangkitkan sebuah matriks A dengan:

```
>>A=[2 3 5; 4 1 7;6 3 9]
```

```
A =
```

```

2     3     5
4     1     7
6     3     9
```

Untuk menjumlahkan semua elemen matriks pada kolom pertama dan disimpan pada peubah X, dituliskan:

```
>>X=A(1,1)+A(2,1)+A(3,1)
```

```
X =
```

```
12
```

Kesalahan yang sering terjadi adalah indeks bukan bilangan bulat positif ataupun indeks d luar jangkauan seperti berikut ini:

```
>>C=A(1,0)
```

```
??? Subscript indices must either be real positive integers or logicals
```

```
>>X=D(4,1)
```

```
??? Index exceeds matrix dimensions
```

c. Operator

Operator yang banyak digunakan adalah ':'. Operator ini sangat penting dalam pemrograman MATLAB, karena banyak sekali perintah yang menggunakan operator ini. Misalnya diinginkan memperoleh bilangan 1 s.d.

5. Menggunakan operator : dapat dituliskan:

```
>>F=1:5
```

F =

1 2 3 4 5

Bila diinginkan sederetan bilangan bulat dengan kelipatan -5 dapat dituliskan:

```
>>B=30:-5:1
```

B =

30 25 20 15 10 5

Menggunakan perintah:

```
>>C=B(1:3)
```

C =

30 25 20

```
>>K=B(2:4)
```

K =

25 20 15

Yakni mendapat elemen vektor B, kolom 1 sampai kolom 3. Ataupun perintah:

```
>>D=A(:,1:2)
```

D =

2 3  
4 1  
6 3

Yakni menampilkan seluruh elemen matriks pada kolom 1 dan kolom 2.

#### d. Concatenation

Penggabungan beberapa matriks kecil menjadi sebuah matriks besar disebut dengan *Concatenation*. Dalam MATLAB, operasi ini dilakukan dengan sangat sederhana.

```
>>G=[ C K]
```

30 25 20 25 20 15

## BAB III

### TUJUAN DAN MANFAAT PENELITIAN

#### A. TUJUAN YANG INGIN DICAPAI

Adapun tujuan penelitian ini adalah

1. Mengimplementasikan algoritma genetik untuk optimalisasi filter digital.
2. Merancang program komputer dengan bahasa pemrograman MATLAB untuk optimalisasi filter digital menggunakan algoritma genetik.

#### B. MANFAAT PENELITIAN

Penelitian ini diharapkan mempunyai manfaat sebagai berikut :

1. Tersedianya program komputer yang mempunyai kemampuan untuk optimalisasi filter digital menggunakan algoritma genetik.
2. Program komputer yang dihasilkan dapat digunakan sebagai alternatif dalam perancangan filter digital.

## BAB IV METODE PENELITIAN

### A. TEMPAT DAN WAKTU PENELITIAN

Penelitian ini akan dilakukan di Laboratorium Fisika FMIPA UNIMED selama 8 (delapan) bulan.

### B. ALAT DAN BAHAN

Materi penelitian ini adalah data frekuensi untuk filter yang dioptimasi. Alat yang dipergunakan adalah perangkat keras berupa personal komputer (PC) dengan spesifikasi Prosesor Intel P IV 2,4 G Hz dan RAM 128 MB, serta Perangkat lunak Matlab 5,3 untuk mengimplementasikan metode yang digunakan.

### C. RANCANGAN PENELITIAN

Penelitian ini dilakukan setelah melakukan studi pustaka terhadap jurnal dan buku untuk mendapatkan pemahaman metode. Adapun langkah-langkah yang ditempuh adalah :

- a. Menetapkan persamaan matematika yang akan terlibat di dalam program.
- b. Membuat algoritma yang akan diimplementasikan dalam wujud program.
- c. Mengimplementasikan persamaan matematika di dalam bentuk program.
- d. Menguji program yang telah dibuat.

### D. PENGUJIAN PROGRAM

Program diuji menggunakan data frekuensi beberapa filter melalui pendekatan, dan dibandingkan hasilnya dengan data yang diperoleh melalui implementasi Algoritma Genetik..



## BAB V

### HASIL DAN PEMBAHASAN

#### A. Hasil

##### 1. Listing Program dalam MATLAB

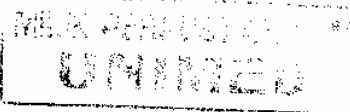
```
%Program untuk mendapatkan optimasi fungsi
%menggunakan GA P294
for i=1:25;
[x1 f1]=optga('f804',[0.001 0.3],8,10,40,0.005,0.6);
fprintf('%12.8f %12.8f\n',x1,f1);
end

function vv=f804(x)
[m,n]=size(x);
p=ones(m,n);
vv=10+(p./((x-.16).^2+0.1)).*sin(p./x);

%Program pemilih
function newchrom=selectga(criteria,chrom,a,b)
[pop bitlength]=size(chrom);
fit=[];

%Hitung fitness
[fit,fitot]=fitness(criteria,chrom,a,b);
for chromnum=1:pop
sval(chromnum)=sum(fit(1,1:chromnum));
end;

%Hitung berdasarkan fitness
parname=[];
for i=1:pop;
rval=floor(fitot*rand);
if rval<sval(1)
```



```

parname=[parname 1];
else
for j=1:pop-1
sl=sval(j);
su=sval(j)+fit(j+1);
if (rval>=sl) & (rval<=su)
parname=[parname j+1];
end
end
end
newchrom(1:pop,:)=chrom(parname,:);

%Program Optimasi GA
function [xval,maxf]=optga(fun,range,bits,pop,gens,mu,matenum)
newpop=[];
a=range(1);
b=range(2);
newpop=genbin(bits,pop);
for i=1:gens
selpop=selectga(fun,newpop,a,b);
newgen=matesome(selpop,matenum);
newgen1=mutate(newgen,mu);
newpop=newgen1;
end
[fit,fitot]=fitness(fun,newpop,a,b);
[maxf,mostfit]=max(fit);
xval=binvreal(newpop(mostfit,:),a,b);

%Program Mutasi kromosom
function chrom=mutate(chrom,mu)

%Mutasi kromosom
[pop bitlength]=size(chrom);

```

```

for i=1:pop;
for j=1:bitlength
if rand<mu
if chrom(i,j)==1;
chrom(i,j)=0;
else
chrom(i,j)=1;
end
end
end
end

% Program Pembangkit Matesome
function chrom1=matesome(chrom,matenum)
mateind=[];
chrom1=chrom;
[pop bitlength]=size(chrom);
ind=1:pop;
u=floor(pop*matenum);
if floor(u/2)~=u/2,
    u=u-1;
end;

%Memilih persentase secara random
while length(mateind)~=u
    i=round(rand*pop);
    if i==0,
        i=1;
    end;
    if ind(i)~=1;
        mateind=[mateind i];
        ind(i)=-1;
    end
end
end

```

```

%Membuat crossover
for i=1:2:u-1
    splitpos=floor(rand*bitlength);
    if splitpos==0,
        splitpos=1;
    end
    i1=mateind(i);
    i2=mateind(i+1);
    tempgene=chrom(i1,splitpos+1:bitlength);
    chrom1(i1,splitpos+1:bitlength)=chrom(i2,splitpos+1:bitlength)
;
    chrom1(i2,splitpos+1:bitlength)=tempgene;
end;

%Program Pembuatan Grafik fungsi
%untuk pengujian program
%Genetic Algorithm Grafikf804.m
x=0.001:0.0001:0.3;
[m,n]=size(x);
p=ones(m,n);
vv=10+(p./((x-.16).^2+0.1)).*sin(p./x);
plot(x,vv,'-k')
title('Fungsi Y(x)-vs-x')
xlabel('x')
ylabel('Y(x)=10+(1./((x-.16).^2+0.1)).*sin(1./x)')

%Program pembangkit generasi
function chromosome=genbin(bitl,numchrom);
maxchros=2^bitl;
if numchrom>=maxchros
    numchrom=maxchros;
end

```

```

for k=1:numchrom
    for bd=1:bitl
        if rand>=0.5
            chromosome(k,bd)=1;
        else
            chromosome(k,bd)=0;
        end;
    end;
end;

%Program pembangkit Fitness
function [fit,fitot]=fitness(criteria,chrom,a,b)
%Menghitung fitness kromosome dalam range a,b
[pop bitl]=size(chrom);
for k=1:pop
    v(k)=binvreal(chrom(k,:),a,b);
    fit(k)=feval(criteria,v(k));
end;
fitot=sum(fit);

%Program pembangkit bit
function rval=binvreal(chrom,a,b)
[pop bitlength]=size(chrom);
maxchrom=2^bitlength-1;
realel=chrom.*((2*ones(1,bitlength)).^fliplr([0:bitlength-1]));
tot=sum(realel);
rval=a+tot*(b-a)/maxchrom;

%Program Pembuatan Grafik fungsi
%untuk pengujian program
%Genetic Algorithm Grafikf804.m
x=0.001:0.0001:0.3;

```



```

[m,n]=size(x);
p=ones(m,n);
vv=10+(p./((x-.16).^2+0.1)).*sin(p./x);
plot(x,vv,'-k')
title('Fungsi Y(x)-vs-x')
xlabel('x')
ylabel('Y(x)=10+(1./((x-.16).^2+0.1)).*sin(1./x)')

%Program Lowpass Filter
x=[0.3712 -0.8545 0.8911 -0.5842 -1.3368 0.7339 0.1152 -
0.3806]';
nn=max(size(x));
for j=1:nn-1
    sol(j)=x(j);
    fprintf('\nNilai Variabel sol(%1.0f)=%8.4f\\',j,x(j));
end
fs=1000;
n=256;
k=0;
for Wo=0:0.02:1
    k=k+1;
    if Wo<=0.2 %Frekuensi Pass
        G(k)=1;
        GG(k)=0.95;
    elseif Wo>=0.3 %Frekuensi Stop
        G(k)=0;
        GG(k)=0.1;
    else
        G(k)=10*(0.3-Wo); %Nilai 0.3 frek. Stop
        GG(k)=G(k);
    end;
    z=exp(Wo*pi*i);
    H=sol(7)*((z+sol(1))*(z^2+sol(2)*z+sol(3)))/...
        ((z+sol(4))*(z^2+sol(5)*z+sol(6)));

```

```

HH(k)=abs(H);
end;
Er1=HH-G;
Er2=GG-HH;
Err1=Er1.^2;
Err2=Er2.^2;
Error1=Err1*ones(k,1);
Error2=Err2*ones(k,1);
val=10/(Error1+Error2);

%Plot hasil
b1=sol(7);
b2=sol(7)*(sol(1)+sol(2));
b3=sol(7)*(sol(3)+(sol(1)*sol(2)));
b4=sol(7)*+sol(1)*sol(3);
B=[b1 b2 b3 b4];
a1=1;
a2=sol(4)+sol(5);
a3=sol(6)+(sol(4)*sol(5));
a4=sol(4)*sol(6);
A=[a1 a3 a3 a4];

hhh=freqz(HH,n);
hy=abs(hhh);
ff=fs/(2*n)*(0:2*n-1);
fff=ff*0.002;
plot(fff/2,hy*(100/6),'-k');
title(' Grafik fungsi Lowpass Filter')
xlabel('frekuensi ')
ylabel(' H(exp(jW)) ')
grid on;

%Program Uji Highpass Filter

```

```

x=[-1.0000 0.8974 0.9163 0.5761 1.3349 0.7373 0.0831 -
0.3806]';
nn=max(size(x));
for j=1:nn-1
    sol(j)=x(j);
    fprintf('\nNilai Variabel sol(%1.0f)=%8.4f\\',j,x(j));
end
fs=1000;
n=256;
k=0;
for Wo=0:0.01:1
    k=k+1;
    if Wo<=0.7 %Frekuensi Pass
        G(k)=0;
        GG(k)=0.1;
    elseif Wo>=0.8 %Frekuensi Stop
        G(k)=1;
        GG(k)=0.95;
    else
        G(k)=10*(Wo-0.7); %Nilai 0.3 frek. Stop
        GG(k)=G(k);
    end;
    z=exp(Wo*pi*i);
    H=sol(7)*((z+sol(1))*(z^2+sol(2)*z+sol(3)))/...
        ((z+sol(4))*(z^2+sol(5)*z+sol(6)));
    HH(k)=abs(H);
end;
%Plot hasil
b1=sol(7);
b2=sol(7)*(sol(1)+sol(2));
b3=sol(7)*(sol(3)+(sol(1)*sol(2)));
b4=sol(7)*+sol(1)*sol(3);
B=[b1 b2 b3 b4];
a1=1;

```

```

a2=sol(4)+sol(5);
a3=sol(6)+(sol(4)*sol(5));
a4=sol(4)*sol(6);
A=[a1 a3 a3 a4];
hhh=freqz(B,A,n);
hy=abs(4*hhh);
ff=fs/(2*n)*(0:n-1);
fff=ff*0.002;
figure(2)
plot(fff,hy/2,'-k');
title(' Grafik fungsi Highpass Filter')
xlabel('frekuensi ')
ylabel(' H(exp(jW)) ')
grid on;

%Program Uji Bandpass Filter
x=[0.3712 -0.8545 0.8911 -0.5842 -1.3368 0.7339 0.1152 -0.3806
... 0.3712 -0.8545 0.8911 -0.5842 -1.3368 0.7339 0.1152 -
0.3806]';
nn=max(size(x));
for j=1:nn-1
    sol(j)=x(j);
    fprintf('\nNilai Variabel sol(%1.0f)=%8.4f\\',j,x(j));
end
k=0;
for Wo=0:0.02:1
    k=k+1;
    z=exp(Wo*pi*i);
    H=sol(13)*((z+sol(1))*(z^2+sol(2))*(z^2+sol(3)*z+sol(4))/...
        (z^2+sol(5)*z+sol(6)))/((z+sol(7))*(z+sol(8))*...
        (z^2+sol(9)*z+sol(10))*(z^2+sol(11)*z+sol(12)));
    HH(k)=abs(H);
end;
%Plot hasil

```

```

p1=sol(1)+sol(2);
p2=sol(7)+sol(8);
q1=sol(1)*sol(2);
q2=sol(7)*sol(8);
r1=sol(3)+sol(5);
r2=sol(9)+sol(11);
s1=sol(6)+sol(4)+(sol(3)*sol(5));
s2=sol(12)+sol(10)+(sol(9)*sol(11));
t1=sol(4)*sol(4);
t2=sol(10)*sol(12);
u1=(sol(3)*sol(6))*(sol(4)*sol(5));
u2=(sol(9)*sol(12))*(sol(10)*sol(11));
b1=sol(13);
b2=b1*(p1+r1);
b3=b1*(q1+(p1*r1)+s1);
b4=b1*(q1*r1)+(p1*s1)+u1;
b5=b1*((q1*s1)+t1+(p1*u1));
b6=b1*((q1*t1)+(q1*u1));
b7=b1*q1*t1;
B=[b1 b2 b3 b4 b5 b6 b7];
a1=1;
a2=(p2+r2);
a3=(q2*+(p2*r2)+s2);
a4=((q2*r2)+(p2*s2)+u2);
a5=((q2*s2)+t2+(p2*u2));
a6=((p2*t2)+(q2*u2));
a7=q2*t2;
A=[a1 a2 a3 a4 a5 a6 a7];
fs=1000;
n=256;
hh=freqz(B,A,n);
hy=abs(hh);
ff=fs/(2*n)*(0:n-1);
fff=ff*0.002;

```



```

plot(fff,hy/(3.46*0.9),'-k');
title(' Grafik Fungsi Bandpass Filter ')
xlabel (' Frekuensi ')
ylabel(' H(exp(jW)) ')
grid on;

%Program Uji Bandstop Filter
x=[0.3712 -0.8545 0.8911 -0.5842 -1.3368 0.7339 0.1152 -0.3806
...
0.3712 -0.8545 0.8911 -0.5842 -1.3368 0.7339 0.1152 -
0.3806]';
nn=max(size(x));
for j=1:nn-1
    sol(j)=x(j);
    fprintf('\nNilai Variabel sol(%1.0f)=%8.4f\\',j,x(j));
end
k=0;
for Wo=0:0.02:1
    k=k+1;
    z=exp(Wo*pi*i);
    H=sol(9)*((z^2+sol(1)*z+sol(2))*(z^2+sol(3)*z+sol(4)))/...
        ((z^2+sol(5)*z+sol(6))*(z^2+sol(7)*z+sol(8)));
    HH(k)=abs(H);
end;

%Plot hasil
b1=sol(9);
b2=b1*(sol(1)+sol(3));
b3=b1*(sol(4)+sol(2)+(sol(1)*sol(3)));
b4=b1*((sol(1)*sol(4))+(sol(2)*sol(3)));
b5=b1*sol(2)*sol(4);
B=[b1 b2 b3 b4 b5];
a1=1;
a2=sol(5)+sol(7);
a3=sol(8)+sol(6)+(sol(5)*sol(7));

```

```

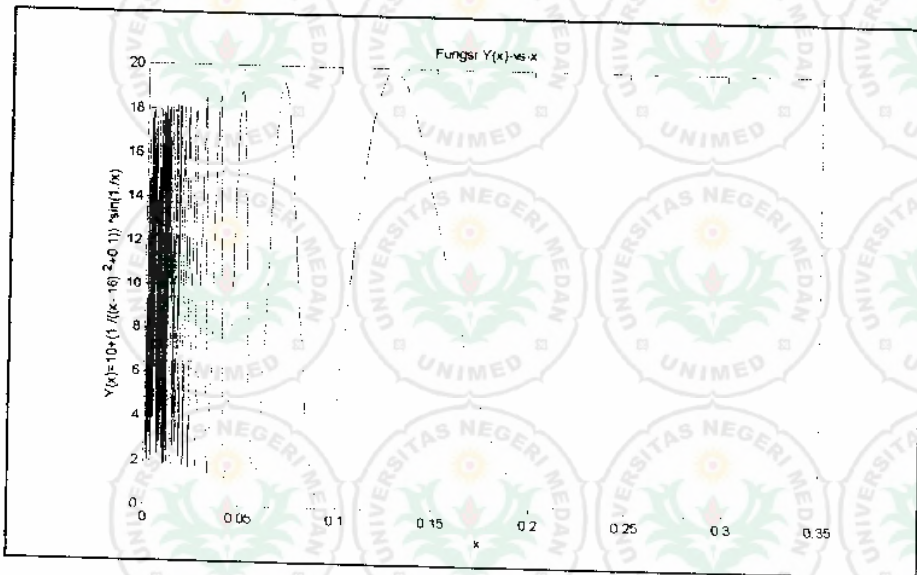
a4=(sol(5)*sol(8))+(sol(6)*sol(7));
a5=sol(6)*sol(8);
A=[a1 a2 a3 a4 a5];
fs=1000;
n=256;
hh=freqz(B,A,n);
hy=(1/4.5)*abs(hh);
ff=fs/(2*n)*(0:n-1);
fff=ff*0.002;
plot(fff,hy,'-k');
title(' Grafik fungsi Bandstop Filter')
xlabel('frekuensi ')
ylabel(' H(exp(jW)) ')
grid on;

```

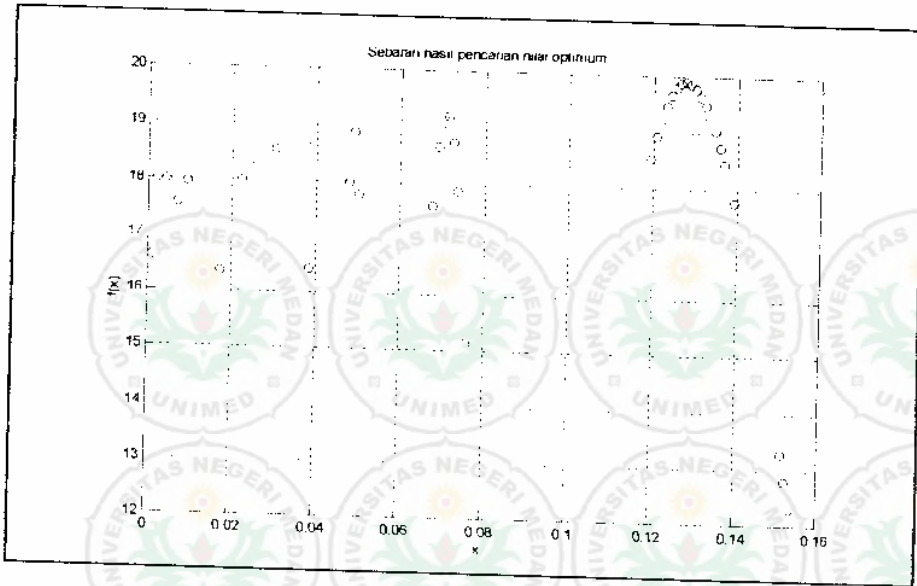
## 2. Hasil Eksekusi

### a. Pengujian fungsi

Untuk menguji program berdasarkan implementasi algoritma genetik maka digunakan sebuah fungsi uji yang memiliki banyak nilai maksimum dan minimum. Grafik fungsi yang digunakan diperlihatkan pada gambar (5.1).



Gambar 5.1 Grafik fungsi uji.



Gambar 5.2 Sebaran nilai hasil pencarian nilai optimum menggunakan algoritma genetik.

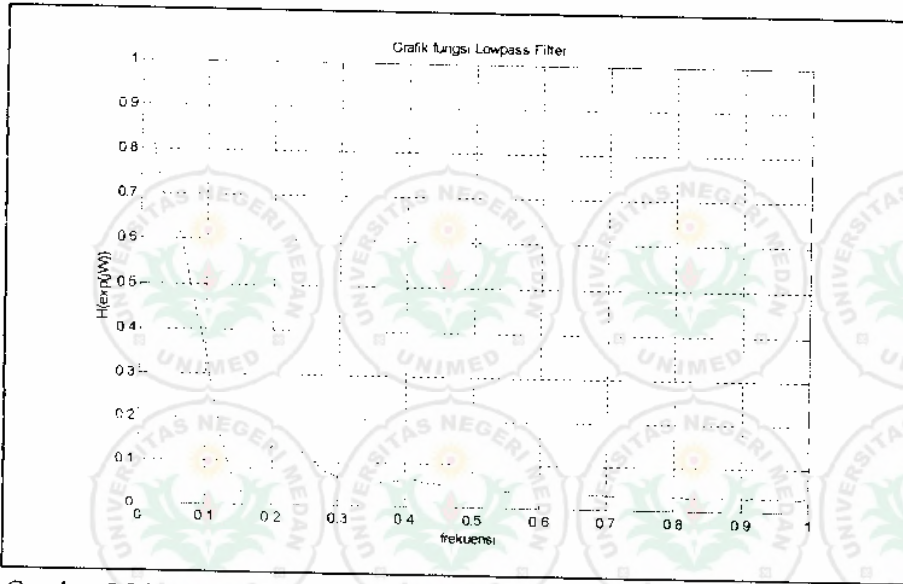
Pencarian nilai optimum menggunakan algoritma genetik pada gambar (5.2) memperlihatkan bahwa algoritma genetik mampu memperoleh nilai yang diharapkan. Hal ini menyatakan bahwa implementasi program telah dapat digunakan untuk menentukan nilai optimum dari suatu fungsi.

Hasil penelitian ini adalah didapatkannya fungsi alih untuk setiap jenis filter digital (LP, HP, BP, dan BS) melalui eksekusi pemrograman genetik secara terpadu. Adapun fungsi evaluasi dengan ruang solusi untuk masing-masing jenis filter dimodelkan sebagaimana dalam tabel (5.1) berikut ini.

Tabel 5.1 Fungsi evaluasi untuk setiap jenis filter

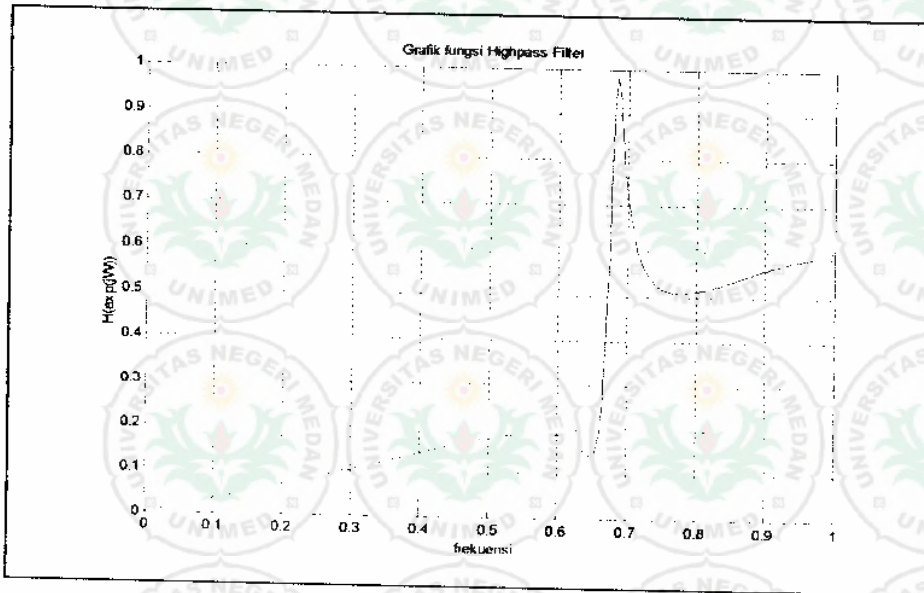
Jenis filter	Fungsi evaluasi	Kriteria
1	2	3
LP	$ H(e^{j\omega})  = \text{sol}(7) \frac{(z + \text{sol}(1))(z^2 + \text{sol}(2)z + \text{sol}(3))}{(z + \text{sol}(4))(z^2 + \text{sol}(5)z + \text{sol}(6))}$ <p>Dengan : <math>z = e^{j\omega}</math>, <math>0 \leq \omega \leq \pi</math></p> <p>Ruang solusi (1)=(2)=(3)=(4)=(6)=(7)=[-1 1]</p> <p>Ruang solusi (5)=[-2 2]</p>	<p>Orde = 3</p> <p>Tolerans :</p> <p>Pass=[1 0.95]</p> <p>Stop=[0.1 0]</p> <p><math>\omega_p = 0.2\pi</math></p> <p><math>\omega_s = 0.3\pi</math></p>
HP	$ H(e^{j\omega})  = \text{sol}(7) \frac{(z + \text{sol}(1))(z^2 + \text{sol}(2)z + \text{sol}(3))}{(z + \text{sol}(4))(z^2 + \text{sol}(5)z + \text{sol}(6))}$ <p>Dengan : <math>z = e^{j\omega}</math>, <math>0 \leq \omega \leq \pi</math></p> <p>Ruang solusi (1)=(2)=(3)=(4)=(6)=(7)=[-1 1]</p> <p>Ruang solusi (5)=[-2 2]</p>	<p>Orde = 3</p> <p>Tolerans :</p> <p>Pass=[1 0.95]</p> <p>Stop=[0.1 0]</p> <p><math>\omega_p = 0.8\pi</math></p> <p><math>\omega_s = 0.7\pi</math></p>
BP	$ H(e^{j\omega})  = \text{sol}(13) \frac{(z + \text{sol}(1))(z + \text{sol}(2))}{(z + \text{sol}(7))(z + \text{sol}(8))} \times \frac{(z^2 + \text{sol}(3)z + \text{sol}(4))(z^2 + \text{sol}(5)z + \text{sol}(6))}{(z^2 + \text{sol}(9)z + \text{sol}(10))(z^2 + \text{sol}(11)z + \text{sol}(12))}$ <p>Dengan : <math>z = e^{j\omega}</math>, <math>0 \leq \omega \leq \pi</math></p> <p>Ruang solusi (1)=(2)=(3)=(4)=(5)=(6)=(7)=(8)=(10)=(12)=(13)=[-1 1]</p> <p>Ruang solusi (9)=(11)=[-2 2]</p>	<p>Orde = 6</p> <p>Tolerans :</p> <p>Pass=[1 0.95]</p> <p>Stop=[0.1 0]</p> <p><math>\omega_{s1} = 0.24\pi</math></p> <p><math>\omega_{p1} = 0.4\pi</math></p> <p><math>\omega_{p2} = 0.6\pi</math></p> <p><math>\omega_{s2} = 0.76\pi</math></p>
BS	$ H(e^{j\omega})  = \text{sol}(9) \frac{(z^2 + \text{sol}(1)z + \text{sol}(2))}{(z^2 + \text{sol}(5)z + \text{sol}(6))} \times \frac{(z^2 + \text{sol}(3)z + \text{sol}(4))}{(z^2 + \text{sol}(7)z + \text{sol}(8))}$ <p>Dengan : <math>z = e^{j\omega}</math>, <math>0 \leq \omega \leq \pi</math></p> <p>Ruang solusi (1)=(2)=(3)=(4)=(6)=(8)=(9)=[-1 1]</p> <p>Ruang solusi (5)=(7)=[-2 2]</p>	<p>Orde = 4</p> <p>Tolerans :</p> <p>Pass=[1 0.95]</p> <p>Stop=[0.1 0]</p> <p><math>\omega_{p1} = 0.24\pi</math></p> <p><math>\omega_{s1} = 0.4\pi</math></p> <p><math>\omega_{s2} = 0.6\pi</math></p> <p><math>\omega_{p2} = 0.76\pi</math></p>

### b. Implementasi pada Lowpass Filter



Gambar 5.3 Lowpass Filter hasil pencarian algoritma genetik.

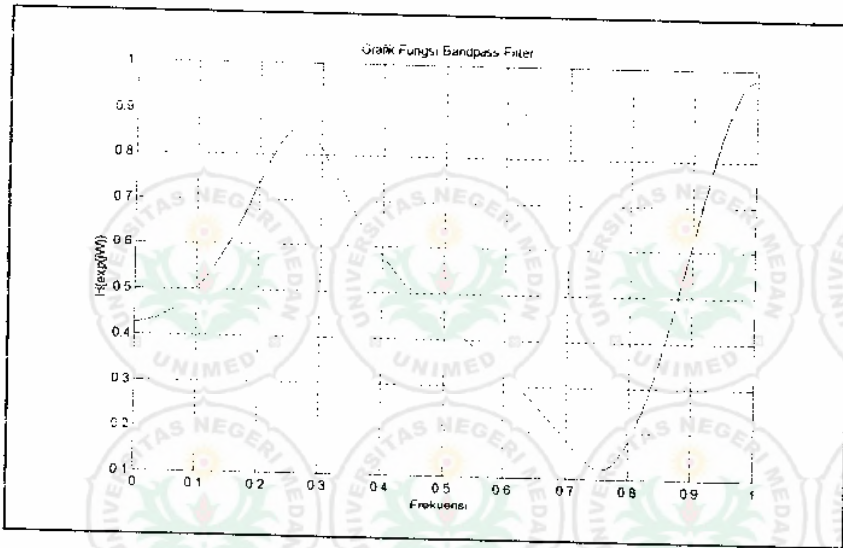
### c. Implementasi pada Highpass Filter



Gambar 5.4 Highpass Filter hasil pencarian algoritma genetik.

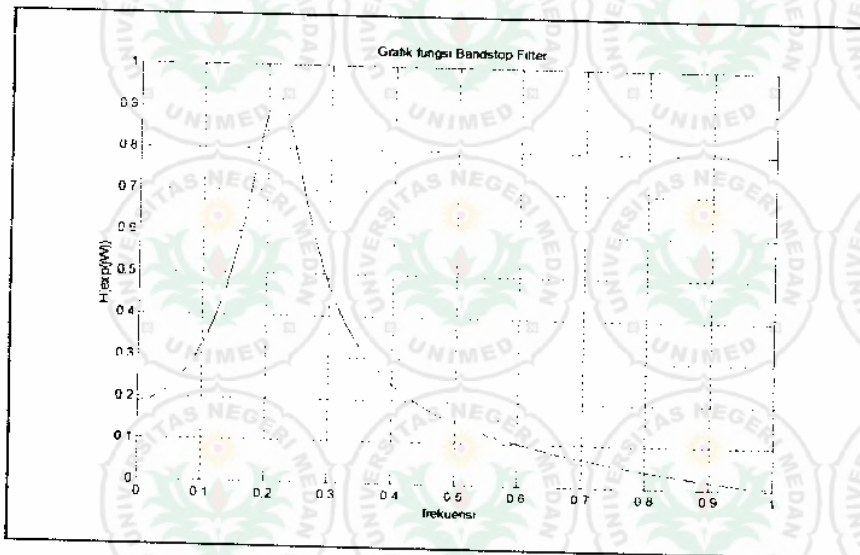


#### d. Implementasi pada Bandpass Filter



Gambar 5.5 Bandpass Filter hasil pencarian algoritma genetik.

#### e. Implementasi pada Bandstop Filter



Gambar 5.6 Bandstop Filter hasil pencarian algoritma genetik.

## B. Pembahasan

Jumlah populasi dan generasi diperoleh secara *trial and error* ketika melakukan uji program. Parameter ini sangat menentukan untuk menghasilkan nilai optimal atas fungsi alih pada setiap jenis filter. Jumlah populasi yang cukup potensial untuk mendapatkan hasil optimum berada pada kisaran 80 hingga 200 populasi, sedangkan jumlah generasi yang sesungguhnya merupakan jumlah iterasi, tergantung kepada penetapan jumlah populasi. Namun demikian, perlu diketahui bahwa ketika jumlah populasi diubah-ubah, nilai parameter yang lain harus ditetapkan.

Individu-individu yang menghasilkan nilai *fitness* fungsi evaluasi paling besar berarti juga memiliki nilai *fitness* yang tinggi. Selanjutnya individu-individu (populasi terbaik) tersebut mempunyai peluang paling besar untuk terpilih menjadi induk-induk guna melahirkan populasi baru yang lebih tangguh pada generasi berikutnya.

Berdasarkan data percobaan, nilai fungsi evaluasi dari generasi pertama menuju generasi selanjutnya, selalu mengalami perbaikan atau peningkatan nilai *fitness*, hingga pada akhirnya mencapai titik optimum. Hanya saja, perubahan nilai tersebut tidak linear pada setiap generasi.

Setelah program dieksekusi maka diperoleh solusi optimal untuk setiap jenis filter yang memenuhi kriteria. Selanjutnya fungsi evaluasi dapat dinyatakan dalam *domain z* berupa fungsi alih.

Berdasarkan hasil yang diperoleh, penelitian ini telah menjawab pertanyaan yang diajukan pada perumusan masalah dan mencapai tujuan yang diharapkan. Karenanya dapat ditarik dinyatakan bahwa algoritma genetik dapat diterapkan sebagai metoda untuk menyelesaikan masalah optimalisasi perancangan filter digital. Meskipun langkah yang ditempuh cukup panjang, namun formulasi matematis yang rumit dapat dihindari. Beban komputasi yang amat besar (terutama untuk jumlah generasi dan populasi yang besar) hingga memperlambat proses penampilan hasil, menjadi kendala utama penggunaan algoritma genetik. Tetapi hal ini kemudian dapat diatasi dan tak menjadi persoalan lagi dengan semakin tingginya kecepatan komputer.

## BAB VI

### KESIMPULAN DAN SARAN

#### A. KESIMPULAN

Berdasarkan hasil yang telah diperoleh dapat disimpulkan bahwa:

1. Program komputer yang dirancang telah dapat diimplementasikan dalam bahasa pemrograman MATLAB.
2. Algoritma genetik dapat digunakan dalam optimalisasi suatu fungsi dengan terlebih dahulu membentuk fungsi objektif.
3. Program yang dibuat dapat digunakan dalam optimalisasi filter digital.

#### B. SARAN

Algoritma yang diterapkan dalam penelitian ini dapat digunakan untuk berbagai persoalan yang terkait dengan optimalisasi, namun akan memberikan hasil yang lebih baik manakala ide dasar yang dikembangkan, dari masalah genetik, dipahami dengan baik.

## DAFTAR PUSTAKA

- Goldberg, David E. 1989. *Genetic Algorithms In Search, Optimization, And Machine Learning*. New York : Addison-Wesley Publishing Company Inc.
- Koonin, Steven E. dan Meredith, Dawn C..1990. *Computational Physics. fortran* edition, Addison-Wesley Publishing Company, New York
- Kuc, Roman. 1982. *Introduction To Digital Signal Processing*. New york : McGraw-Hill Book Co.
- Man, Tang, Kwong dan Halang. 1997. *Genetic Algorithms For Control And Signal Processing, Advances In Industrial Control*. Great Britain: Springer-Verlag London Limited.
- Proakis, John G. Rader, C. M., Ling, Fuyun dan Nikias, C. L. 1992. *Advanced Digital Signal Processing*. New York : Macmillan Publishing Company.
- Yao dan Sethares. 1994. *Nonlinier Parameter Estimation via The Genetic Algorithms*. IEEE Trans. Sign. Proc., vol. 42, no. 4, apr., pp. 927 – 935.
- Widrow, Bernard., Stearns, Samuel D. 1985. *Adaptive Signal Processing*. New Jersey : Prentice-Hall Inc., Englewood Cliffs.

Lampiran A

PERSONALIA PENELITIAN

1. Ketua Peneliti :

- a. Nama : Alkhafi Maas Siregar, S.Si, M.Si
- b. Golongan / NIP : III-b/132125659
- c. JabatanFungsional : Asisten Ahli
- d. Fakultas/Jurusan : MIPA/Fisika
- e. Perguruan Tinggi : Universitas Negeri Medan
- f. Bidang Keahlian : Fisika/Geofisika

2. Anggota Peneliti : 4 orang

- a. Nama : Drs. Abu Bakar
- Gol/NIP : III-a/131901403
- Fakultas : MIPA
- Bidang keahlian : Fisika
- b. Nama : Drs. Juniar Hutahaean, M.Si
- Gol/NIP : III-b/131950154
- Fakultas : MIPA
- Bidang keahlian : Fisika
- c. Nama : Drs. Khairul Amdani, M. Si
- Gol/NIP : III-c/132050020
- Fakultas : MIPA
- Bidang keahlian : Fisika
- d. Nama : Dra. Betty M. Turnip, M. Pd
- Gol/NIP : III-d/131663518
- Fakultas : MIPA
- Bidang keahlian : Fisika





# UNIVERSITAS NEGERI MEDAN

## (STATE UNIVERSITY OF MEDAN)

Jl. Wiliem Iskandar Psr. V Kotak Pos No.1589 – Medan 20221  
Telp. (061) 6613365, 6613276, 6618758 Fax.(061) 6614002 - 6613313

### SURAT PERINTAH KERJA (SPK)

Nomor : 01444A / J39.10/LK/2005

Tanggal : 24 Agustus 2005

Pada hari ini, Rabu tanggal dua puluh empat, bulan Agustus tahun dua ribu lima, kami yang bertanda tangan dibawah ini :

- 1. Drs. Evendi Ritonga, M.Pd** : Berdasarkan Surat Keputusan Rektor UNIMED No.: 09764 / J39/ KEP/2005, tanggal 02 Mei 2005 dalam hal ini Pejabat Pembuat Komitmen / Kuasa Penanggungjawab Administrasi Umum UNIMED (Kegiatan 5584) bertindak untuk dan atas nama Rektor untuk selanjutnya dalam SPK ini disebut sebagai **PIHAK PERTAMA.**
- 2. Prof.Dr.Abdul Muin Sibuea, M.Pd** : Ketua Lembaga penelitian UNIMED. Berdasarkan SK Pejabat Pembuat Komitmen/Kuasa Administrasi Umum UNIMED (Kegiatan 5584) Nomor : 599H/J39.16/SK/2005, tanggal 16 Mei 2005, dalam hal ini bertindak untuk dan atas nama Dosen Pelaksana Kegiatan Penelitian serta Seminar Hasil Penelitian, untuk selanjutnya dalam SK ini disebut sebagai : **PIHAK KEDUA.**

Kedua belah pihak secara bersama-sama telah sepakat mengadakan Perjanjian Kerja dengan ketentuan sebagai berikut :

#### **PASAL 1 JENIS PEKERJAAN**

Pihak Pertama memberi tugas kepada Pihak Kedua, dan Pihak Kedua menerima tugas tersebut untuk melaksanakan/koordinasi pelaksanaan 4 (empat) kegiatan Pelaksanaan Penelitian berjudul :  
1. Penelitian Tindakan Kelas (PTK) dan Penelitian Peningkatan Kualitas Pembelajaran (PPKP),  
2. Penelitian Ilmu Humaniora (Sosial, Ekonomi dan Bahasa/Seni), 3. Penelitian Pendidikan, Keolahragaan dan Kesehatan, 4. Penelitian Sains, Teknologi dan Rekayasa.

#### **PASAL 2 NILAI PEKERJAAN**

Pihak Pertama memberi dana Pelaksanaan untuk 4 (empat) Kegiatan Penelitian tersebut sebesar Rp. 94.000.000.- (Sembilan puluh empat juta rupiah), termasuk pajak-pajak yang dibebankan kepada Dana DIPA Administrasi Umum UNIMED (Kegiatan 5584) TA. 2005, dan pembayarannya secara bertahap sebagai berikut :

#### **PASAL 3 CARA PEMBAYARAN**

1. Tahap I (Pertama) sebesar 70 % yaitu Rp.65.800.000.- (Enam puluh lima juta delapan ratus ribu rupiah), dibayar sewaktu Surat Perintah Kerja (SPK) Ini ditandatangani oleh kedua belah pihak.
2. Tahap II (Kedua) sebesar 30 % yaitu Rp. 28.200.000.- (Dua puluh delapan juta dua ratus ribu rupiah), dibayar setelah Pihak Kedua menyerahkan 4 (empat) Laporan Hasil Penelitian (Kegiatan 5584) Kepada Pihak Pertama.



# UNIVERSITAS NEGERI MEDAN

## (STATE UNIVERSITY OF MEDAN)

Jl. Wiliem Iskandar Psr. V Kotak Pos No.1589 – Medan 20221  
Telp. (061) 6613365, 6613276, 6618758 Fax.(061) 6614002 - 6613319

### PASAL 4 JANGKA WAKTU PELAKSANAAN

Pihak Kedua wajib menyelesaikan Kegiatan Pelaksanaan Penelitian dimaksud dalam pasal 1 SPK ini selambat-lambatnya tanggal 14 Nopember 2005, sejak tanggal SPK ini.

### PASAL 5 LAPORAN

- Pihak Kedua menyampaikan 4 (empat) Laporan akhir Kegiatan Pelaksanaan Penelitian kepada Pihak Pertama sebanyak 6 (enam) eksemplar yang akan didistribusikan kepada :
  - Pihak Pertama sebanyak 4 (empat) laporan, masing-masing 1 (satu) eksemplar (ASLI) + copy
  - Lembaga Penelitian sebanyak 4 (empat) laporan, masing-masing 1 (satu) eksemplar beserta artikel dan berkas lain yang diminta oleh LP UNIMED
  - Kantor Pelayanan dan Perbendaharaan Negara (KPPN) Medan sebanyak 4 (empat) laporan, masing-masing 1 (satu) eksemplar.
  - Direktorat Pembinaan Penelitian dan Pengabdian Kepada Masyarakat (DP3M) Direktorat Jenderal Pendidikan Tinggi Depdiknas RI sebanyak 4 (empat) laporan, masing-masing 2 (dua) eksemplar.
- Sistematika Laporan Akhir Kegiatan Pelaksanaan Penelitian harus mematuhi ketentuan seperti yang ditetapkan dalam buku Panduan Pelaksanaan Penelitian dan Pengabdian pada Masyarakat Edisi VI Tahun 2002 yang dikeluarkan oleh DP3M Direktorat Jenderal Pendidikan Tinggi Depdiknas RI.
- Bersamaan dengan Laporan Akhir Pelaksanaan, PIHAK KEDUA juga menyampaikan Ringkasan Hasil Kegiatan dan artikel ilmiah.

### PASAL 6 SANKSI

Apabila Pihak Kedua dalam melaksanakan kegiatan seperti tercantum pada pasal 1 penyelesaian laporan hasil, maka Pihak Kedua dikenakan sanksi :

- Denda sebesar 1 % perhari dengan maksimum denda sebesar 5 % dari nilai Surat Perintah Kerja (SPK)
- Tidak akan diikutsertakan dalam kegiatan Penelitian berikutnya.

### PASAL 7

Surat Perintah Kerja (SPK) ini dibuat rangkap 6 (enam) dengan ketentuan sebagai berikut :

- (satu) lembar pada : Administrasi Umum UNIMED
- (satu) lembar pada : Ketua Pelaksana Kegiatan Pelaksanaan Penelitian
- (tiga) lembar pada : Kantor Pelayanan dan Perbendaharaan Negara (KPPN) Medan
- (satu) lembar pada : Lembaga Penelitian UNIMED

Pihak Kedua :

Ketua Tim Pelaksana,

  
Prof. Dr. Abdul Muhi Sibuea, M.Pd.  
NIP. 130935473

Pihak Pertama :

Pejabat Pembuat Komitmen /  
Kuasa Penanggungjawab Kegiatan 5584

  
Drs. Evendi Ritonga, M.Pd  
NIP. 131272205