

LAPORAN PENELITIAN

**PENYELESAIAN FUNGSI NON-KONVEKS
MENGUNAKAN TEKNIK OPTIMASI BERBASIS
ANNEALING**



Oleh :
Supardi, S.Si
Restu Widiatmono, S.Si

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2002**

KATA PENGANTAR

Puji syukur kehadirat Alloh swt yang telah memberikan rahmat dan hidayahNya, sehingga pada saat ini kami dapat menyelesaikan dan melaporkan hasil penelitian yang berjudul “ **PENYELESAIAN FUNGSI NON-KONVEKS MENGGUNAKAN TEKNIK OPTIMASI BERBASIS ANNEALING** ”. Melalui penelitian ini diharapkan dapat meningkatkan kualitas penelitian di bidang komputasi di Jurusan Fisika Universitas Negeri Yogyakarta.

Penelitian ini dapat dilakukan dan diselesaikan dengan baik atas bantuan beberapa pihak yang secara keseluruhan tidak dapat kami sebutkan satu persatu, untuk itu pada kesempatan ini peneliti ingin menyampaikan penghargaan yang setinggi-tingginya kepada :

1. Bapak Dekan FMIPA Universitas Negeri Yogyakarta, yang telah memberikan kesempatan dan fasilitas kepada peneliti
2. Bapak Ketua Jurusan Pendidikan Fisika FMIPA UNY yang telah memberikan dorongan untuk terus melakukan penelitian.
3. Bapak Dr.Mundlilarto selaku Badan Pertimbangan Penelitian FMIPA UNY yang telah memberika masukan untuk perencanaan, pelaksanaan, dan penyusunan laporan ini.
4. Bapak dan Ibu Dosen Jurusan Pendidikan Fisika yang telah memberikan masukan-masukan demi sempurnanya laporan penelitian ini.

Peneliti berharap semoga hasil penelitian ini bermanfaat bagi pengembangan ilmu Fisika khususnya pada bidang komputasi.

Yogyakarta, Mei 2002

Supardi, M.Si

DAFTAR ISI

HALAMAN JUDUL	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
ABSTRAK	iv
BAB I PENDAHULUAN	1
Rumusan Masalah	2
Tujuan Penelitian	2
Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	3
BAB III METODE KOMPUTASI	7
BAB IV PEMBAHASAN	8
BAB V KESIMPULAN	11
DAFTAR PUSTAKA	12
LAMPIRAN-LAMPIRAN	13
LISTING PROGRAM	13

PENYELESAIAN FUNGSI NON-KONVEKS MENGGUNAKAN TEKNIK OPTIMASI BERBASIS ANNEALING

**Oleh : Supardi dan Restu Widiatmono
Jurdik Fisika FMIPA Universitas Negeri Yogyakarta**

ABSTRAK

Telah dilakukan kajian terhadap metode optimasi berbasis annealing untuk mengoptimasi fungsi non-konfiks. Inti dari metode ini merupakan analogi didalam termodinamika yaitu suatu cara bagaimana cairan dibekukan dari suhu tinggi hingga terbentuk kristal. Pada suhu tinggi molekul-molekul cairan dapat bergerak bebas. Jika cairan tersebut didinginkan secara perlahan-lahan maka mobilitas termal molekul-molekul tersebut akan hilang. Dengan cara demikian, molekul-molekul memiliki kesempatan untuk menata diri sehingga terbentuk kristal. Bentuk kristal tersebut berada dalam keadaan energi paling minimum (minimum global) sistem tersebut. Hasil yang telah diperoleh dari kajian ini menunjukkan bahwa optimasi dengan menggunakan metode simulated annealing sangat efektif dan handal dalam menyelesaikan masalah optimasi fungsi non-konfiks.

Kata kunci : Simulated annealing, fungsi non konfiks, optimasi.

BAB I

PENDAHULUAN

Di dalam kehidupan sehari-hari, sering dihadapkan oleh permasalahan yang memerlukan penyelesaian yang tepat dan cepat. Seringkali harus dipilih diantara banyak kemungkinan penyelesaian yang nampak optimal dalam beberapa hal. Langkah berikutnya, segera dianalisa beberapa kemungkinan penyelesaian tersebut dan dipilih satu penyelesaian yang paling efektif dan efisien yang dapat menyelesaikan berbagai masalah minimisasi dengan optimal.

Dari sudut pandang teknis, prosedur optimasi adalah suatu masalah pencarian harga ekstremum dari suatu fungsi tujuan (*objective function*) yang telah sebelumnya didefinisikan, misalnya suatu fungsional di dalam ruang dimensi D ($X \in R^D$). Berbagai macam masalah dalam matematika, fisika, kimia, teknik, ekonomi atau disiplin lain dapat disajikan menjadi masalah minimisasi.

Apabila fungsi tujuan adalah *konveks* (yaitu fungsi tujuan yang memiliki harga minimum tunggal), maka masalah tersebut dapat diselesaikan dengan mudah dengan metode *gradient descent* (Press et al, 1986) atau dengan metode sederhana yang lain. Akan tetapi ketika dihadapkan dengan fungsi tujuan yang memiliki banyak minimum lokal (fungsi *non-konveks*), algoritma *gradient descent* atau metode optimasi yang lain juga akan menemukan harga minimum lokal, alih-alih minimum global yang diinginkan.

Masalah yang timbul dari fungsi tujuan yang demikian adalah menemukan algoritma yang tepat untuk memperoleh minimum global fungsi tersebut. Algoritma tersebut harus memiliki fitur yang memungkinkan untuk menaiki bukit agar dapat keluar

dari jebakan minimum lokal. Algoritma yang paling efisien untuk menangani masalah yang demikian adalah *algoritma Simulated Annealing* (Kirkpatrick et al,1983).

Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana mendapatkan harga minimum global dari fungsional yang memiliki tiga harga minimum. Kemudian, hasilnya akan dibandingkan dengan metode *bracketting*.

Tujuan Penelitian

Penelitian ini dimaksudkan untuk mencari harga minimum global sebuah fungsional non-konveks dengan menggunakan metode optimasi berbasis annealing dan hasilnya akan dibandingkan dengan metode minimisasi lain (*bracketting*).

Manfaat Penelitian

Hasil penelitian ini akan memberi solusi yang tepat di dalam penyelesaian terhadap masalah optimasi fungsi yang memiliki harga minimum lebih dari satu. Metode tersebut dapat diterapkan di dalam berbagai disiplin ilmu fisika, kimia, biologi bahkan ekonomi.

BAB II

TINJAUAN PUSTAKA

Istilah simulated annealing diambil dari sebuah proses yang digunakan dalam bidang metalurgi, yaitu pendinginan secara perlahan-lahan dari sebuah bahan logam yang dicairkan hingga mencapai keadaan optimum berbentuk kristal. Mengambil analogi dengan proses termal dalam mekanika statistik, maka di dalam prosedur simulated annealing diperkenalkan *suhu buatan* sebagai sumber stokastik yang menunjukkan terjadinya pelepasan diri dari jebakan minimum lokal (Stariolo dan Tsalis, 1995). Langkah selanjutnya adalah menemukan proses annealing untuk mencapai pendinginan suhu tercepat yang masih menjamin tidak terjebak pada minimum lokal. Dengan kata lain, harus dicari cara annealing tercepat yang belum dikategorikan dalam *proses quenching*.

Algoritma Metode Simulated Berbasis Annealing Untuk Variabel Kontinu

Dalam penelitian ini digunakan algoritma simulated annealing untuk masalah optimasi global variabel kontinu mengambil bentuk

$$f^* = \min_{x \in X} f(x) \dots\dots\dots(1)$$

dengan $x \in R^n$ adalah domain kontinu (Locatelli, 2000).

Sebagaimana telah disinggung di depan bahwa algoritma simulated annealing didasarkan pada analogi gejala fisis. Metode Metropolis Monte Carlo telah diusulkan untuk mensimulasikan proses annealing secara fisis tersebut. Analogi yang kedua yaitu antara konfigurasi sistem fisis dengan titik-titik fisibel di dalam masalah optimasi, dan antara fungsi energi dengan fungsi tujuan pada masalah optimasi menuju pada suatu

definisi tentang algoritma simulated annealing untuk menyelesaikan masalah optimasi kombinatorial. Kajian selanjutnya telah diperluas untuk masalah optimasi secara umum dengan variabel kontinu. Lebih jelasnya, algoritma simulated annealing untuk optimasi global variabel kontinu dapat digambarkan sebagai berikut (Locatelli, 2000).

Langkah 0 Dimisalkan $x_0 \in X$ adalah titik awal yang diberikan, sedangkan

$$z_0 = \{x_0\} \text{ dan } k=0.$$

Langkah 1 Sampel titik y_{k+1} dari *distribusi kandidat berikutnya*

Langkah 2 Sampel bilangan random uniform p pada interval $[0,1]$ dan diambil

$$x_{k+1} = \begin{cases} y_{k+1} & \text{jika } p \leq A(x_{k+1}, y_{k+1}, t_k) \\ x_{k+1} & \text{jika sebaliknya} \end{cases}$$

dengan A disebut fungsi penerimaan dan memiliki rentang nilai $[0,1]$, sedangkan t_k adalah parameter kontrol disebut suhu pada iterasi yang ke k .

Langkah 3 Diambil $z_{k+1} = z_k \cup \{y_{k+1}\}$, dengan z_k memuat semua informasi yang dikumpulkan sampai pada iterasi yang ke k , atau dengan kata lain semua titik diobservasi sampai pada iterasi ini.

Langkah 4 diambil $t_{k+1} = U(z_{k+1})$, dengan U disebut annealing schedule dan merupakan fungsi dengan nilai tak negatif (*nonnegative values*).

Langkah 5 Ditentukan kriteria henti (*stopping criterion*), apabila tak berhasil maka harus diambil lagi $k:=k+1$ dan kembali ke langkah 1.

Komponen Dasar Algoritma Metode Optimasi Berbasis Annealing

Fungsi Penerimaan

Dalam banyak kasus, fungsi penerimaan (*acceptance function*) A dikenal dengan *fungsi Metropolis* yang mengambil bentuk

$$A(x, y, t) = \min\left\{1, \exp\left\{-\frac{f(y) - f(x)}{t}\right\}\right\} \dots\dots\dots(2-2)$$

Fungsi penerimaan Metropolis selalu menerima langkah yang menurun atau langkah menuju kandidat titik yang mengalami perbaikan, atau paling tidak membuat nilai fungsi tidak lebih buruk dari sebelumnya atau $f(y_{k+1}) \leq f(x_k)$. Akan tetapi, agar tidak terperangkap pada minimum lokal, maka akan diterima pula langkah yang bergerak naik atau $f(y_{k+1}) \geq f(x_k)$.

Kriteria Penerimaan (Acceptance Criterion)

Hukum termodinamik menjelaskan bahwa pada suhu T probabilitas kenaikan energi δE diberikan oleh

$$P(\delta E) = \exp(-\delta E / kT) \dots\dots\dots(2-3)$$

dengan k merupakan konstanta disebut konstanta Boltzman(Metropolis, 1953).

Dengan memperhatikan algoritma Metropolis pada simulasi perilaku suatu sistem pada suhu tertentu, di setiap iterasi selalu dibangkitkan secara random solusi baru sol' yang merupakan tetangga terdekat dari penyelesaian sol . Apabila penyelesaian yang ditawarkan oleh sol' tersebut lebih baik dari penyelesaian sebelumnya sol , maka penyelesaian baru tersebut akan diterima dan selanjutnya akan menjadi penyelesaian sol yang baru. Penyelesaian sol yang baru akan dipakai sebagai titik tolak untuk mencari penyelesaian yang ditawarkan oleh tetangganya yang baru lagi secara acak sampai diperoleh penyelesaian baru lagi sol' . Demikian aktifitas ini selalu diulang-ulang hingga diperoleh penyelesaian yang paling optimal.

Syarat bahwa penyelesaian yang ditawarkan oleh tetangga diterima memiliki probabilitas sebesar p bergantung kepada selisih energi dari dua penyelesaian (energi tetangga dikurangi dengan energi kini). Probabilitas p akan mengalami peningkatan apabila suhu yang dikenakan semakin tinggi dan p akan mengalami penurunan jika selisih energi berkurang. Skema pemilihan yang demikian disebut dengan *kriteria Metropolis* (Metropolis, 1953).

Persamaan (2-3) digunakan secara langsung pada simulated annealing, meskipun biasanya bentuk konstanta Boltzman tidak di munculkan lagi untuk mengatasi masalah yang berbeda. Oleh karena itu, probabilitas penerimaan terhadap keadaan yang tidak lebih baik diberikan dengan persamaan

$$P = \exp\left(-\frac{c}{T}\right) > r \quad \dots\dots\dots(2-4)$$

dengan c : perubahan fungsi, T : suhu kini, r : bilangan random antara 0 dan 1.

Simulated annealing terdiri atas sederetan rantai Metropolis pada setiap penurunan suhu yang berbeda. Tujuan dari rantai Metropolis ini tak lain adalah untuk mengijinkan sistem mencapai kesetimbangan termalnya. Perubahan perlahan-lahan dalam simulated annealing disebut *annealing schedule*. Terdapat empat komponen annealing schedule yaitu, suhu awal, suhu akhir, aturan penurunan suhu, iterasi pada setiap suhu tertentu.

Temperature schedule merupakan komponen penting algoritma simulated annealing. Schedule yang baik harus efisien yaitu menghasilkan penyelesaian akhir paling baik dan umum (general) yang dapat diterapkan pada berbagai masalah.

Annealing Schedule

Sebagaimana telah disinggung di atas, bahwa ada empat komponen annealing schedule atau *cooling schedule* yaitu, suhu awal, suhu akhir, aturan penurunan suhu, dan iterasi pada setiap suhu tertentu. Untuk lebih jelasnya akan dijelaskan satu persatu.

Suhu awal

Suhu awal harus cukup tinggi untuk memungkinkan perpindahan pada keadaan tetangga. Apabila suhu yang dipilih tidak cukup tinggi, maka penyelesaian yang akan diperoleh hanya akan sama (atau sangat dekat) dengan penyelesaian awal. Akan tetapi, pemilihan suhu yang terlalu tinggi dapat menyebabkan pencarian akan berpindah-pindah hampir ke seluruh tetangga dan mentransformasi pencarian tersebut menjadi pencarian yang sifatnya acak. Hal ini setidaknya terjadi pada putaran atau langkah pertama.

Masalah yang muncul kemudian adalah bagaimana mendapatkan suhu awal yang tepat mengingat adanya metode analitik yang dapat digunakan sebagai panduan. Oleh karena itu, teknik yang ditempuh biasanya adalah secara heuristik. Apabila telah diketahui selisih maksimum (selisih dari fungsi tujuan) antara satu tetangga dengan tetangga lain, maka informasi ini dapat dipergunakan sebagai informasi awal dalam mengkalkulasi suhu awal.

Metode lain telah disarankan oleh Rayward dan Smith (1995). Mereka mengusulkan suhu awal sebaiknya dimulai dari suhu yang sangat tinggi, kemudian mendinginkannya dengan cepat sampai diterima sekitar 60% dari penyelesaian lebih buruk (*worse solution*). Setelah itu, barulah dimulai pendinginan secara perlahan-lahan.

Metode serupa telah diusulkan oleh Dowsland (1996). Dia menyarankan untuk memanaskan sistem dengan cepat sampai proporsi penyelesaian lebih buruk berjumlah cukup. Setelah itu, pendinginan perlahan-lahan baru dimulai. Hal ini bisa dilihat pada

aspek annealing secara fisis yakni bahan dipanaskan hingga berada pada keadaan cair, kemudian pendinginan secara perlahan-lahan baru dimulai.

Suhu Akhir

Idealnya, proses optimasi berhenti ketika sistem berada pada titik beku yaitu keadaan ketika sudah tidak terjadi lagi perubahan energi sistem. Dalam algoritma yang digunakan, prinsip ini digunakan, yakni optimasi dihentikan ketika selisih energi sudah tidak terjadi lagi perubahan pada rantai berturut-turut.

Penurunan Suhu (Temperature Decreament)

Setelah dimiliki suhu awal dan suhu akhir, selanjutnya diperlukan cara untuk mendapatkan satu suhu dengan suhu yang lain. Dalam hal ini maka diperlukan suatu cara untuk menurunkan suhu sedemikian hingga akhirnya sampai pada kriteria pemberhentian.

Suatu cara bagaimana menurunkan suhu merupakan hal penting untuk sukses atau tidaknya algoritma dalam program. Teori menerangkan bahwa pada setiap suhu tertentu, maka dibutuhkan iterasi yang cukup, supaya sistem berada pada keadaan stabil pada suhu tersebut (Ingber, 1999). Sayangnya, ada teori juga yang mengatakan bahwa untuk mencapai cacah iterasi yang cukup pada setiap suhu tertentu adalah eksponensial terhadap ukuran masalah. Tetapi hal ini dapat dikompromikan. Untuk mendapatkan minimum global dapat dilakukan baik dengan melakukan iterasi yang banyak pada sejumlah kecil suhu atau melakukan iterasi sedikit tetapi jumlah suhu yang banyak ataupun keduanya seimbang.

Sedangkan aturan penurunan suhu yang sering digunakan adalah $T_i = \alpha T_{i-1}$

.....(2-6)

dengan $\alpha < 1$. Menurut pengalaman, nilai α seharusnya berada antara 0.8 dan 0.99 dan hasil yang lebih baik akan diperoleh pada nilai di ujung range. Konsekuensinya, penurunan suhu akan membutuhkan waktu yang lebih lama untuk sampai pada kriteria pemberhentian.

Iterasi pada Setiap Suhu Tertentu

Keputusan terakhir yang harus diambil adalah berapa banyak iterasi yang akan dilakukan pada setiap suhu tertentu. Cacah iterasi yang konstan merupakan cara yang biasa dilakukan.

Metode lain telah disarankan oleh Lundy (1986). Dia menyarankan untuk hanya melakukan satu kali iterasi pada setiap suhu, akan tetapi penurunan suhu yang dilakukan sangat ekstra perlahan-lahan. Formulasi yang digunakan mengambil bentuk

$$t = \frac{t}{1 + \beta t} \dots\dots\dots(2-7)$$

dengan β adalah konstanta bernilai kecil yang sesuai.

Alternatif lain yang dapat dilakukan adalah dengan mengubah secara dinamis cacah iterasi pada setiap suhu selama algoritma berlangsung. Pada suhu rendah, sangat penting untuk mengenakan cacah iterasi dalam jumlah besar, sehingga optimum lokal dapat digali secara penuh. Sedangkan pada suhu tinggi, cacah iterasi dapat lebih kecil.

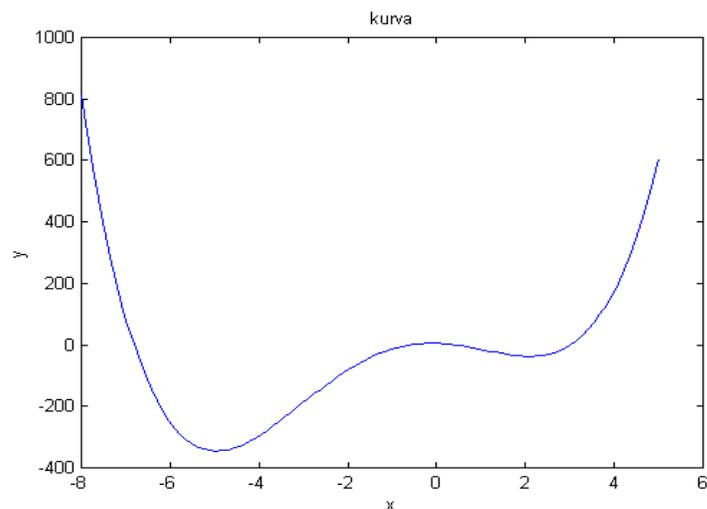
BAB III

METODE PENELITIAN

Salah satu keunggulan dari metode metode simulated annealing adalah kemampuannya untuk mengoptimasi variabel-variabel pada suatu fungsional dengan banyak harga minimum. Disamping itu, metode ini juga dapat mengatasi jebakan-jebakan minimum lokal yang dimiliki sistem.

Sehubungan dengan hal itu, dalam penelitian ini akan dikaji sebuah fungsi non-konveks yaitu sebuah fungsi yang memiliki harga minimum lebih dari satu. Oleh sebab itu, bentuk paling sederhana fungsi non-konveks adalah fungsi dengan cacah harga minimum sama dengan dua. Satu harga minimum merupakan minimum lokal, sedangkan yang lainnya adalah minimum globalnya. Pemilihan terhadap fungsi non-konveks paling sederhana tersebut mengambil bentuk

$$f(x) = x^4 + 4x^3 - 20x^2 - 5x + 5$$



Gambar 1. Bentuk grafis dari fungsi

Jika diperhatikan pernyataan () terlihat bahwa suku-suku persamaan tersebut memiliki bentuk non-linier. Ini berarti bahwa persamaan tersebut memiliki lebih dari satu harga minimum sebagaimana terlihat pada gambar 1. Dengan menggunakan metode simulated annealing, akan dicari bentuk paling minimum harga dari fungsional tersebut. Harga paling minimum dari fungsional tersebut dikenal dengan *harga minimum global*, sedangkan harga minimum yang berada di atas minimum global disebut dengan *harga minimum lokal*. Metode optimasi simulated annealing telah memberi janji bahwa metode tersebut akan dapat mengatasi jebakan-jebakan minimum lokal yang ada pada fungsional tersebut. Flowchart untuk metode simulated annealing dapat dilihat pada gambar (2.).

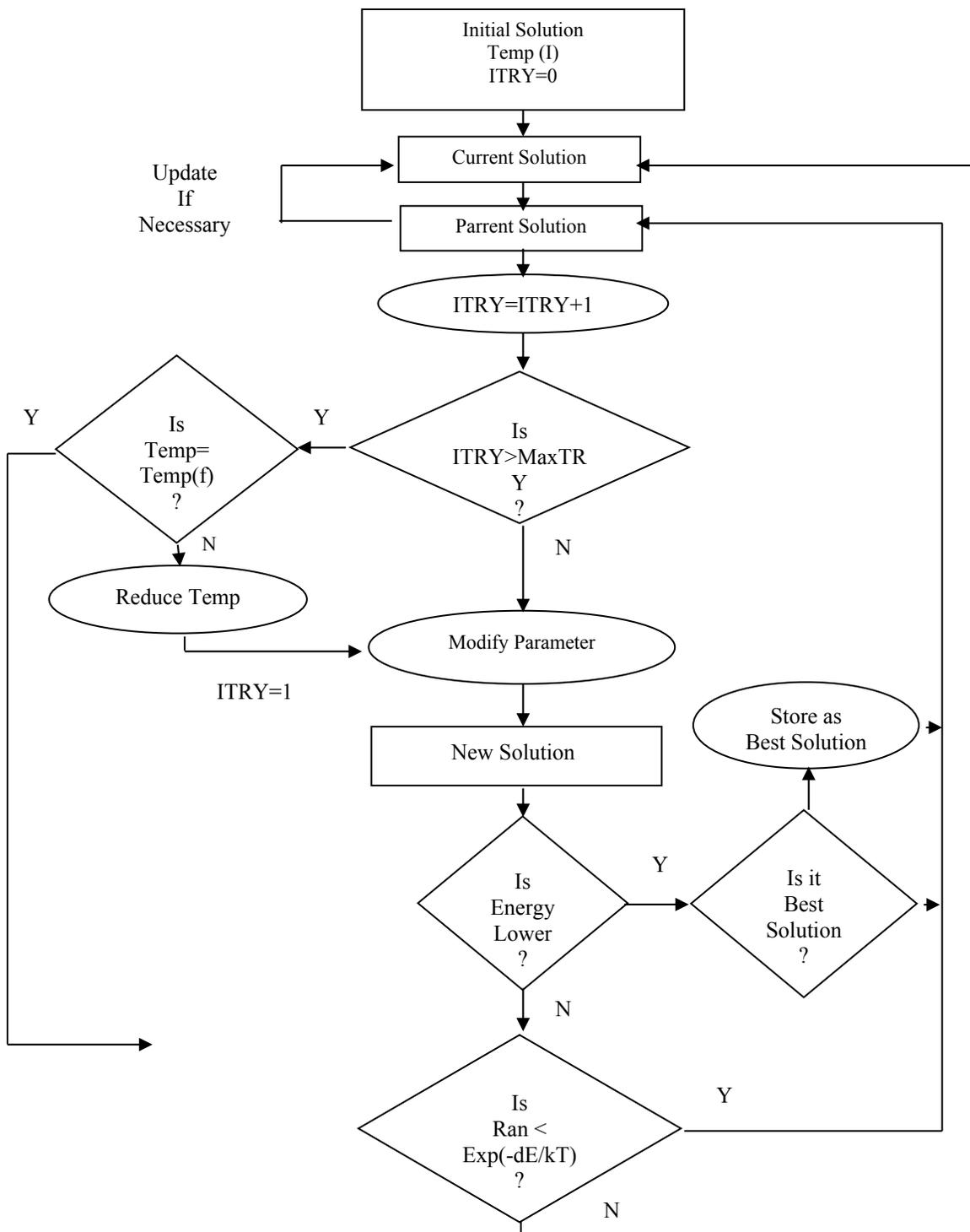


Diagram alir metode Simulated Annealing

BAB IV

HASIL DAN PEMBAHASAN

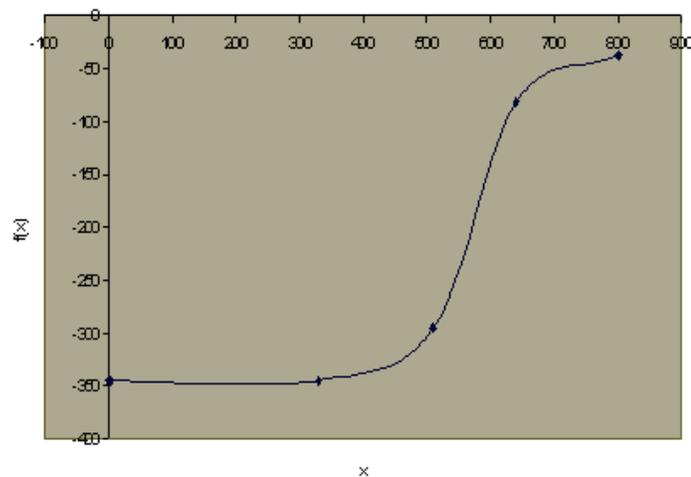
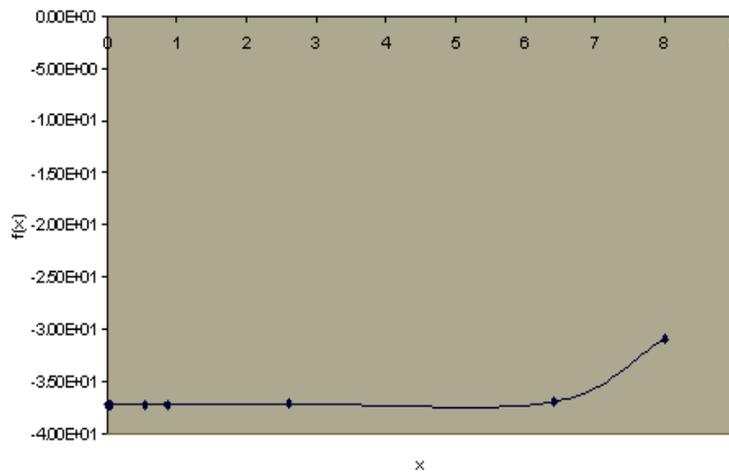
Tabel 5.1 ditunjukkan hasil dari proses minimisasi yang dilakukan terhadap fungsional Ginzburg-Landau dengan menggunakan metode simulated annealing diawali dengan masukan terhadap suhu awal sistem T_{awal} . Suhu awal T_{awal} telah dipilih sedemikian hingga proses annealing pada bahan memberikan kesempatan kepada sistem untuk menata diri hingga seluruh keadaan (*state*) sistem berada pada keadaan setimbang. Secara fisis, apabila seluruh keadaan sistem pada tiap-tiap suhu yang dilewati dalam keadaan setimbang, maka susunan atom-atom sistem pada keadaan akhir akan membentuk formasi kristal sempurna (*regular crystal*). Sebaliknya, apabila dalam proses penurunan suhu terdapat satu atau beberapa keadaan tak setimbang, maka susunan atom-atom terbolehjadi yang terjadi pada sistem bukanlah merupakan bentuk kristal melainkan bentuk amorf.

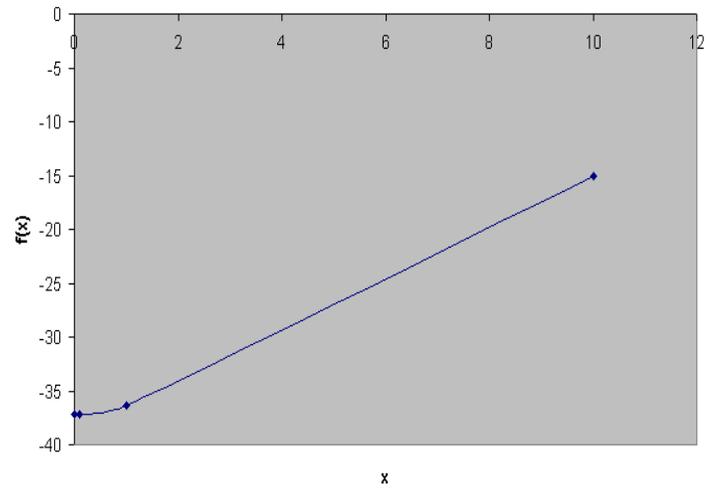
Formasi atom-atom sistem dijamin berbentuk kristal sempurna setelah dilakukan penurunan suhu secara perlahan dan hati-hati. Pada penelitian ini telah dilakukan penurunan suhu dengan aturan $T_{baru} = 0,8T_{lama}$. Penurunan suhu dengan aturan demikian penulis yakini sebagai aturan yang moderat, artinya penurunan tidak terlalu lambat, akan tetapi tidak pula terlalu cepat. Aturan penurunan suhu tersebut

dimaksudkan supaya proses annealing pada sistem tidak terlalu lambat, tetapi masih dapat dijamin hasil akhir yang diperoleh.

Masukan untuk suhu awal $T_{awal} = 1000,0$. Pada ketinggian suhu yang demikian diharapkan sistem tidak lagi terjebak kepada minimum lokal, akan tetapi dapat menemukan minimum globalnya.

Masukan nilai awal untuk setiap variabel yang akan dioptimasi untuk kasus radius bola jauh lebih besar dari panjang koherensi adalah $\{g_{i,j}\} = 10,0$ untuk $i = 1..5, j = 2..10$. Sedangkan untuk kasus radius bola superkonduktor jauh lebih kecil dibanding panjang koherensinya masukan nilai awal $\{g_{i,j}\} = 1,0$ untuk $i = 1..5, j = 1..11$. Pada setiap suhu tertentu dilakukan pengaturan konfigurasi sebanyak 50 kali langkah acak, sehingga jumlah evaluasi terhadap fungsional Ginzburg-Landau terlinierisasi untuk setiap suhu





Listing Program Optimasi Berbasis Annealing

```
SUBROUTINE amebssa(p,y,mp,np,ndim,pb,yb,ftol,funk,iter,temptr)
INTEGER iter,mp,ndim,np,NMAX
REAL ftol,temptr,yb,p(mp,np),pb(np),y(mp),funk
PARAMETER (NMAX=200)
EXTERNAL funk
CU  USES amotsa,funk,ran1
INTEGER i,idum,ih,i,ilo,inhi,j,m,n
REAL rtol,sum,swap,tt,yhi,ylo,ynhi,ysave,yt,ytry,psum(NMAX),
*amotsa,ran1
COMMON /ambssa/ tt,idum
tt=-temptr
1  do 12 n=1,ndim
    sum=0.
    do 11 m=1,ndim+1
        sum=sum+p(m,n)
11  continue
    psum(n)=sum
12  continue
2  ilo=1
    inhi=1
    ihi=2

    ylo=y(1)+tt*log(ran1(idum))
    ynhi=ylo
    yhi=y(2)+tt*log(ran1(idum))
    if (ylo.gt.yhi) then
        ihi=1
        inhi=2
        ilo=2
        ynhi=yhi
        yhi=ylo
```

```

    ylo=ynhi
endif
do 13 i=3,ndim+1
    yt=y(i)+tt*log(ran1(idum))
    if(yt.le.ylo) then
        ilo=i
        ylo=yt
    endif
    if(yt.gt.yhi) then
        inhi=ihl
        ynhi=yhi
        ihl=i
        yhi=yt
    else if(yt.gt.ynhi) then

        inhi=i
        ynhi=yt
    endif
13 continue
    rtol=2.*abs(yhi-ylo)/(abs(yhi)+abs(ylo))
    if (rtol.lt.ftol.or.iter.lt.0) then
        swap=y(1)
        y(1)=y(ilo)
        y(ilo)=swap
        do 14 n=1,ndim
            swap=p(1,n)
            p(1,n)=p(ilo,n)
            p(ilo,n)=swap
14 continue
        return
    endif
    iter=iter-2
    ytry=amotsa(p,y,psum,mp,np,ndim,pb,yb,funk,ihl,yhi,-1.0)
    if (ytry.le.ylo) then
        ytry=amotsa(p,y,psum,mp,np,ndim,pb,yb,funk,ihl,yhi,2.0)
    else if (ytry.ge.ynhi) then
        ysave=yhi
        ytry=amotsa(p,y,psum,mp,np,ndim,pb,yb,funk,ihl,yhi,0.5)

```

```

    if (ytry.ge.ysave) then

        do 16 i=1,ndim+1
            if(i.ne.ilo)then
                do 15 j=1,ndim
                    psum(j)=0.5*(p(i,j)+p(ilo,j))
                    p(i,j)=psum(j)
15                continue
                    y(i)=funkt(psum)
                endif
16            continue
                iter=iter-ndim
                goto 1
            endif
        else
            iter=iter+1
        endif
        goto 2
    END

```

xamebsa.for

```

PROGRAM xamebsa
C  driver for routine ambsa
INTEGER NP,MP,MPNP
REAL FTOL
PARAMETER(NP=4,MP=5,MPNP=20,FTOL=1.0E-6)
INTEGER i,idum,iiter,iter,j,jiter,ndim,nit
REAL tempr,tt,yb,ybb
COMMON /ambsa/ tt,idum
REAL p(MP,NP),x(NP),y(MP),xoff(NP),pb(NP),tfunkt
EXTERNAL tfunkt
DATA xoff/4*10./
DATA p/MPNP*0.0/
idum=-64
1  continue
ndim=NP

```

```

do 11 j=2,MP
  p(j,j-1)=1.
11  continue
do 13 i=1,MP
  do 12 j=1,NP
    p(i,j)=p(i,j)+xoff(j)
    x(j)=p(i,j)
12  continue
    y(i)=tfunk(x)

13  continue
  yb=1.e30
  write(*,*) 'Input T, IITER:'
  read(*,*,END=999) temptr,iiter
  ybb=1.e30
  nit=0
  do 14 jiter=1,100
    iter=iiter
    temptr=temptr*0.8
    call amesba(p,y,MP,NP,ndim,pb,yb,FTOL,tfunk,iter,temptr)
    nit=nit+iiter-iter
    if (yb.lt.ybb) then
      ybb=yb
      write(*,'(1x,i6,e10.3,4f11.5,e15.7)')
*      nit,temptr,(pb(j),j=1,NP),yb
    endif
    if (iter.gt.0) goto 80
14  continue
80  write(*,'(1x,a)') 'Vertices of final 3-D simplex and'

  write(*,'(1x,a)') 'function values at the vertices:'
  write(*,'(/3x,a,t11,a,t23,a,t35,a,t45,a/)') 'I',
*  'X(I)', 'Y(I)', 'Z(I)', 'FUNCTION'
  do 15 i=1,MP
    write(*,'(1x,i3,4f12.6,e15.7)') i,(p(i,j),j=1,NP),y(i)
15  continue
  write(*,'(1x,i3,4f12.6,e15.7)') 99,(pb(j),j=1,NP),yb
  goto 1

```

```
999 write(*,*) 'NORMAL COMPLETION'  
STOP  
END
```

```
REAL FUNCTION tfunk(p)  
INTEGER N  
REAL RAD,AUG  
PARAMETER (N=4,RAD=0.3,AUG=2.0)  
INTEGER j  
REAL q,r,sumd,sumr,p(N),wid(N)  
DATA wid /1.,3.,10.,30./  
sumd=0.
```

```
sumr=0.  
do 11 j=1,N  
  q=p(j)*wid(j)  
  r=nint(q)  
  sumr=sumr+q**2  
  sumd=sumd+(q-r)**2  
11 continue  
if (sumd.gt.RAD**2) then  
  tfunk=sumr*(1.+AUG)+1.  
else  
  tfunk=sumr*(1.+AUG*sumd/RAD**2)+1.  
endif  
return  
END
```