

**SIMULASI TEMPERATUR TRANSISI
SUPERKONDUKTOR TBCCO
DENGAN MODEL ASYNNNI**

TESIS

**SAEPUL QODAR
0706171705**



**PROGRAM STUDI ILMU FISIKA
PASCASARJANA FMIPA
UNIVERSITAS INDONESIA
2009**

**SIMULASI TEMPERATUR TRANSISI
SUPERKONDUKTOR TBCCO
DENGAN MODEL ASYNNNI**

TESIS

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Sains**

Oleh

**Saepul Qodar
0706171705**



**PROGRAM STUDI ILMU FISIKA
PASCASARJANA FMIPA
UNIVERSITAS INDONESIA
2009**

HALAMAN PENGESAHAN

Tesis ini diajukan oleh :
Nama : Saepul Qodar
NPM : 0706171705
Program Studi : Ilmu Fisika
Judul Tesis : Simulasi Temperatur Transisi Superkonduktor
TBCCO dengan Model ASYNNNI

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Sains pada Program Studi Ilmu Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. Dedi Suyanto (.....)
Penguji I : Dr. Kebamoto (.....)
Penguji II : Dr. Budhy Kurniawan (.....)
Penguji III : Dr. Efta Yudiarsah (.....)

Ditetapkan di :

Tanggal :

KATA PENGANTAR

Segala puji dan syukur saya panjatkan kepada Allah SWT, karena berkat rahmat dan petunjuk-Nya saya dapat menyelesaikan tesis ini. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad SAW beserta keluarga, sahabat dan segenap umat-Nya sampai akhir zaman. Penelitian ini mengambil topik temperatur transisi superkonduktor dan merupakan lanjutan dari penelitian sebelumnya. Tujuan jangka panjang dari penelitian ini yaitu mencari atau mendapatkan bahan superkonduktor dengan temperatur transisi yang tinggi sehingga dapat digunakan dalam suhu kamar. Pada tesis ini hanya dilakukan simulasi temperatur transisinya. Penyusunan tesis ini dapat diselesaikan atas bantuan dan bimbingan yang sangat berharga dari berbagai pihak. Oleh karena itu penulis menyampaikan ucapan terima kasih kepada yang terhormat:

- (1) Dr. Dedi Suyanto, selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan tesis ini.
- (2) Segenap dosen Departemen Fisika Universitas Indonesia yang telah membuka pemikiran penulis mengenai ilmu fisika.
- (3) Seluruh keluarga atas dorongan, bantuan dan cintanya yang tulus.
- (4) Rekan-rekan mahasiswa S1 dan S2 fisika Universitas Indonesia.
- (5) Semua pihak yang turut berperan dalam penyusunan tesis ini.

Penulis menyadari bahwa penyusunan tesis ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan masukan baik kritik, saran maupun koreksi yang membangun. Semoga tesis ini bermanfaat bagi pengembangan ilmu.

Salemba, 8 Juli 2009

Saepul Qodar

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Saepul Qodar

NPM : 0706171705

Program Studi : Ilmu Fisika

Departemen : Fisika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

Jenis karya : Tesis

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Simulasi Temperatur Transisi Superkonduktor TBCCO dengan Model ASYNNNI

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di :

Pada tanggal :

Yang menyatakan

(Saepul Qodar)

ABSTRAK

Nama : Saepul Qodar
Program Studi : Ilmu Fisika
Judul : Simulasi Temperatur Transisi Superkonduktor TBCCO dengan model ASYNNNI

Penelitian ini akan mensimulasikan temperatur transisi pada superkonduktor TBCCO. Secara umum, superkonduktor TBCCO dapat dituliskan sebagai $Tl_m Ba_2 Ca_{n-1} Cu_n O_{2(n+1)+m}$ dimana m adalah jumlah bidang TlO dan n adalah jumlah bidang CuO. Simulasi temperatur transisi superkonduktor TBCCO dapat dipelajari dengan menggunakan model ASYNNNI. Dalam model ASYNNNI, oksigen di tempatkan pada bidang CuO secara acak. Kandungan oksigen dan jumlah bidang CuO akan mempengaruhi besarnya temperatur transisi. Temperatur transisi tertinggi terjadi pada superkonduktor TBCCO yang mempunyai tiga bidang CuO dengan $m = 2$. Untuk $m = 2$, temperatur transisi tidak dihasilkan dalam simulasi empat bidang CuO, karena bahan ini tidak bersifat superkonduktor.

Kata kunci:

Simulasi, temperatur transisi, model ASYNNNI, bidang CuO.

ABSTRACT

Name : Saepul Qodar
Study Program : Pure Physics
Title : Simulation of Transition Temperature in Superconductors TBCCO with ASYNNNI Model

This research will simulate the transition temperature in superconductors TBCCO. Generally, superconductors TBCCO can be written as $Tl_m Ba_2 Ca_{n-1} Cu_n O_{2(n+1)+m}$ where m is the number of TlO planes and n is the number of CuO planes. Simulation of transition temperature in superconductors TBCCO can be studied using ASYNNNI model. In ASYNNNI model, oxygen is placed in CuO planes randomly. The oxygen content and the number of CuO planes will affect on transition temperature level. The highest transition temperature level is in three CuO planes superconductors TBCCO with $m = 2$. For $m = 2$, transition temperature is not resulted in four CuO planes simulation because the material is not superconductor.

Key words:

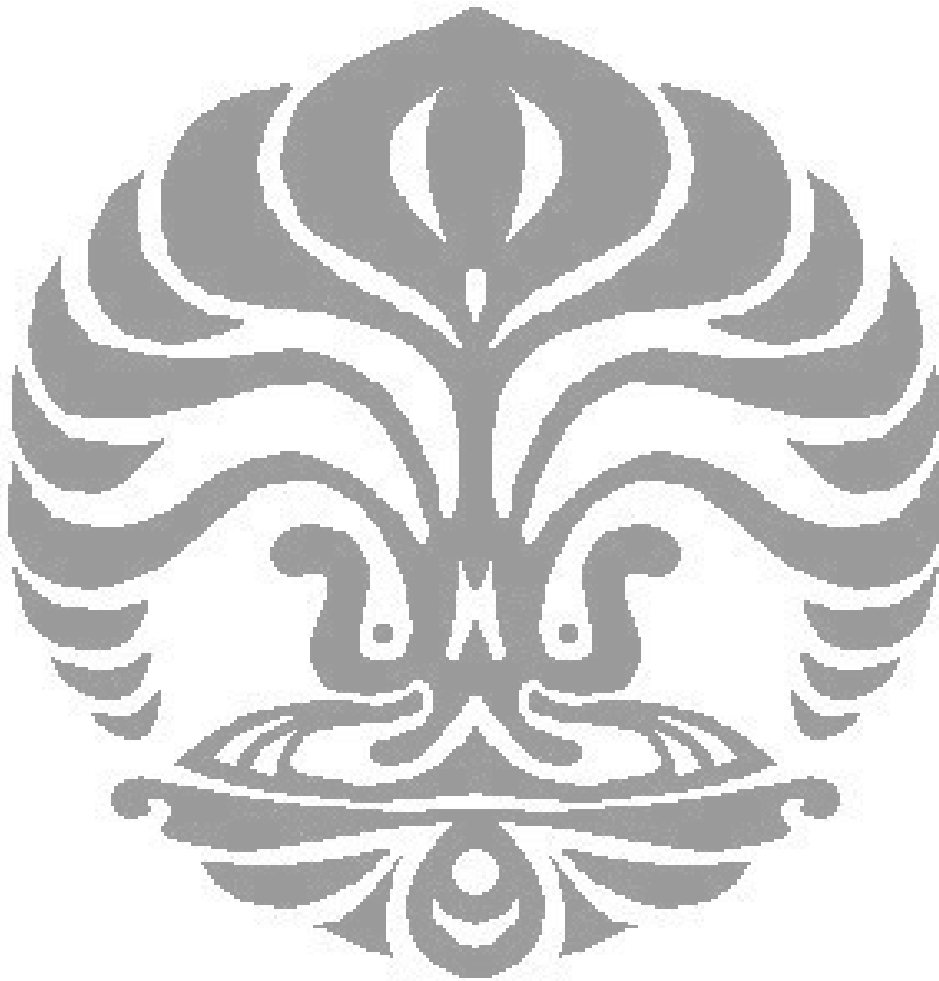
Simulation, transition temperature, ASYNNNI model, CuO planes

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
KATA PENGANTAR	iii
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
ABSTRAK	v
DAFTAR ISI	vi
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Pembatasan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Sistematika Penulisan.....	3
BAB II STRUKTUR KRISTAL TBCCO	4
2.1 Struktur Dasar.....	4
2.2 Bidang CuO Superkonduktor TBCCO.....	5
BAB III MODEL ASYNNNI	8
3.1 Definisi Model ASYNNNI.....	8
3.2 Hamiltonian Interaksi.....	9
3.3 Energi Interaksi antar Bidang.....	10
3.4 Perbandingan Model ASYNNNI dan Model Ising.....	11
BAB IV MONTE CARLO	12
4.1 Metode Monte Carlo.....	12
4.2 Algoritma Metropolis.....	13
4.3 Dinamika Glauber.....	15
BAB V HASIL SIMULASI DAN PEMBAHASAN	17
5.1 Lonjakan Kalor Jenis pada Temperatur Transisi.....	17
5.2 Pengaruh Kandungan Oksigen terhadap Temperatur Transisi.....	34
5.3 Temperatur Transisi terhadap Jumlah Bidang CuO.....	34
BAB VI KESIMPULAN DAN SARAN	37
DAFTAR REFERENSI	39

DAFTAR TABEL

Tabel 2.1.	Parameter kisi untuk senyawa TBCCO.....	5
Tabel 3.1.	Perbandingan model ASYNNNI dan model Ising.....	11



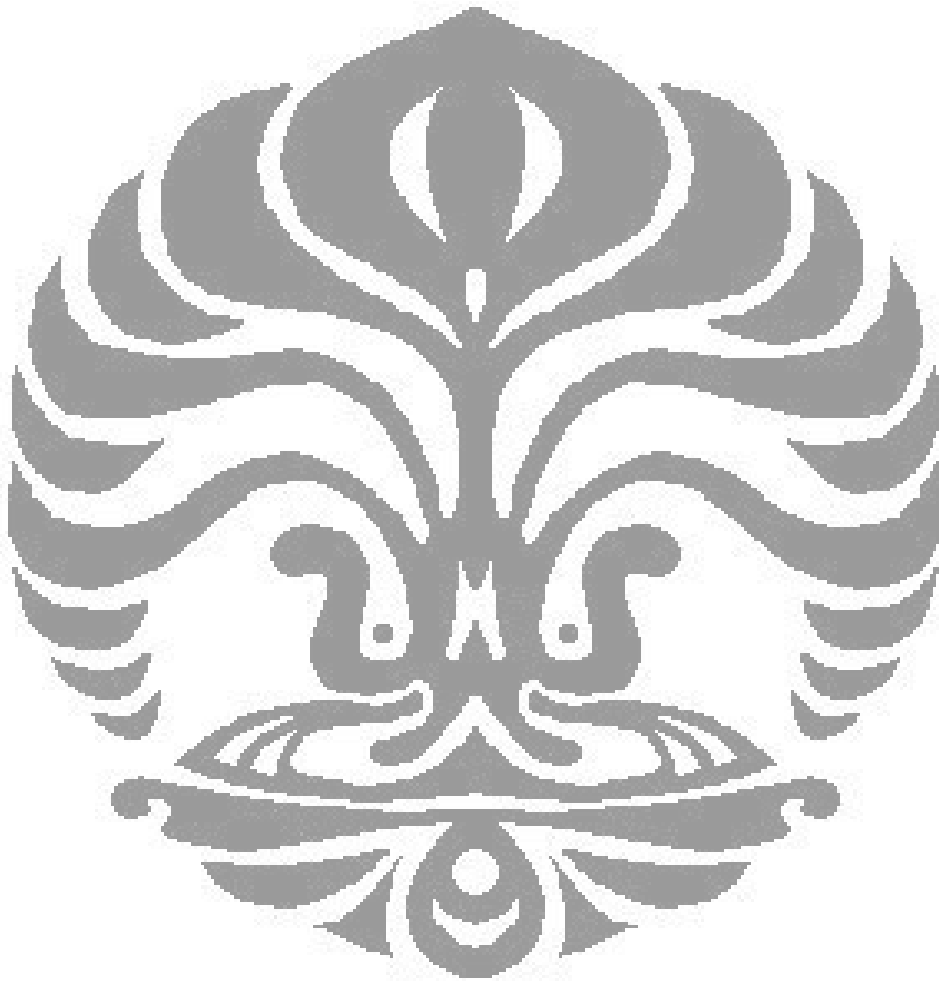
DAFTAR GAMBAR

Gambar 2.1.	Struktur kristal tetragonal TBCCO untuk (a) $Tl_1Ba_2Cu_1O_5$, (b) $Tl_1Ba_2Ca_1Cu_2O_7$, (c) $Tl_1Ba_2Ca_2Cu_3O_9$, (d) $Tl_1Ba_2Ca_3Cu_4O_{11}$, (e) $Tl_2Ba_2Cu_1O_6$, (f) $Tl_2Ba_2Ca_1Cu_2O_8$ dan (g) $Tl_2Ba_2Ca_2Cu_3O_{10}$	4
Gambar 2.2.	Lapisan-lapisan pada struktur kristal superkonduktor $Tl_2Ba_2Ca_{n-1}Cu_nO_{2n+4}$ dengan $n = 1, 2$ dan 3	5
Gambar 2.3.	Sekema lapisan CuO_2	7
Gambar 5.1.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 125 K.....	20
Gambar 5.2.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 122 K.....	20
Gambar 5.3.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 120 K.....	21
Gambar 5.4.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 119 K.....	21
Gambar 5.5.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 118 K.....	22
Gambar 5.6.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 108 K.....	23
Gambar 5.7.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 105 K.....	23
Gambar 5.8.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 100 K.....	24
Gambar 5.9.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 95 K.....	24
Gambar 5.10.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 80 K.....	25
Gambar 5.11.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 70 K.....	26
Gambar 5.12.	Grafik kalor jenis terhadap temperatur dengan temperatur transisi 50 K.....	26

Gambar 5.13. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 40 K.....	27
Gambar 5.14. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO.....	28
Gambar 5.15. Kalor jenis beberapa bahan zat padat sebagai fungsi dari $\frac{T}{\theta_D}$	29
Gambar 5.16. Kalor jenis tembaga sebagai fungsi dari temperatur pada tekanan konstan 1 atm.....	29
Gambar 5.17. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 50 K.....	31
Gambar 5.18. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 65 K.....	31
Gambar 5.19. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 110 K.....	32
Gambar 5.20. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 122 K.....	32
Gambar 5.21. Grafik kalor jenis terhadap temperatur untuk lima bidang CuO dengan temperatur transisi 90 K.....	33
Gambar 5.22. Grafik temperatur transisi terhadap kandungan oksigen.....	34
Gambar 5.23. Grafik hasil simulasi dan eksperimen pada TBCCO.....	35

DAFTAR LAMPIRAN

Lampiran 1.	Grafik Hasil Simulasi.....	43
Lampiran 2.	Flow Chart Program Simulasi.....	48
Lampiran 3.	Program Simulasi.....	51



BAB I PENDAHULUAN

1.1 Latar Belakang

TBCCO mempunyai rumus umum $Tl_mBa_2Ca_{n-1}Cu_nO_{2(n+1)+m-x}$ [1, 2]. Senyawa ini telah ditemukan pada tahun 1988. TBCCO merupakan superkonduktor temperatur tinggi dimana temperatur transisi senyawa ini berkisar antara 20 K sampai 125 K. Superkonduktor TBCCO mempunyai temperatur transisi untuk nilai m dan n yang berbeda-beda. Nilai m menunjukkan jumlah bidang TlO dan nilai n menunjukkan jumlah bidang CuO. Untuk nilai $m = 1$ (TlO *monolayers*) dan $n = 1, 2, 3, 4$ masing-masing diberi kode Tl-1201, Tl-1212, Tl-1223 dan Tl-1234. Untuk nilai $m = 2$ (TlO *bilayers*) dan $n = 1, 2, 3$ masing-masing diberi kode Tl-2201, Tl-2212 dan Tl-2223. Tl-1201 mempunyai temperatur transisi antara 4 K sampai 50 K, Tl-1212 mempunyai temperatur transisi antara 65 K sampai 85 K, Tl-1223 mempunyai temperatur transisi antara 100 K sampai 110 K, Tl-1234 mempunyai temperatur transisi sekitar 122 K, Tl-2201 mempunyai temperatur transisi antara 20 K sampai 80 K, Tl-2212 mempunyai temperatur transisi antara 95 K sampai 108 K dan Tl-2223 mempunyai temperatur transisi antara 118 K sampai 125 K [3].

Temperatur transisi yang sudah didapatkan masih jauh di bawah suhu kamar. Penelitian terus dilakukan supaya didapatkan temperatur transisi pada suhu kamar. Tl-1201 dan Tl-2201 mempunyai satu lapis bidang CuO, Tl-1212 dan Tl-2212 mempunyai dua lapis bidang CuO, Tl-1223 dan Tl-2223 mempunyai tiga lapis bidang CuO, sedangkan Tl-1234 mempunyai empat lapis bidang CuO [4, 5, 6]. Temperatur transisi senyawa ini bertambah dengan bertambahnya lapisan bidang CuO. Karena besarnya temperatur transisi bertambah dengan bertambahnya bidang CuO, maka perlu dilakukan penelitian dengan jumlah bidang CuO yang lebih banyak lagi dengan harapan mendapatkan temperatur transisi yang lebih tinggi.

Tinggi rendahnya temperatur transisi antara keadaan superkonduktor dan non superkonduktor sangat dipengaruhi oleh konsentrasi kandungan oksigen x. konsentrasi kandungan oksigen ini berkisar antara $0 < x < 1$ [7, 8, 9, 10, 11].

Selain dipengaruhi oleh kandungan oksigen, juga dipengaruhi oleh pengaturan penempatan oksigen pada bidang CuO [12]. Pengaturan penempatan oksigen pada bidang CuO dilakukan dengan membuat keadaan acak yang peluangnya masih bisa diterima. Beberapa model yang menggambarkan keadaan ini telah diajukan. Salah satu model yang sukses adalah model ASYNNNI (Asymetri Next Nearest Neighbour Interaction) [13, 14]. Model ASYNNNI ini digunakan dalam membuat program simulasi komputer yang dapat menunjukkan temperatur transisi dari senyawa $Tl_m Ba_2 Ca_{n-1} Cu_n O_{2(n+1)+m-x}$.

Dari hasil simulasi komputer dengan menggunakan model ASYNNNI, dapat digambarkan grafik kalor jenis terhadap temperatur. Sifat umum dari superkonduktor yaitu terjadi lonjakan kalor jenis pada temperatur transisinya. Pada senyawa yang tidak bersifat superkonduktor, maka kalor jenis yang didapatkan tidak mengikuti sifat tersebut.

1.2 Pembatasan Masalah

Dalam penelitian ini, dilakukan pembatasan dengan:

1. Memperhitungkan interaksi antar atom oksigen yang terletak pada bidang CuO dalam arah vertikal di atas dan di bawahnya untuk Tl-1212, Tl-1223, Tl-1234, Tl-2212 dan Tl-2223.
2. Memperhitungkan interaksi antar bidang untuk Tl-1212, Tl-1223, Tl-1234, Tl-2212 dan Tl-2223.
3. Menghitung besaran kalor jenis.

1.3 Tujuan Penelitian

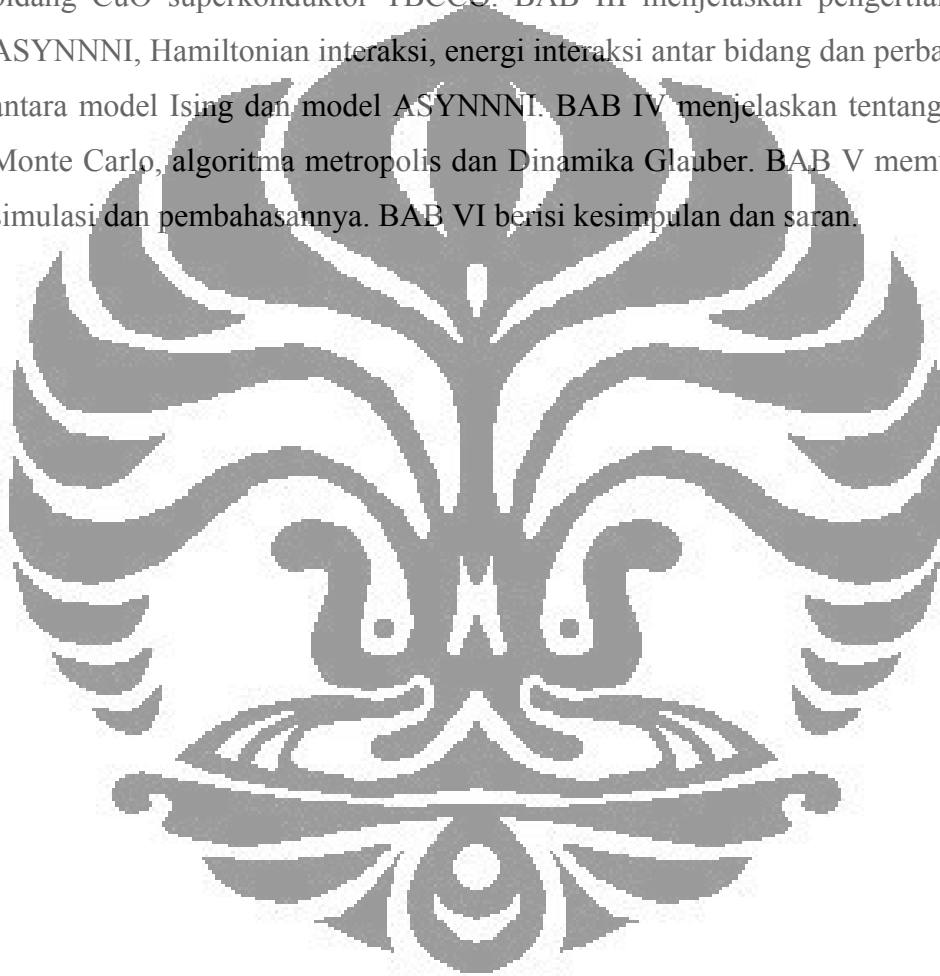
Tujuan penelitian ini adalah menghasilkan simulasi komputer yang dapat memperlihatkan dengan jelas:

1. Kalor jenis sebagai fungsi temperatur.
2. Temperatur transisi untuk senyawa $Tl_m Ba_2 Ca_{n-1} Cu_n O_{2(n+1)+m-x}$ dengan kandungan oksigen yang berbeda-beda.
3. Temperatur transisi tertinggi yang masih bisa didapatkan pada senyawa $Tl_m Ba_2 Ca_{n-1} Cu_n O_{2(n+1)+m-x}$.

4. Membuktikan bahwa untuk senyawa yang mempunyai dua lapis TlO dan terdiri dari empat bidang CuO tidak bersifat superkonduktor.

1.4 Sistematika Penulisan

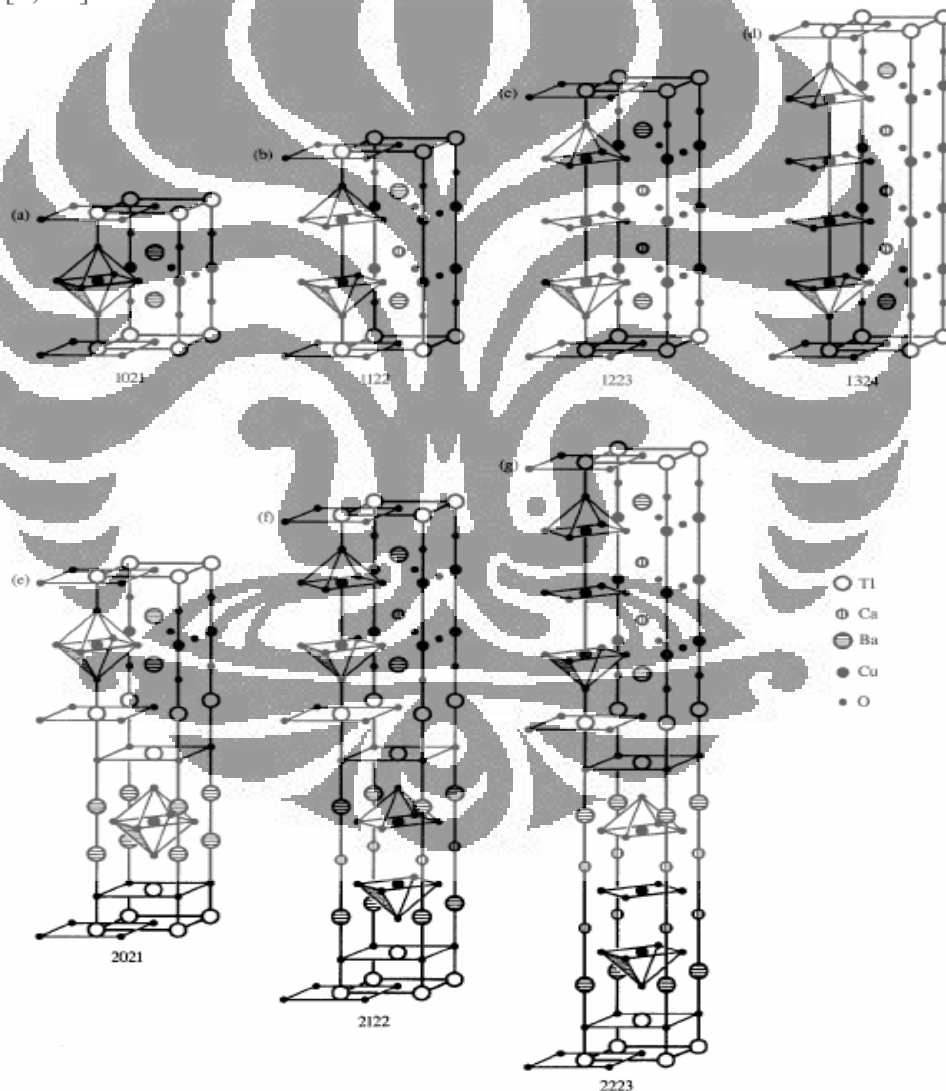
Tesis ini terdiri dari enam BAB. BAB I menjelaskan gambaran ringkas superkonduktor TBCCO, model SYNNNI, pembatasan masalah, tujuan penelitian dan sistematika penulisan. BAB II menjelaskan struktur dasar kristal TBCCO dan bidang CuO superkonduktor TBCCO. BAB III menjelaskan pengertian model ASYNNNI, Hamiltonian interaksi, energi interaksi antar bidang dan perbandingan antara model Ising dan model ASYNNNI. BAB IV menjelaskan tentang metode Monte Carlo, algoritma metropolis dan Dinamika Glauber. BAB V memuat hasil simulasi dan pembahasannya. BAB VI berisi kesimpulan dan saran.



BAB II STRUKTUR KRISTAL TBCCO

2.1 Struktur Dasar

Senyawa $Tl_mBa_2Ca_{n-1}Cu_nO_{2(n+1)+m-x}$ terdiri dari beberapa fasa untuk nilai m dan n yang berbeda [15]. Fasa-fasa tersebut yaitu $Tl_1Ba_2Cu_1O_5$, $Tl_1Ba_2Ca_1Cu_2O_7$, $Tl_1Ba_2Ca_2Cu_3O_9$, $Tl_1Ba_2Ca_3Cu_4O_{11}$, $Tl_2Ba_2Cu_1O_6$, $Tl_2Ba_2Ca_1Cu_2O_8$ dan $Tl_2Ba_2Ca_2Cu_3O_{10}$. Semua fasa ini mempunyai struktur kristal yang sama yaitu tetragonal. Struktur kristal ini digambarkan pada Gambar 2.1 [3, 16].



Gambar 2.1. Struktur kristal tetragonal TBCCO untuk (a) $Tl_1Ba_2Cu_1O_5$, (b) $Tl_1Ba_2Ca_1Cu_2O_7$, (c) $Tl_1Ba_2Ca_2Cu_3O_9$, (d) $Tl_1Ba_2Ca_3Cu_4O_{11}$, (e) $Tl_2Ba_2Cu_1O_6$, (f) $Tl_2Ba_2Ca_1Cu_2O_8$ dan (g) $Tl_2Ba_2Ca_2Cu_3O_{10}$

Struktur TBCCO mempunyai parameter kisi yang berbeda-beda seperti pada Tabel 2.1 [3].

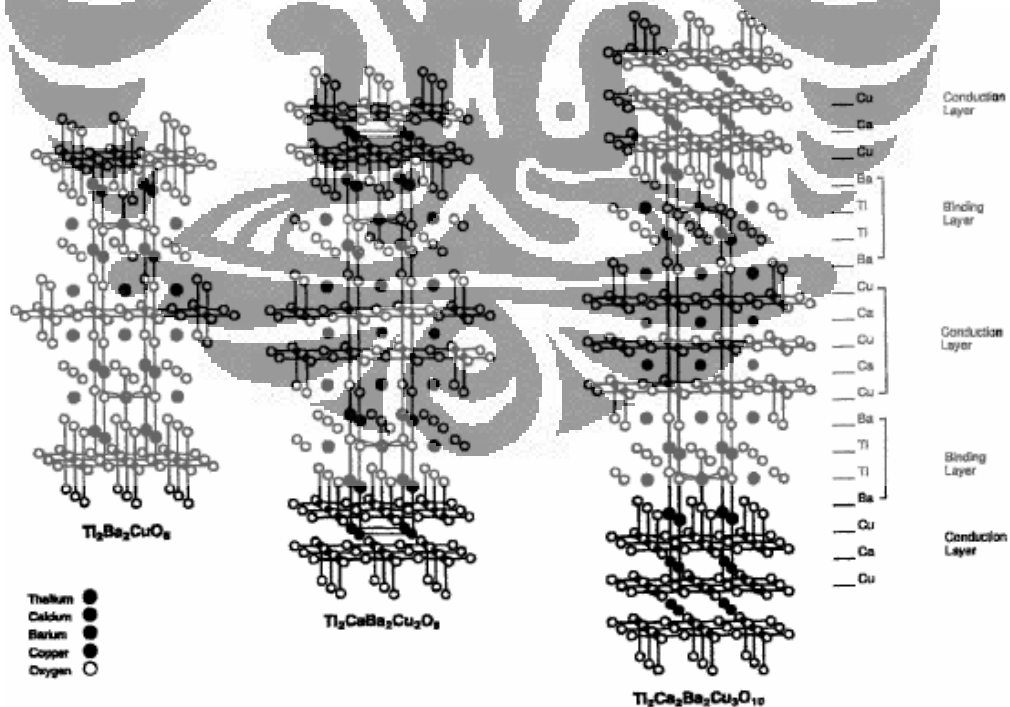
Tabel 2.1. Parameter kisi untuk senyawa TBCCO

No	Kode	Parameter kisi (Å)
1	Tl-1201	$a = 3.869, c = 9,694$
2	Tl-1212	$a = 3.8505, c = 12,728$
3	Tl-1223	$a = 3.8429, c = 15,871$
4	Tl-1234	$a = 3.85, c = 19,1$
5	Tl-2201	$a = 3.8587, c = 23,152$
6	Tl-2212	$a = 3.857, c = 29,39$
7	Tl-2223	$a = 3.822, c = 36,26$

Tl-1201 mempunyai parameter kisi c yang paling kecil sedangkan Tl-2223 mempunyai parameter kisi c yang paling besar.

2.2 Bidang CuO Superkonduktor TBCCO

Senyawa TBCCO mempunyai lapisan bidang-bidang dalam struktur kristalnya. Lapisan bidang-bidang ini digambarkan pada Gambar 2.2 [17].

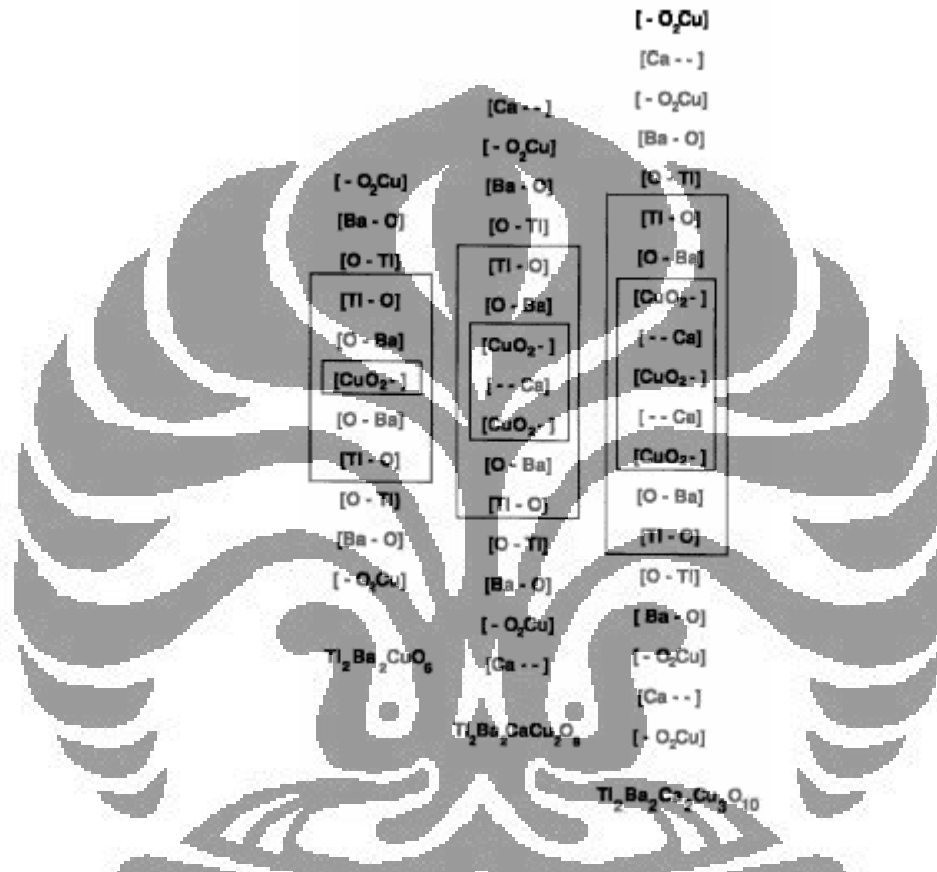


Gambar 2.2. Lapisan-lapisan pada struktur kristal superkonduktor $Tl_2Ba_2Ca_{n-1}Cu_nO_{2n+4}$ dengan $n = 1, 2$ dan 3

Gambar 2.2 menunjukkan posisi atom-atom pada tiga struktur superkonduktor TlO *bilayers* [1, 2, 17]. Struktur yang pertama adalah Tl-2201 terdiri dari satu lapisan $[\text{Cu O}_2 -]$ dan tidak mempunyai lapisan $[- - \text{Ca}]$. Struktur ini mempunyai temperatur transisi yang rendah. Struktur yang kedua yaitu Tl-2212 terdiri dari dua lapisan $[\text{Cu O}_2 -]$ yang dipisahkan oleh lapisan $[- - \text{Ca}]$. Struktur ini mempunyai temperatur transisi yang lebih tinggi dibandingkan dengan struktur yang pertama. Struktur yang ketiga adalah Tl-2223 terdiri dari tiga lapisan $[\text{Cu O}_2 -]$ yang masing-masing dipisahkan oleh lapisan $[- - \text{Ca}]$. Struktur ini mempunyai temperatur transisi yang paling tinggi dibandingkan dengan dua struktur sebelumnya. Posisi atom-atom pada superkonduktor TlO *monolayers* sama halnya seperti pada superkonduktor TlO *bilayers*, hanya berbeda pada jumlah bidang CuO. Pada superkonduktor TlO *bilayers* terdiri dari tiga struktur, sedangkan pada superkonduktor TlO *monolayers* terdiri dari empat struktur. Struktur yang keempat pada superkonduktor TlO *monolayers*, terdiri dari empat lapisan $[\text{Cu O}_2 -]$ yang masing-masing dipisahkan oleh lapisan $[- - \text{Ca}]$. Struktur ini mempunyai temperatur transisi yang paling tinggi dibandingkan dengan tiga struktur sebelumnya tetapi masih dibawah temperatur transisi superkonduktor TlO *bilayers* dengan tiga bidang CuO.

Lapisan konduksi bidang CuO pada superkonduktor TlO *bilayers* dan superkonduktor TlO *monolayers* dapat digambarkan membentuk suatu grup. Grup lapisan konduksi CuO diikat bersama oleh lapisan TlO dan BaO. Gambar 2.3 menunjukkan sekema lapisan bidang CuO pada superkonduktor $\text{Tl}_2\text{Ba}_2\text{Ca}_{n-1}\text{Cu}_n\text{O}_{2n+4}$ [1, 2, 17]. Arus super mengalir pada lapisan konduksi bidang CuO_2 . Tiap-tiap ion tembaga (Cu^{2+}) dikelilingi oleh empat ion oksigen (O^{2-}). Bidang CuO dihubungkan oleh ion kalsium (Ca^{2+}) yang terletak diantara bidang tersebut. Pada keadaan normal di atas temperatur transisi, elektron konduksi dilepaskan oleh atom tembaga. Elektron ini bergerak disekitar bidang CuO sehingga menimbulkan arus listrik. Pada keadaan superkonduktor di bawah temperatur transisi, elektron yang dilepaskan oleh atom tembaga membentuk pasangan elektron (*Cooper pairs*) [18]. Interaksi antara elektron dengan elektron menyebabkan interaksi tolak-menolak yang biasa disebut dengan gaya Coulomb.

Interaksi elektron-ponon [19] akan menyebabkan interaksi tarik-menarik. Pada keadaan superkonduktor, interaksi antara elektron-ponon lebih besar daripada interaksi Coulomb. Sehingga gaya total yang bekerja antar elektron adalah tarik menarik. Keadaan ini menyebabkan pasangan elektron (*Cooper pairs*) yang dapat bergerak bebas tanpa hambatan.



Gambar 2.3. Sekema lapisan CuO_2

Pada Gambar 2.3, lapisan CuO_2 ditandai dengan garis yang membentuk kotak kecil berbentuk persegi panjang. Lapisan yang membentuk satu formula ditandai dengan garis yang membentuk kotak besar berbentuk persegi panjang. Kumpulan bidang $[CuO_2^-]$ dan bidang bayangan $[-O_2Cu]$ berulang sepanjang sumbu c . Ini merupakan lapisan tembaga oksida yang berperan dalam sifat-sifat superkonduktor.

BAB III MODEL ASYNNNI

3.1 Definisi Model ASYNNNI

Beberapa model telah dikemukakan untuk menggambarkan keteraturan atom oksigen pada bidang basal CuO. Model-model tersebut sebagian besar mengasumsikan bahwa hanya pada bidang CuO interaksi oksigen penting. Salah satu model yang sukses adalah model ASYNNNI. Model ini pertama kali dikemukakan oleh Fontaine dkk. ASYNNNI adalah singkatan dari Asymmetric Next-Nearest-Neighbour Interaction. Ada juga yang menyebut model ASYNNNI dengan model Anisotropic Next-Nearest-Neighbour Lattice Gas / model Ising. Penyebutan ini dikarenakan bahwa model ASYNNNI merupakan pengembangan dari model Ising.

Model ASYNNNI menggambarkan atom-atom oksigen yang terletak pada suatu bidang dengan posisi tertentu [20]. Masing-masing oksigen memiliki interaksi dengan atom oksigen tetangganya. Ada 4 jenis atom oksigen tetangga yaitu atom tetangga terdekat (*nearest neighbour*), atom tetangga setelah tetangga terdekat (*next nearest neighbour*) yang dijabatani oleh kekosongan (*vacancie*), atom tetangga setelah tetangga terdekat (*next nearest neighbour*) yang dipisahkan oleh atom Cu dan atom oksigen pada posisi bidang basal yang berdampingan. Oksigen dengan tetangga terdekatnya memiliki interaksi tertentu. Interaksi antara pasangan oksigen digambarkan dengan parameter interaksi V_1 , V_2 , V_3 dan V_4 . V_1 merupakan interaksi tolak-menolak antara oksigen dengan tetangga terdekatnya. Interaksi ini merupakan interaksi terkuat yang menghubungkan dua sub kisi. V_2 merupakan interaksi tarik-menarik antara oksigen dengan tetangga setelah tetangga terdekat yang diselingi atom Cu. Interaksi ini juga menghubungkan dua sub kisi yang sama. V_3 merupakan interaksi tolak-menolak antara oksigen dengan tetangga setelah tetangga terdekat yang diselingi daerah kekosongan. Interaksi ini juga menghubungkan dua sub kisi yang sama. V_4 merupakan interaksi tarik-menarik antara pasangan oksigen yang terletak pada bidang CuO dalam arah vertikal di atas dan di bawahnya.

3.2 Hamiltonian Interaksi

Untuk menghitung energi dari berbagai keadaan, digunakan Hamiltonian interaksi. Perbedaan energi antar keadaan digunakan untuk menentukan keadaan-keadaan yang masih bisa diterima. Hanya keadaan-keadaan yang bisa diterima yang akan diperhitungkan. Hamiltonian model ASYNNNI tiga dimensi adalah [21]

$$H = V_1 \sum_{\langle rr' \rangle}^{NN} \sigma(r)\sigma(r') - V_2 \sum_{\langle rr' \rangle}^{NNCu} \sigma(r)\sigma(r') + V_3 \sum_{\langle rr' \rangle}^{NNV} \sigma(r)\sigma(r') - V_4 \sum_{\langle rr' \rangle}^{NNPI} \sigma(r)\sigma(r') + a \sum_r \sigma(r) \quad (1)$$

Dimana V_1, V_2, V_3 dan V_4 adalah parameter interaksi antara atom oksigen dengan tetangganya yaitu tetangga terdekat dan tetangga terdekat berikutnya. Sedangkan $\sigma(r)$ adalah variabel keterisian, berharga nol jika tidak terisi oksigen dan berharga satu jika terisi oksigen. Suku pertama pada persamaan (1) merupakan penjumlahan seluruh pasangan oksigen antar tetangga terdekatnya, suku kedua merupakan penjumlahan seluruh pasangan oksigen antar tetangga terdekat berikutnya yang dipisahkan oleh atom Cu, suku ketiga merupakan penjumlahan seluruh pasangan oksigen antar tetangga terdekat berikutnya yang dipisahkan oleh kekosongan, suku keempat merupakan penjumlahan seluruh pasangan oksigen antar tetangga terdekat berikutnya yang berada di atasnya pada bidang CuO, sedangkan suku terakhir adalah energi interaksi antar bidang. Pada bidang CuO, karena tiap atom memiliki 4 atom tetangga terdekat dan 2 tetangga setelah tetangga terdekat baik yang dipisahkan oleh atom Cu atau oleh kekosongan, maka harga masing-masing penjumlahan yang meliputi seluruh pasangan oksigen $\langle rr' \rangle$, yang merupakan tetangga terdekat (nearest neighbour) atau tetangga terdekat berikutnya (next nearest neighbour) adalah:

$$\sum_{\langle rr' \rangle}^{NN} \sigma(r)\sigma(r') = 0,1,2,3,4$$

$$\sum_{\langle rr' \rangle}^{NNCu} \sigma(r)\sigma(r') = 0,1,2$$

$$\sum_{\langle rr' \rangle}^{NNV} \sigma(r)\sigma(r') = 0,1,2$$

$$\sum_{\langle rr' \rangle}^{NNPI} \sigma(r)\sigma(r') = 0,1,2,3,4$$

dan $a \sum_r \sigma(r)$ adalah energi interaksi antar bidang dengan a adalah konstanta.

Hamiltonian pada persamaan (1) digunakan untuk menghitung energi dalam U dan kalor jenis C yang masing-masing dirumuskan sebagai berikut [22]

$$U = \frac{1}{N} \langle H \rangle \quad (2)$$

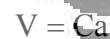
$$C = \frac{N}{(k_B T)^2} (\langle U^2 \rangle - \langle U \rangle^2) \quad (3)$$

3.3 Energi Interaksi Antar Bidang

Energi interaksi antar bidang dapat ditentukan dari teori *Block Layer*. Teori *Block Layer* ini menggambarkan lapisan-lapisan pada senyawa yang saling berinteraksi. Senyawa Tl-2223 terdiri dari beberapa lapis yang saling berinteraksi. Lapisan-lapisan ini dapat dituliskan sebagai berikut:



dimana,



masing-masing komponen mempunyai muatan listrik sebagai berikut



Energi interaksi antar bidang adalah

$$a \sum_r \sigma(r) = k \left(\frac{q_1}{w_1} + \frac{3q_2}{w_2} + \frac{2q_3}{w_3} \right) \text{ eV} \quad (4)$$

Senyawa Tl-2223 mempunyai $w_1 = 15,33 \text{ \AA}$, $w_2 = 3,84 \text{ \AA}$, $w_3 = 3,88 \text{ \AA}$ sehingga didapatkan energi interaksi antar bidang $a \sum_r \sigma(r) = -0,4012 \text{ eV}$.

Sedangkan untuk senyawa Tl-2212 didapatkan energi interaksi antar bidangnya

adalah $a \sum_r \sigma(r) = -1,1061 \text{ eV}$. Untuk senyawa Tl-2201, karena hanya satu bidang CuO, maka energi interaksi antar bidangnya adalah $a \sum_r \sigma(r) = 0 \text{ eV}$.

Dari teori *Block Layer* juga didapatkan bahwa untuk Tl-1212 energi interaksinya $a \sum_r \sigma(r) = -3,8462 \text{ eV}$, untuk Tl-1223 energi interaksinya $a \sum_r \sigma(r) = 1,0746 \text{ eV}$ dan untuk Tl-1234 energi interaksi antar bidangnya adalah $a \sum_r \sigma(r) = 1,9037 \text{ eV}$. Untuk senyawa Tl-1201, karena hanya satu bidang, maka energi interaksi antar bidangnya adalah $a \sum_r \sigma(r) = 0 \text{ eV}$.

3.4 Perbandingan Model ASYNNNI dan Model Ising

Sebagaimana telah diungkapkan sebelumnya, model ASYNNNI merupakan pengembangan dari model Ising. Secara ringkas perbandingan model Ising untuk feromagnetik dengan model ASYNNNI adalah sebagai berikut [13, 21, 22]:

Tabel 3.1 Perbandingan model ASYNNNI dan model Ising.

	Model Ising	Model ASYNNNI
Hamiltonian	$E\{s_i\} = -J \sum_{\langle ij \rangle} s_i s_j - H \sum_{i=1}^N s_i$	$H = V_1 \sum_{\langle rr' \rangle}^{NN} \sigma(r) \sigma(r') - V_2 \sum_{\langle rr' \rangle}^{NNCu} \sigma(r) \sigma(r')$ $+ V_3 \sum_{\langle rr' \rangle}^{NNNI} \sigma(r) \sigma(r') - V_4 \sum_{\langle rr' \rangle}^{NNPI} \sigma(r) \sigma(r')$ $+ \sum \sigma(r)$
Konstanta	Rapat arus (J) dan medan magnet H	Parameter interaksi V_1, V_2, V_3 dan V_4
Variabel	Variabel spin: $s_i = 1$, jika spin up $s_i = -1$, jika spin down	Variabel keterisian posisi: $\sigma(r) = 0$, tidak terisi oksigen $\sigma(r) = 1$, terisi oksigen
Jenis interaksi	Homogen untuk semua tetangga	Berbeda-beda untuk arah yang berbeda

BAB IV MONTE CARLO

4.1 Metode Monte Carlo

Nama Monte Carlo berasal dari nama sebuah kota, tempat kasino terkenal di Monaco. Jadi nama Monte Carlo diasosiasikan dengan perjudian yang menggunakan sistem acak atau random. Suatu sistem yang terdiri dari keadaan acak, dimana sistem tersebut berubah dengan berubahnya waktu dapat disimulasikan dengan menggunakan metode monte carlo. Keadaan sistem yang acak dapat dimunculkan selama berjalannya simulasi. Keadaan acak ini diusahakan mendekati sistem yang nyata sehingga selama simulasi tidak didapatkan keadaan acak yang berulang. Misalnya dalam sebuah kisi kristal, partikel dapat disebarkan secara acak sesuai dengan konsentrasinya dimana jumlah partikel yang akan menempati kisi tersebut tidak melebihi tempat yang sudah tersedia. Dengan demikian maka akan terbentuk keadaan acak antara posisi terisi dan kosong. Keadaan acak ini berubah seiring dengan berjalannya waktu dan perlakuan yang diberikan. Sistem-sistem fisika yang lain yang dapat disimulasikan dengan metode Monte Carlo misalnya gerakan acak partikel dalam fluida (gerak Brown), keadaan acak spin dalam material magnetik, campuran logam dan sebagainya.

Dalam metode Monte Carlo keadaan partikel disebarkan secara acak dengan menggunakan bilangan random yang dihasilkan oleh komputer. Fungsi bilangan random untuk mensimulasikan suatu sistem diusahakan menghasilkan bilangan random yang tidak menghasilkan keadaan acak yang berulang. Untuk sistem yang sangat besar misalnya terdiri dari banyak partikel, diperlukan kemampuan komputer yang memadai. Simulasi akan lebih cepat dan lebih baik jika menggunakan komputer yang mempunyai kapasitas CPU dan memori yang besar. Selain kemampuan komputer yang bagus diperlukan juga pemrograman yang efisien khususnya dalam menggunakan memori. Selain pemrograman yang efisien, algoritma yang digunakan juga mendukung untuk memperoleh hasil yang bagus. Algoritma yang digunakan dalam pemrograman disesuaikan dengan kebutuhan. Dalam simulasi Monte Carlo sekarang ini digunakan algoritma metropolis.

Simulasi Monte Carlo dimulai dengan ukuran sistem yang kecil, kemudian tahap selanjutnya meningkatkan ukuran sistem sehingga dihasilkan simulasi yang sesuai dengan teori dan eksperimen. Jika simulasi ini telah berjalan dengan baik, maka dapat digunakan untuk mensimulasikan keadaan yang lain yang lebih kompleks dengan parameter yang lebih banyak. Keadaan ini biasanya digunakan untuk memprediksi keadaan yang akan terjadi dimana belum ada teori yang menjelaskan dan eksperimen yang dilakukan.

Ciri dari simulasi Monte Carlo adalah menggunakan bilangan random untuk mensimulasi keadaan acak. Bilangan random yang digunakan dalam probabilitas adalah bilangan riil dengan jangkauan antara 0 sampai mendekati 1 ($0 \leq \text{bilangan random} < 1$). Bilangan ini dihasilkan oleh komputer dengan cara pengocokan (pengolahan angka), sehingga dihasilkan keluaran berupa bilangan riil yang berkisar antara 0 sampai mendekati 1 secara acak.

Secara teori banyaknya bilangan random (bilangan riil) antara 0 sampai mendekati 1 adalah tak terhingga, tetapi dalam program komputer, banyaknya bilangan random terbatas. Misalnya bila ada 1000 bilangan random, maka bilangan random tersebut berkisar dari 0.000 sampai 0.999. Salah satu bilangan random ini akan dimunculkan oleh komputer secara sembarang, seperti halnya dadu yang dilempar [23].

4.2 Algoritma Metropolis

Metode metropolis bekerja berdasarkan peluang transisi dari suatu keadaan ke keadaan berikutnya dimana peluang ini bergantung pada perbedaan energi antara keadaan awal dan keadaan berikutnya. Peluang keadaan berikutnya ini bergantung pada keadaan sebelumnya. Energi dari berbagai keadaan yang masih bisa diterima kemudian dirata-ratakan. Banyak faktor yang mempengaruhi dalam perhitungan dengan menggunakan algoritma metropolis diantaranya temperatur, ukuran kisi, syarat batas dan sebagainya. Tahap awal biasanya dimulai dengan sistem yang sederhana, kemudian pada tahap selanjutnya dimasukkan parameter-parameter yang lebih kompleks sehingga didapatkan hasil yang bagus. Model sistem fisika yang menggunakan algoritma metropolis diantaranya adalah model Ising dan model SYNNNI.

Dalam model ASYNNI, bila dalam sistem terdapat N buah posisi oksigen terisi atau kosong, maka jumlah keadaan yang mungkin adalah 2^N keadaan. Menurut distribusi Boltzman, peluang untuk menemukan sebuah sistem dalam keadaan s dengan energi E_s dimana sistem tersebut kontak termal dengan sebuah reservoir besar pada temperatur T adalah

$$P = \frac{\exp\left(-\frac{E_s}{k_b T}\right)}{Z} \quad (5)$$

dimana

$$Z = \sum_s \exp\left(-\frac{E_s}{k_b T}\right) \quad (6)$$

Z adalah total seluruh keadaan atau fungsi partisi. Untuk harga N yang besar, Z akan sulit dihitung. Sebagai contoh untuk harga $N = 10 \times 10$, maka ada 2^{100} keadaan. Untuk mengurangi kesulitan ini, maka diperlukan rumusan yang akan menghitung sebagian keadaan saja tetapi sudah mewakili semua keadaan. Rumusan ini disebut metode Monte Carlo.

Keadaan yang besar dapat digantikan dengan keadaan yang lebih kecil, tetapi dapat mewakili semua keadaan. Keadaan ini didapatkan dengan menggunakan algoritma metropolis. Algoritma metropolis ini akan membentuk keadaan jumlah sampel yang menggantikan 2^N keadaan. Pembentukan keadaan dilakukan dengan mengubah posisi atom oksigen terisi atau kosong secara acak.

Dalam model ASYNNI, konstanta posisi oksigen terisi atau kosong disebarkan secara sembarang dengan menggunakan bilangan random. Walaupun oksigen disebarkan secara sembarang, tetapi hanya keadaan yang dapat diterima saja yang akan diperhitungkan. Keadaan ini dapat diketahui dengan menghitung perbedaan energi ataupun peluangnya. Banyaknya keadaan yang mungkin sangat bergantung pada posisi oksigen terisi atau kosong. Keadaan jumlah posisi terisi atau kosong ini sesuai dengan kandungan oksigen. Sedangkan peluangnya dapat dihitung dengan menggunakan persamaan probabilitas $p = \exp\left(-\frac{\Delta E}{k_b T}\right)$

Kemudian dilakukan penyapuan dengan menggunakan algoritma metropolis.

Secara lengkap langkah-langkah algoritma metropolis adalah [24]:

1. Dilakukan langkah coba untuk mengubah keadaan awal S_1 dengan cara membuat keadaan baru S_2 yang berbeda posisi terisi dan kosong.
2. Dihitung $\Delta E = E_2 - E_1$.
3. Jika $\Delta E \leq 0$, maka langkah coba diterima dan dilanjutkan kelangkah 8.
4. Jika $\Delta E > 0$, maka dihitung probabilitas $p = \exp\left(-\frac{\Delta E}{k_b T}\right)$.
5. Kemudian dimunculkan bilangan random r oleh komputer.
6. Jika $r \leq p$, maka dianggap peluang keadaan dari langkah coba lebih besar daripada keadaan sebelumnya, sehingga langkah coba diterima dan dilanjutkan kelangkah 8. Jika $r > p$, maka langkah coba ditolak.
7. Diulangi langkah 2 sampai 6 untuk kemungkinan keadaan berikutnya, sampai seluruh keadaan dijelajahi.
8. Selanjutnya dihitung kuantitas fisika yang diinginkan (energi dan kalor jenis).
9. Diulangi langkah 2 sampai 8 sehingga jumlah sampel mencukupi.
10. Kemudian semua kuantitas sampel diperhitungkan.

4.3 Dinamika Glauber

Keadaan-keadaan acak yang masih dapat diterima didasarkan pada perubahan energi antara keadaan awal dan keadaan berikutnya, dimana perubahan energi ini akan berpengaruh juga pada peluang yang akan dihitung. Hasil perhitungan peluang selanjutnya dibandingkan dengan bilangan random yang muncul. Jika peluang yang dihitung lebih besar dari bilangan random yang muncul, maka keadaan berikutnya masih dapat diterima. Sebaliknya jika peluang yang dihitung lebih kecil dibandingkan dengan bilangan random yang muncul, maka keadaan berikutnya ditolak. Peluang yang dihitung dengan persamaan

$p = \exp\left(-\frac{\Delta E}{k_b T}\right)$ hanya berlaku untuk temperatur yang rendah. Peluang ini

didapatkan dari sistem klasik. Pada temperatur rendah, sistem akan mengikuti sistem klasik sehingga digunakan probabilitas untuk sistem klasik. Sedangkan pada temperatur yang tinggi, nilai yang didapatkan selalu mendekati 1 sehingga

peluang yang dihitung nilainya selalu lebih besar dibandingkan dengan bilangan random yang muncul. Keadaan ini menyebabkan semua keadaan acak akan selalu diterima. Untuk mengatasi keadaan ini digunakan persamaan peluang yang lebih umum. Peluang ini tentunya harus berlaku untuk temperatur tinggi dan temperatur rendah. Peluang yang berlaku untuk semua nilai temperatur dinamakan probabilitas Glauber.

Jika pada temperatur tinggi peluang yang digunakannya adalah peluang untuk sistem klasik, maka algoritma metropolis akan mem-*flip* semua keadaan pada setiap sapuan karena nilai probabilitasnya selalu mendekati 1. keadaan ini akan menyebabkan harga bilangan random selalu lebih rendah dari probabilitasnya. Untuk mengatasi keadaan ini maka digunakan probabilitas Glauber.[22]

$$p = \frac{1}{2} \left(1 - \tanh \left(\frac{\Delta H}{2k_b T} \right) \right) \quad (7)$$

dimana p adalah peluang, ΔH adalah perubahan energi antara keadaan awal dan keadaan berikutnya, k_b adalah konstanta Boltzman dan T adalah temperatur yang digunakan. Temperatur ini besarnya diubah-ubah sesuai dengan perlakuan yang diinginkan. Langkah-langkah yang lain untuk mengatasi semua nilai temperatur sama dengan algoritma metropolis. Jadi hanya berbeda pada penghitungan peluangnya.

BAB V HASIL SIMULASI DAN PEMBAHASAN

5.1 Lonjakan Kalor Jenis pada Temperatur Transisi

Sifat umum dari superkonduktor adalah terjadinya lonjakan kalor jenis pada temperatur transisi T_c [25, 26, 27]. Kalor jenis ini dapat dihitung pada keadaan volume tetap yang merupakan turunan parsial dari energi dalam terhadap temperatur $C_V = \left(\frac{\partial U}{\partial T}\right)_V$. Sedangkan pada pengukuran, Kalor jenis ini diukur pada keadaan tekanan tetap atau merupakan turunan parsial dari energi dalam terhadap temperatur $C_p = \left(\frac{\partial U}{\partial T}\right)_p$. Pada temperatur rendah, Perbedaan antara C_V dan C_p sangat kecil sehingga dapat diabaikan [28].

Perubahan fasa antara keadaan superkonduktor dan non superkonduktor terjadi pada temperatur transisi. Pada keadaan superkonduktor, mempunyai fasa struktur orthorombik dan pada keadaan non superkonduktor mempunyai fasa struktur tetragonal [29, 30, 31, 32, 33].

Transisi dari keadaan normal ke keadaan superkonduktor merupakan transisi fase orde kedua. Perbedaan energi bebas Helmholtz antara keadaan superkonduktor dan keadaan normal adalah $F_s - F_n = -\frac{H_c^2}{8\pi}$ [13] dimana H_c adalah medan magnet kritis. Pada transisi fasa orde kedua, turunan pertama dari energi bebas Gibbs selalu kontinu sedangkan turunan keduanya diskontinyu. Turunan pertama dari energi bebas Helmholtz akan menghasilkan entropi $S = -\frac{\partial F}{\partial T}$, sehingga didapatkan perbedaan entropi antara keadaan superkonduktor

dan keadaan normal $S_s - S_n = \frac{H_c}{4\pi} \frac{\partial H_c}{\partial T}$. Dimana $\frac{\partial H_c}{\partial T}$ adalah kemiringan dari

kurva temperatur transisi. Dari data yang ada didapatkan bahwa $\frac{\partial H_c}{\partial T} < 0$ pada $0 < T < T_c$, artinya keadaan superkonduktor lebih teratur dari pada keadaan normal. Oleh karena itu, transisi superkonduktor dapat dikatakan sebagai transisi

teratur - tidak teratur. Karena $\frac{\partial H_c}{\partial T} < 0$, maka $S_n - S_s > 0$. Keadaan ini menyatakan bahwa ada kalor laten $Q = T(S_n - S_s) > 0$ yang harus diserap oleh superkonduktor dalam bertransformasi dari keadaan superkonduktor ke keadaan normal. Pada saat $T = T_c$, maka $H_c = 0$ sehingga $S_n - S_s = 0$ dan $Q = T(S_n - S_s) = 0$. Pada $T = 0$, maka kalor laten $Q = T(S_n - S_s) = 0$. Keadaan ini ($Q = 0$) merupakan transisi fase orde kedua tidak hanya terjadi pada $T = T_c$ tetapi juga pada $T = 0$.

Kalor jenis C didefinisikan sebagai $C = T \left(\frac{\partial S}{\partial T} \right)$ sehingga didapatkan bahwa perbedaan antara kalor jenis untuk keadaan superkonduktor dan keadaan normal adalah $C_s - C_n = \frac{T}{4\pi} \left[\left(\frac{\partial H_c}{\partial T} \right)^2 + H_c \frac{\partial^2 H_c}{\partial T^2} \right]$. Pada $T = T_c$ maka didapatkan

diskontinuitas kalor jenis $\Delta C = C_s - C_n = \frac{T_c}{4\pi} \left(\frac{\partial H_c}{\partial T} \right)_{T_c}^2$. ΔC merupakan tingginya lonjakan kalor jenis superkonduktor. Dari teori BCS (Bardeen, Cooper, and Schrieffer) untuk superkonduktor konvensional didapatkan bahwa perbandingan antara diskontinuitas kalor jenis ΔC dengan kalor jenis pada keadaan normal C_n

adalah $\beta \equiv \frac{\Delta C}{C_n} \Big|_{T_c} = \frac{C_s - C_n}{C_n} \Big|_{T_c} = 1.43$ [13]. Dari hasil eksperimen untuk Thallium didapatkan bahwa $\beta = 1.15$ [28].

Hasil simulasi temperatur transisi senyawa TBCCO telah menunjukkan adanya beberapa temperatur transisi. Temperatur-temperatur transisi ini didapatkan dengan mengubah-ubah kandungan oksigennya. Selain kandungan oksigen, temperatur transisi juga berubah dengan berubahnya jumlah bidang CuO. Besarnya temperatur transisi dapat dilihat pada grafik kalor jenis terhadap temperatur. Pada grafik dapat dilihat bahwa diskontinuitas kalor jenis didapatkan ketika terjadi lonjakan kalor jenisnya. Besarnya temperatur transisi didapatkan ketika terjadi lonjakan kalor jenis pada temperatur tersebut.

Ukuran kisi dalam arah x , arah y dan arah z adalah N_x , N_y dan N_z . Parameter interaksinya adalah V_1 , V_2 , V_3 dan V_4 . Dalam program nsweep adalah banyaknya pemindahan tiap atom oksigen secara acak (sweep) dan ngroup adalah banyaknya atom oksigen yang dipindahkan secara acak (grup). Kandungan oksigen adalah x (didalam program variabelnya adalah Besarx). Untuk kandungan oksigen 90%, maka di dalam program dimasukkan Besarx = 1. ini menunjukkan bahwa $x = 0,1$. Grafik untuk senyawa yang lain dapat dilihat pada lampiran. Variabel dalam program untuk perubahan temperatur adalah delta T (ΔT). Gambar 5.1 dan 5.2 adalah grafik hasil simulasi untuk senyawa TI - 2223 dengan kandungan oksigen 90% dan 80%.

Masukan:

$$N_x = 70, N_y = 70, N_z = 20$$

$$V_1 = 0.190 \text{ eV}$$

$$V_2 = 0.136 \text{ eV}$$

$$V_3 = 0.054 \text{ eV}$$

$$V_4 = 0.020 \text{ eV}$$

$$\text{grup} = 30$$

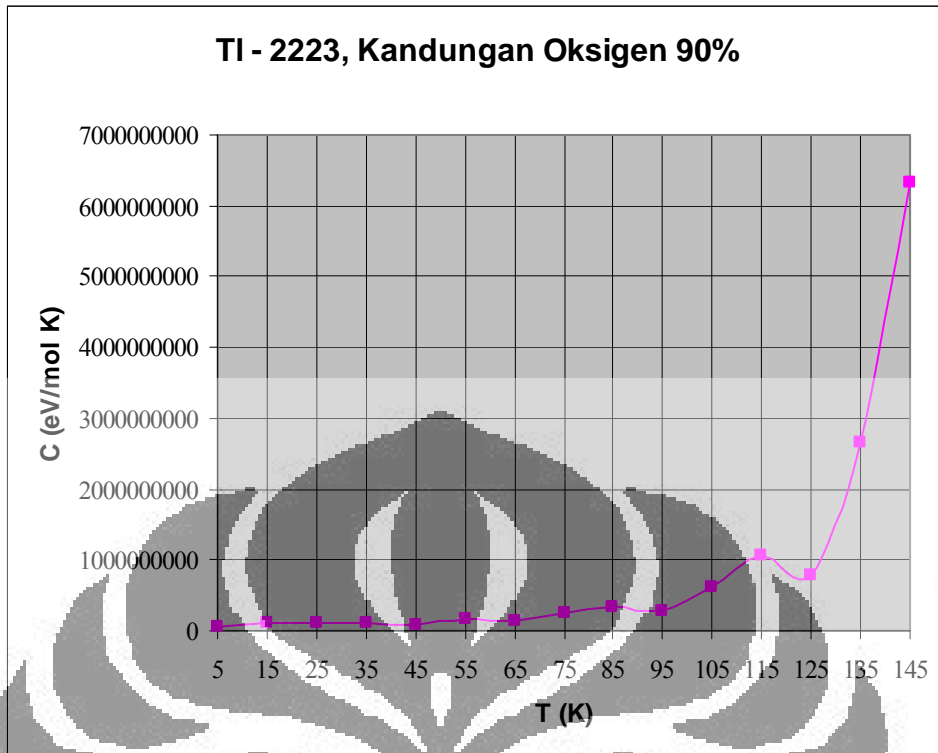
$$\text{sweep} = 30$$

$$kb = 8.125e-05$$

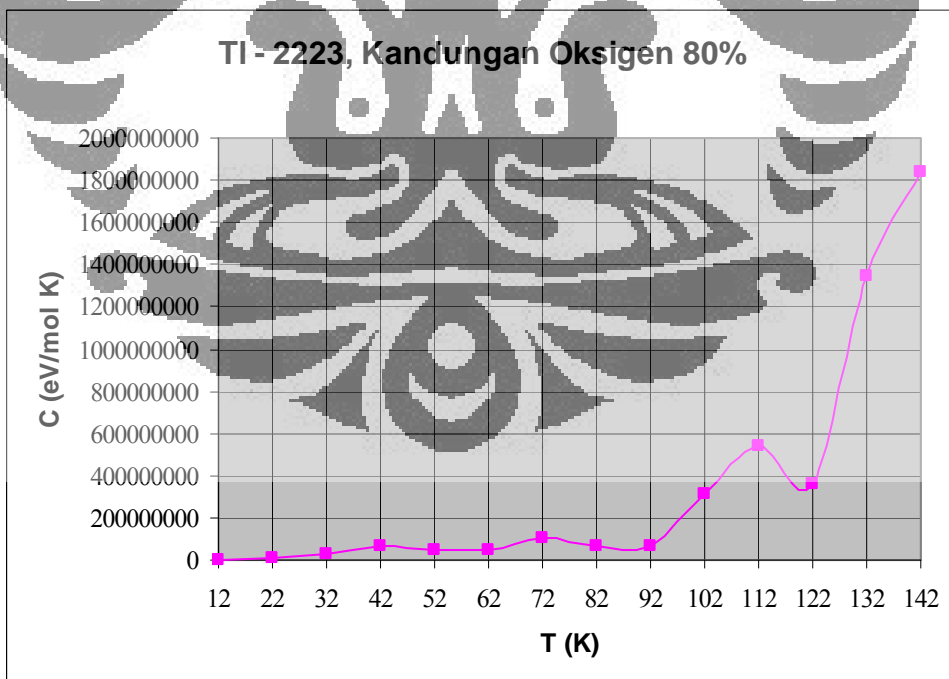
$$x = 0,1$$

$$\Delta T = 10 \text{ K}$$

Parameter masukkan untuk N_x , N_y , N_z , grup dan sweep disesuaikan dengan kemampuan komputer dalam perhitungan. Jika ukuran sistem yang digunakan melebihi nilai tersebut, maka program komputer tidak dapat dijalankan. Pada penelitian ini digunakan komputer dengan procesor 1,4 GB dan memory sebesar 256 MB. Parameter masukkan untuk V_1 , V_2 , V_3 dan V_4 dipilih sehingga didapatkan keluaran yang sesuai dengan eksperimen. Harga x disesuaikan dengan kandungan oksigen yang digunakan dan perubahan suhu ΔT pada penelitian ini adalah 10 K. Jika lebih kecil atau lebih besar dari 10 K, maka akan susah menentukan lonjakkkan kalor jenisnya. Jika lebih kecil dari 10 K, maka akan banyak lonjakkkan yang didapatkan sedangkan jika lebih besar akan susah mendapatkan lonjakkannya.

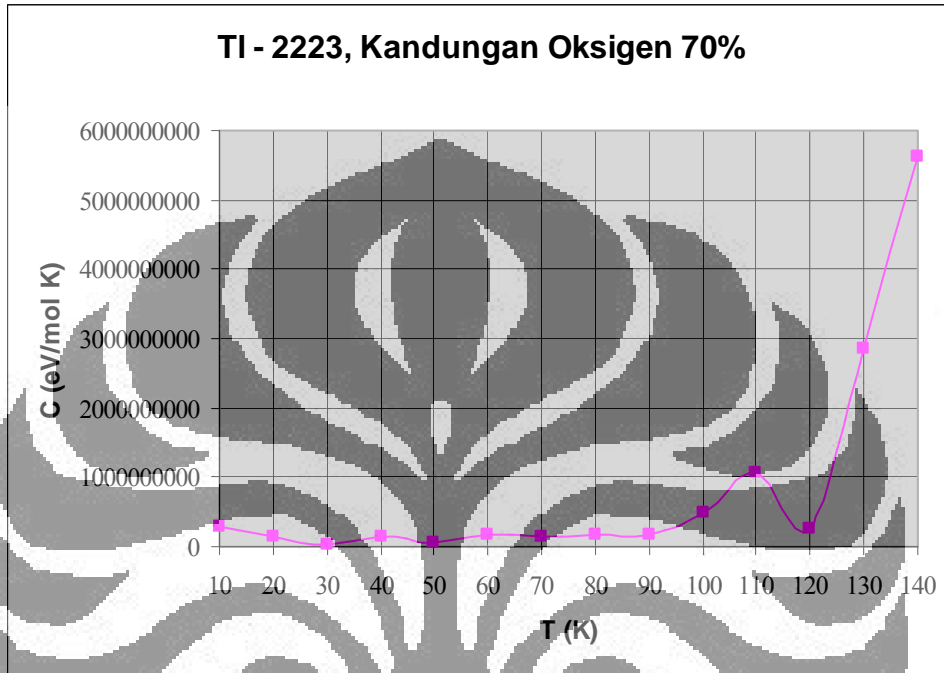


Gambar 5.1. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 125 K

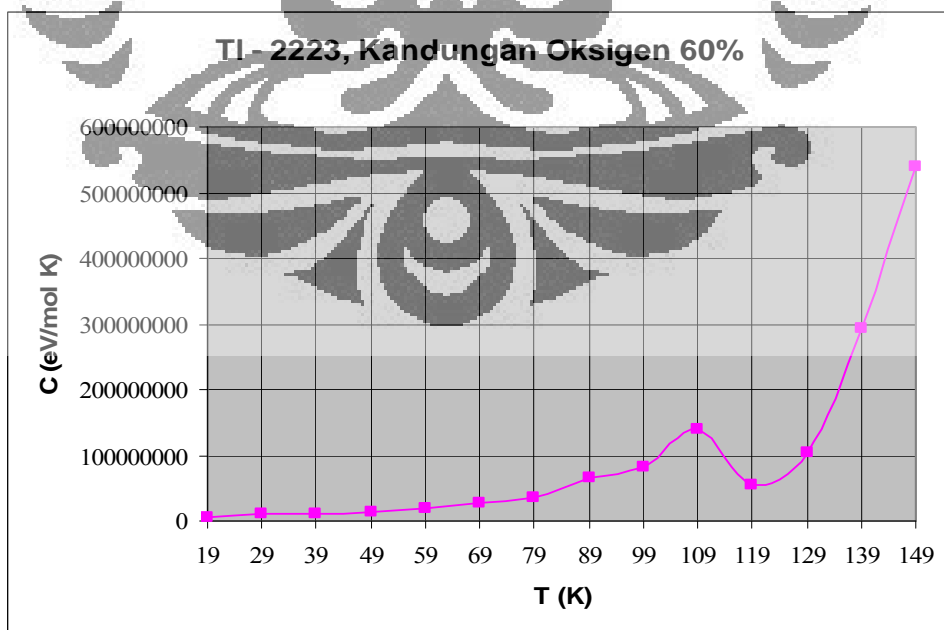


Gambar 5.2. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 122 K

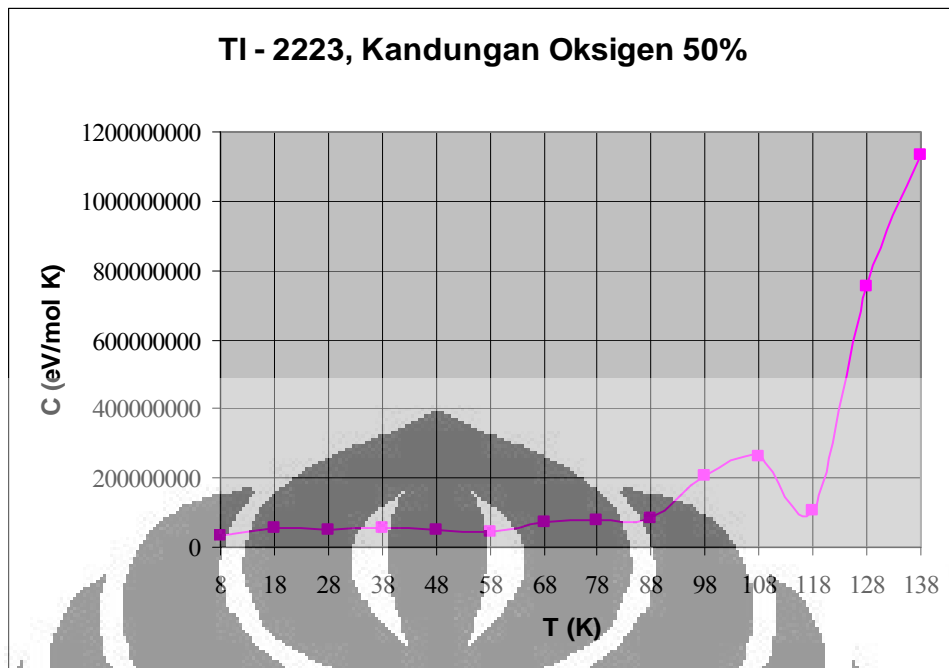
Untuk TI-2223 dengan kandungan oksigen 80%, masukannya sama seperti pada TI-2223 dengan kandungan oksigen 90% hanya berbeda pada nilai x . Untuk kandungan oksigen 80%, maka dalam program $Besarx=2$. ini menunjukkan bahwa $x=0,2$. Untuk kandungan oksigen 70%, 60% dan 50% maka $x=0,3$, $x=0,4$ dan $x=0,5$.



Gambar 5.3. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 120 K



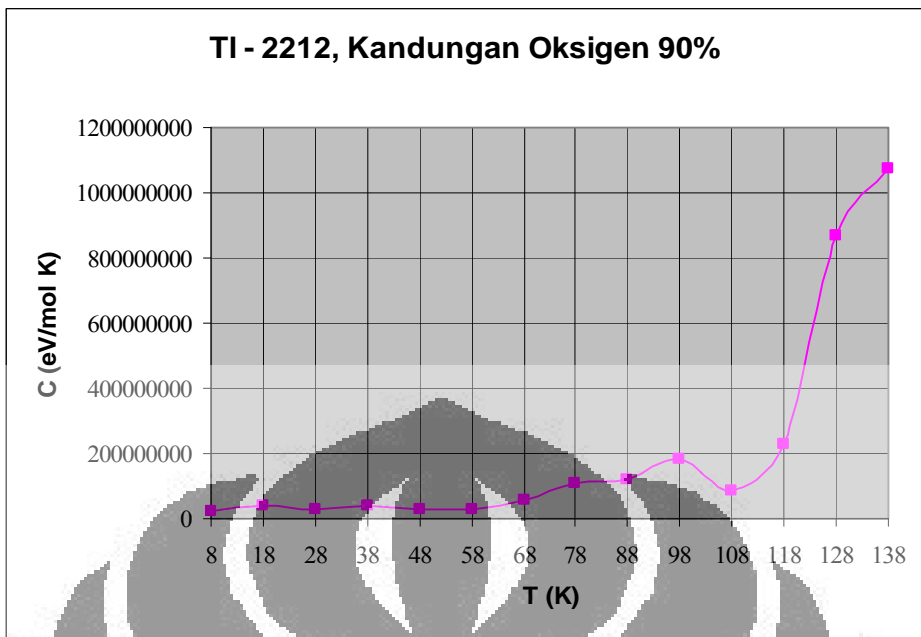
Gambar 5.4. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 119 K



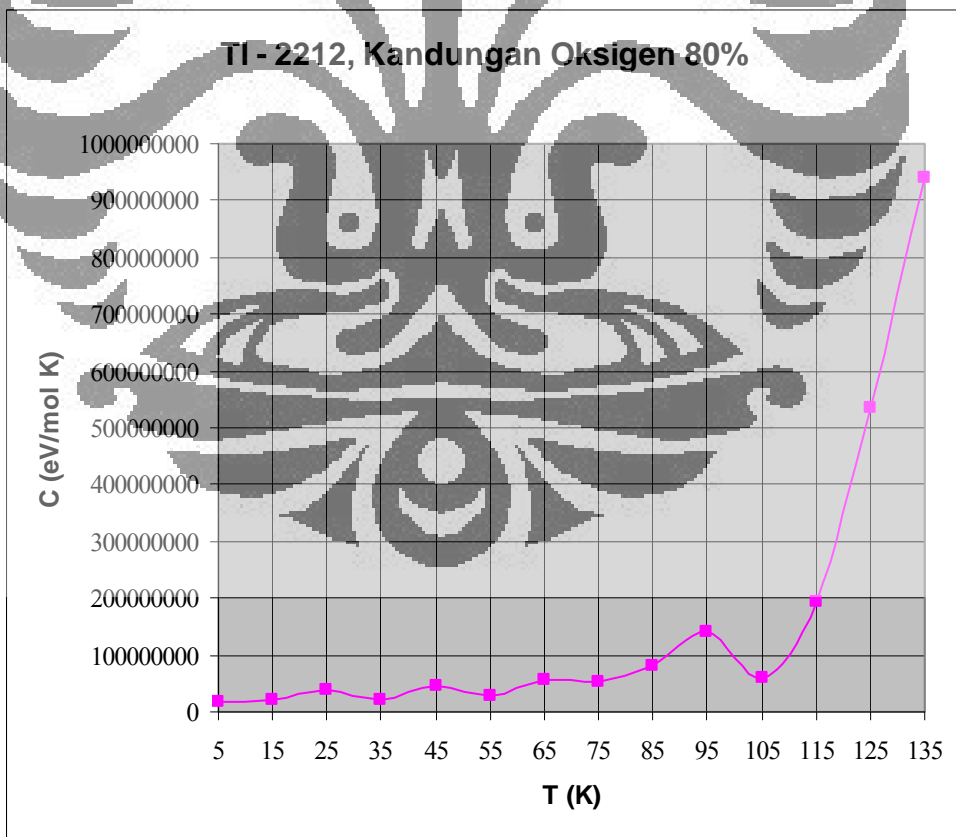
Gambar 5.5. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 118 K

Hasil simulasi untuk TI-2223 menunjukkan ada lima temperatur transisi. Untuk kandungan oksigen 90 %, lonjakan kalor jenis terjadi pada suhu 125 K sehingga temperatur transisinya adalah 125 K. Untuk kandungan oksigen 80 %, 70 %, 60 % dan 50 % temperatur transisinya masing-masing adalah 122 K, 120 K, 119 K dan 118 K.

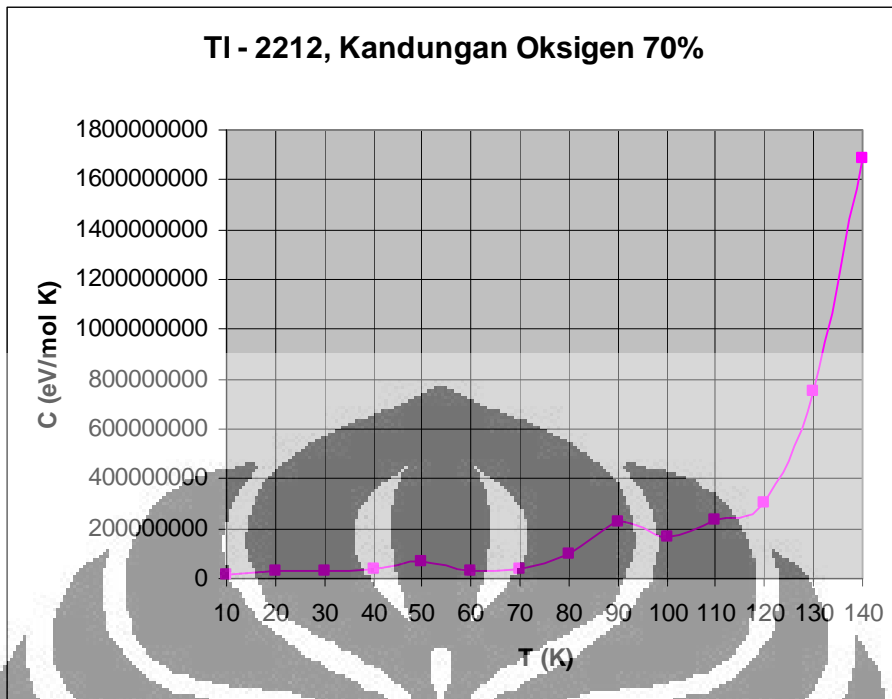
Simulasi pada TI-2212, masukannya sama seperti pada simulasi TI-2223. Hasil simulasi untuk TI-2212 menunjukkan ada empat temperatur transisi. Untuk kandungan oksigen 90 %, lonjakan kalor jenis terjadi pada suhu 108 K sehingga temperatur transisinya adalah 108 K. Untuk kandungan oksigen 80 %, 70 % dan 60 % temperatur transisinya masing-masing adalah 105 K, 100 K dan 95 K. Jadi dengan kandungan oksigen yang berbeda-beda, maka didapatkan temperatur transisi yang berbeda juga. Walaupun temperatur transisi yang didapatkan berbeda tetapi besarnya tetap dalam rentang temperatur transisi senyawa TI-2212. Gambar 5.6 sampai Gambar 5.9 menunjukkan grafik lonjakan kalor jenis terhadap temperatur untuk senyawa TI-2212 dengan kandungan oksigen yang berbeda-beda yaitu 90%, 80%, 70% dan 60%. Masukan untuk program simulasi sama seperti pada TI-2223.



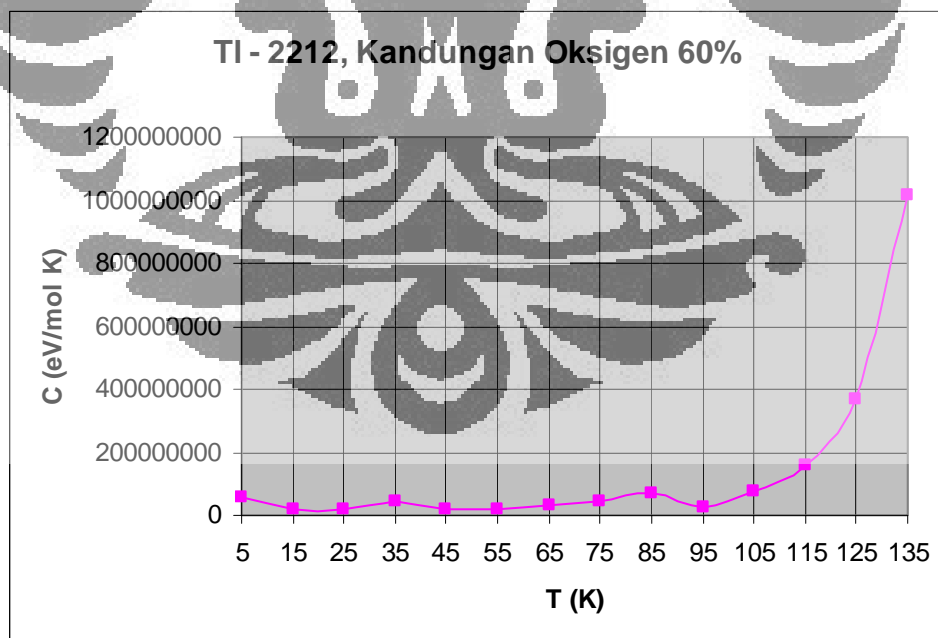
Gambar 5.6. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 108 K



Gambar 5.7. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 105 K

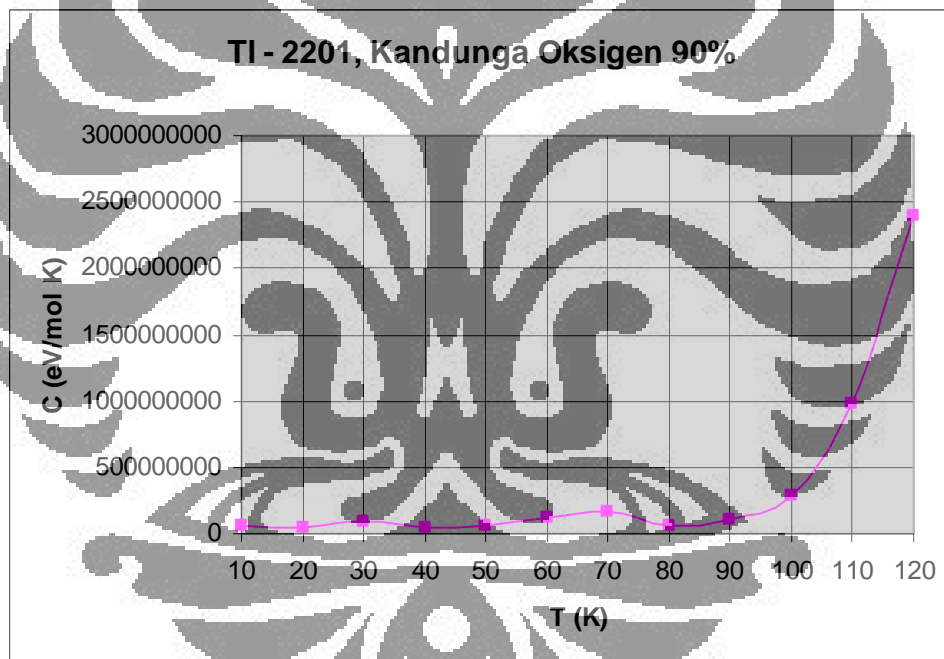


Gambar 5.8. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 100 K

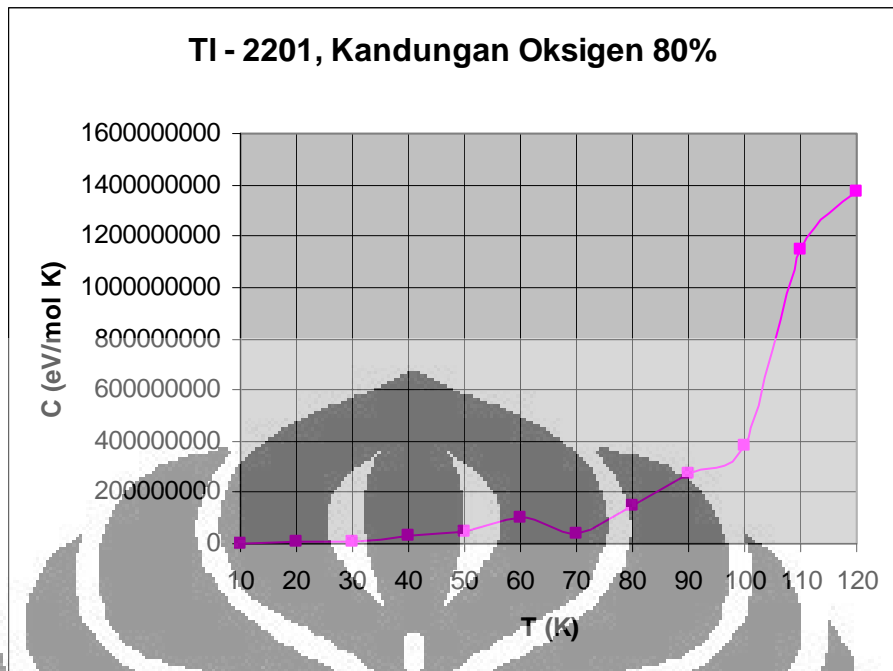


Gambar 5.9. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 95 K

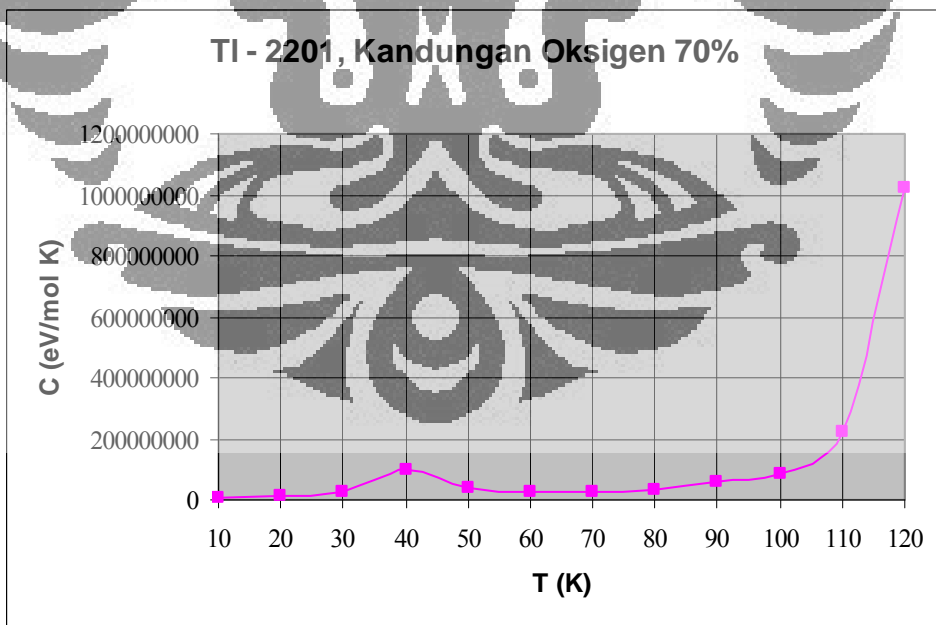
Simulasi pada TI-2201, masukannya sama seperti pada simulasi TI-2223 tetapi $v_4 = 0$. v_4 bernilai 0 karena pada TI-2201 hanya mempunyai satu bidang CuO sehingga tidak ada interaksi oksigen antar bidang tersebut. Hasil simulasi untuk TI-2201 menunjukkan ada empat temperatur transisi. Untuk kandungan oksigen 90 %, lonjakan kalor jenis terjadi pada suhu 80 K sehingga temperatur transisinya adalah 80 K. Untuk kandungan oksigen 80 %, 70 % dan 60 % temperatur transisinya masing-masing adalah 70 K, 50 K dan 40 K. Gambar 5.10 sampai Gambar 5.13 menunjukkan grafik lonjakan kalor jenis untuk senyawa TI-2201. Masukannya sama seperti pada TI-2223 tetapi nilai $V_4 = 0$, karena terdiri dari satu bidang CuO sehingga tidak ada interaksi antar atom oksigen dalam arah vertikal.



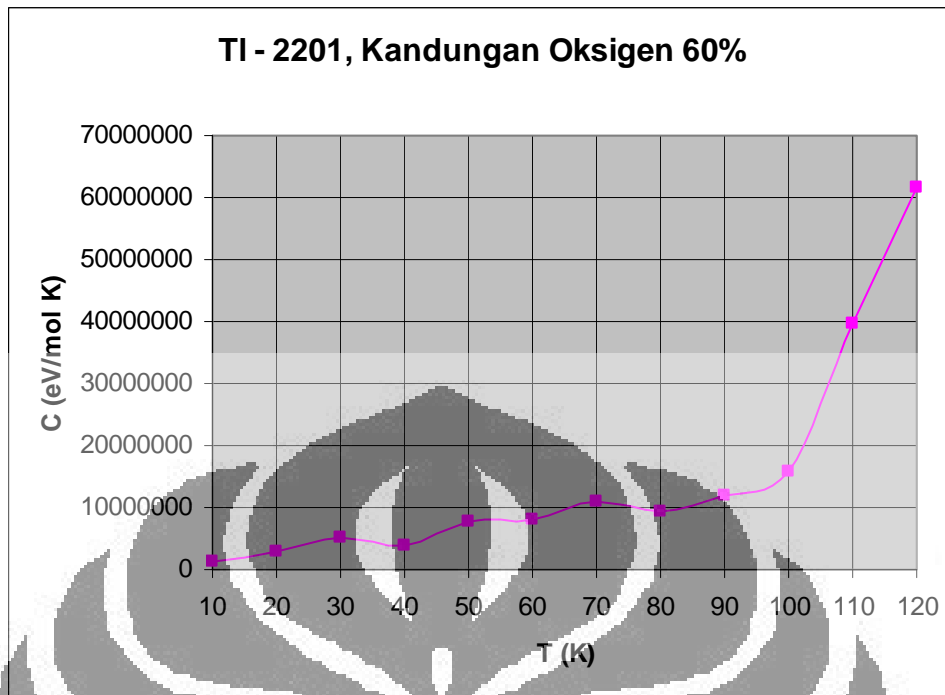
Gambar 5.10. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 80 K



Gambar 5.11. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 70 K



Gambar 5.12. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 50 K

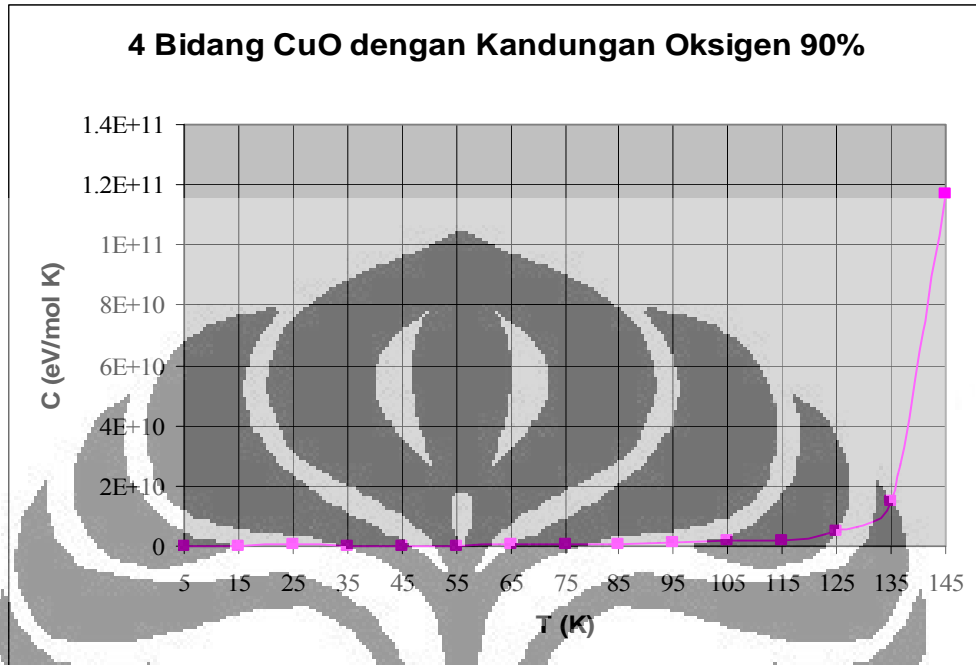


Gambar 5.13. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 40 K

Nilai perbandingan antara diskontinuitas kalor jenis ΔC dengan kalor jenis pada keadaan normal C_n yaitu nilai β dapat dihitung dari grafik kalor jenis terhadap temperatur pada titik terjadinya temperatur transisi. Untuk masing-masing senyawa dengan kandungan oksigen yang berbeda-beda didapatkan beberapa nilai β . Nilai β untuk TI-2201 dengan kandungan oksigen 90 %, 80 %, 70 % dan 60 % masing-masing adalah 1.72, 1.68, 1.38 dan 1.11. Nilai β untuk TI-2212 dengan kandungan oksigen 90 %, 80 %, 70 % dan 60 % masing-masing adalah 1.64, 1.41, 1.37 dan 1.95. Nilai β untuk TI-2223 dengan kandungan oksigen 90 %, 80 %, 70 %, 60 % dan 50% masing-masing adalah 1.21, 1.39, 1.38, 1.79 dan 1.68. Dari teori BCS nilai β adalah 1.43 [13] dan dari eksperimen nilai β untuk Thallium adalah 1.15 [28].

Dari hasil sebelumnya dapat di analisa bahwa dengan bertambahnya jumlah bidang, maka temperatur transisi yang dihasilkan juga naik. Hal ini menjadi dasar untuk dibuatnya program simulasi yang terdiri dari empat bidang CuO dengan harapan temperatur transisi yang dihasilkan lebih tinggi. Gambar 5.14

menunjukkan grafik hasil simulasi TIO *bilayers* untuk empat bidang CuO dengan kandungan oksigen 90%. Masukkannya sama seperti pada TI-2223 hanya berbeda pada energi interkasinya.



Gambar 5.14. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO

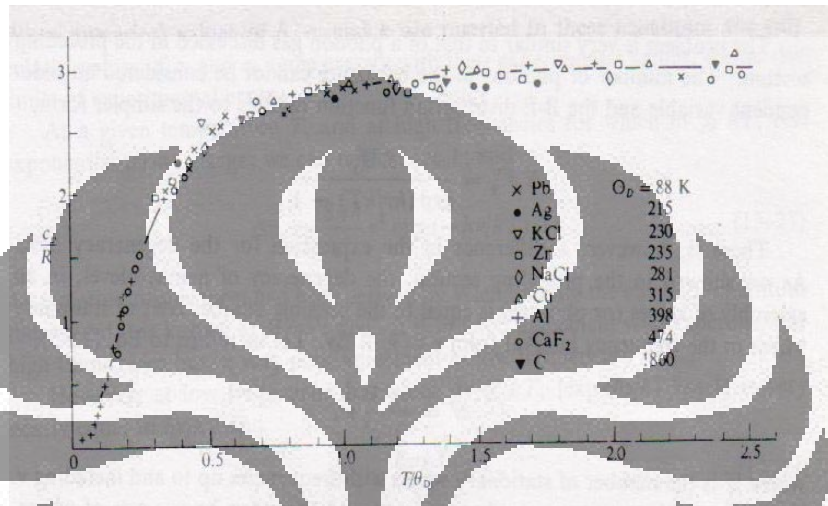
Grafik hasil simulasi TIO *bilayers* untuk empat bidang CuO dengan kandungan oksigen yang lain, dapat dilihat pada Lampiran 1 Gambar 1.a sampai Gambar 1.h.

Hasil simulasi untuk senyawa TIO *bilayers* yang mempunyai empat bidang CuO tidak terdapat temperatur transisi walaupun kandungannya dirubah-rubah dari 90% sampai 10%. Grafik kalor jenis yang didapatkan dari senyawa ini naik mendekati eksponensial dengan naiknya temperatur. Rentang temperatur yang digunakan dalam simulasi dari 5 K sampai 145 K dengan perubahan temperatur sebesar 10 K. Jadi karena tidak terdapatnya temperatur transisi, maka senyawa ini bukan merupakan senyawa superkonduktor. Hasil eksperimen juga menunjukkan bahwa tidak mungkin membuat bahan superkonduktor yang mempunyai empat lapis bidang CuO. Grafik kalor jenis hasil simulasi empat bidang CuO ini dapat dibandingkan dengan beberapa bahan zat padat. Gambar 5.14 menunjukkan grafik dari hasil simulasi, Gambar 5.15 menunjukkan grafik

dari hasil eksperimen dan teori Debye sebagai fungsi dari $\frac{T}{\theta_D}$. Dari teori Debye

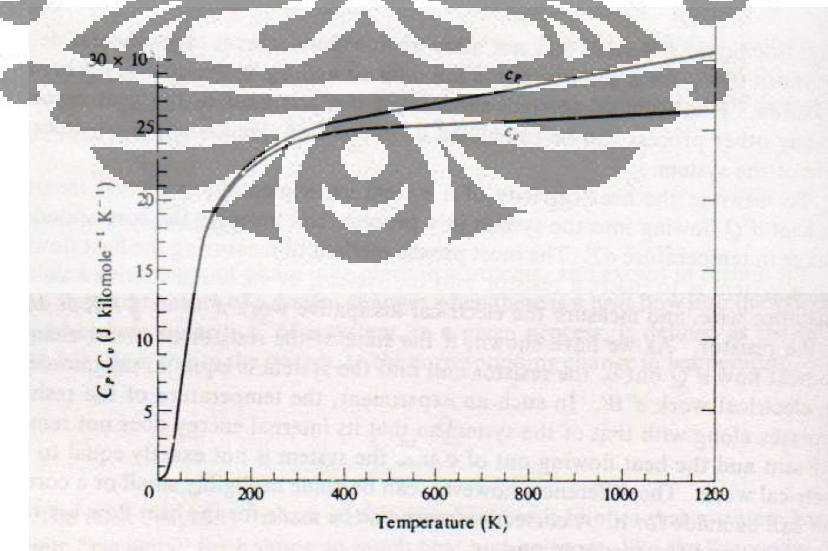
didapatkan kalor jenis $C_v = \frac{12\pi^4}{5} R \left(\frac{T}{\theta_D} \right)^2$ dengan θ_D adalah temperatur Debye.

Untuk Thallium, Calcium dan Tembaga masing-masing mempunyai temperatur Debye 96 K, 226 K dan 315 K [34].



Gambar 5.15. Kalor jenis beberapa bahan zat padat sebagai fungsi dari $\frac{T}{\theta_D}$

Sedangkan Gambar 5.16, menunjukkan grafik hasil eksperimen dari C_v dan C_p untuk tembaga sebagai fungsi dari temperatur pada tekanan konstan 1 atm.



Gambar 5.16. Kalor jenis tembaga sebagai fungsi dari temperatur pada tekanan konstan 1 atm

Pada temperatur sekitar di bawah 145 K, grafik kalor jenis hasil simulasi mendekati grafik kalor jenis beberapa bahan zat padat. Semakin turun suhunya semakin turun juga grafik kalor jenisnya. Pada temperatur sekitar di atas 145 K, hasil eksperimen grafik kalor jenis beberapa zat padat cenderung konstan. Pada suhu ini, simulasi belum dapat menunjukkan bahwa grafik kalor jenis yang dihasilkan cenderung konstan.

Hasil simulasi untuk TIO *monolayers* dilakukan dengan kandungan oksigen 90%. Untuk senyawa ini kandungan oksigennya tidak diubah-ubah bertujuan untuk mendapatkan temperatur transisi yang paling besar dari fasa TIO *monolayers*. Dengan mendapatkan temperatur transisi yang maksimum, maka akan memberikan alasan pada besarnya temperatur transisi yang masih dapat dicapai untuk senyawa TBCCO. Jika temperatur transisi yang didapatkan naik, dengan bertambahnya jumlah bidang dimana jumlahnya lebih dari yang dilakukan dalam eksperimen, maka keadaan ini akan menjadi petunjuk untuk dilakukannya eksperimen selanjutnya. Sedangkan jika menurun, maka akan menjadi alasan untuk temperatur transisi yang masih dapat dicapai untuk fasa TI-2223. Hasil simulasi ini dapat dilihat pada Gambar 5.17 sampai Gambar 5.20. Dari grafik tersebut didapatkan bahwa TI-1201, TI-1212, TI-1223 dan TI-1234 temperatur transisinya masing-masing adalah 50 K, 65 K, 110 K dan 122 K. Temperatur transisi yang didapatkan dari senyawa ini naik dengan naiknya jumlah bidang CuO.

Masukkan:

$$N_x = 70, N_y = 70, N_z = 20$$

$$V_1 = 1.000 \text{ eV}$$

$$V_2 = 0.360 \text{ eV}$$

$$V_3 = 0.120 \text{ eV}$$

$$V_4 = 0.000 \text{ eV}$$

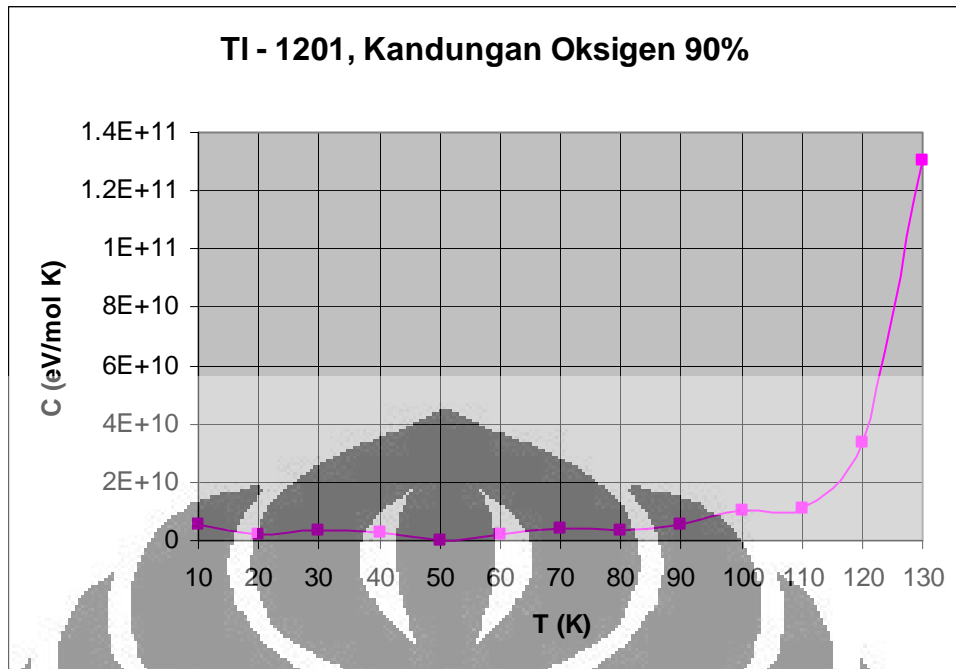
$$\text{grup} = 30$$

$$\text{sweep} = 30$$

$$k_b = 8.125e-05$$

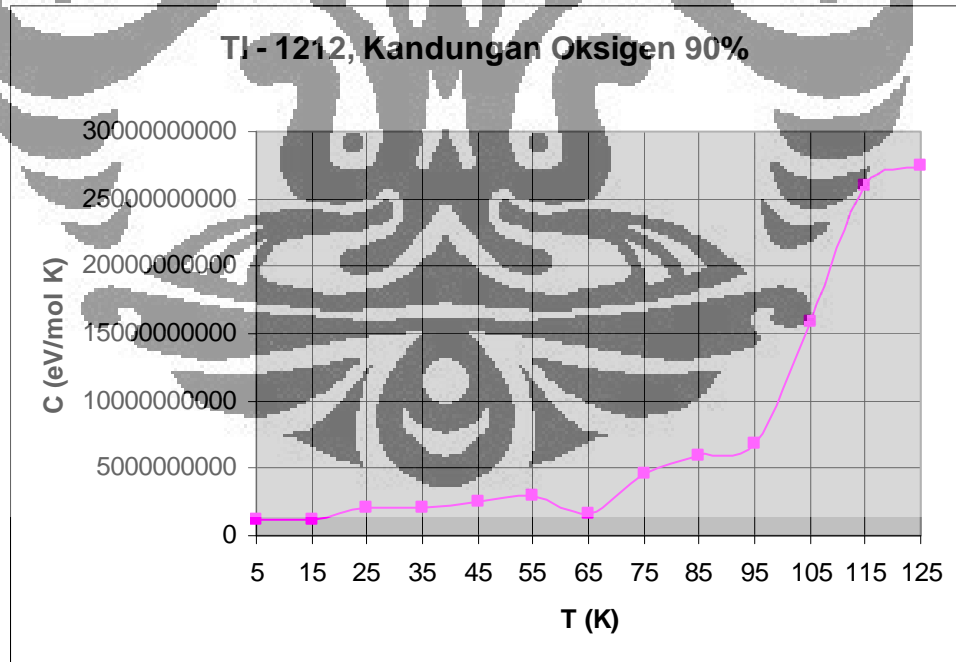
$$x = 0,1$$

$$\Delta T = 10 \text{ K}$$



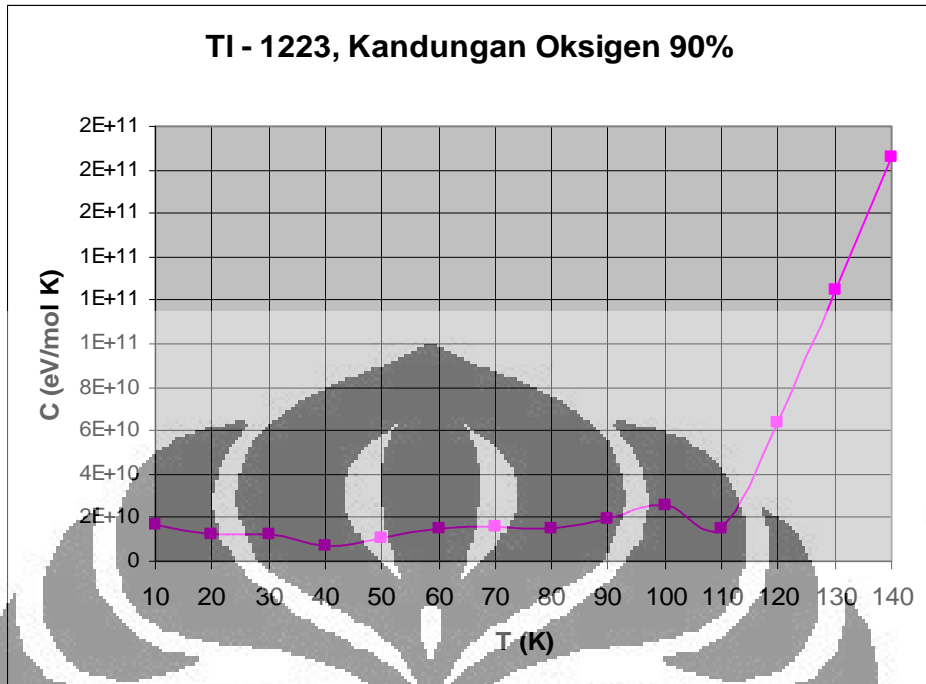
Gambar 5.17. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 50 K

Untuk TI-1212, masukannya sama seperti pada TI-1201 tetapi $v_4 = 0.010$ eV.



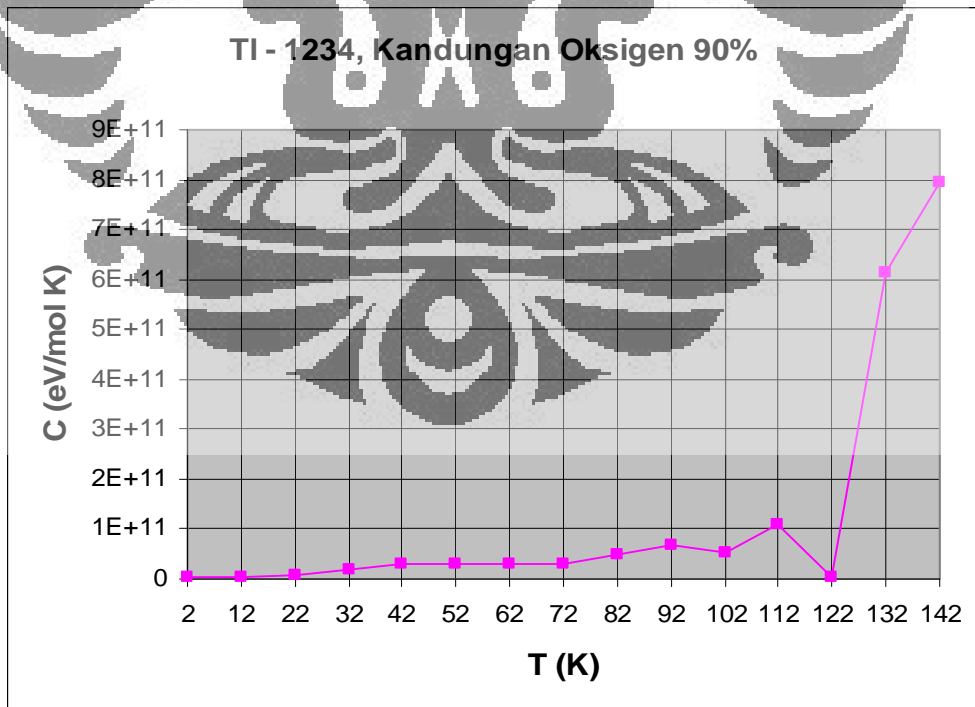
Gambar 5.18. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 65 K

Untuk TI -1223 , masukannya sama seperti pada TI -1212 .



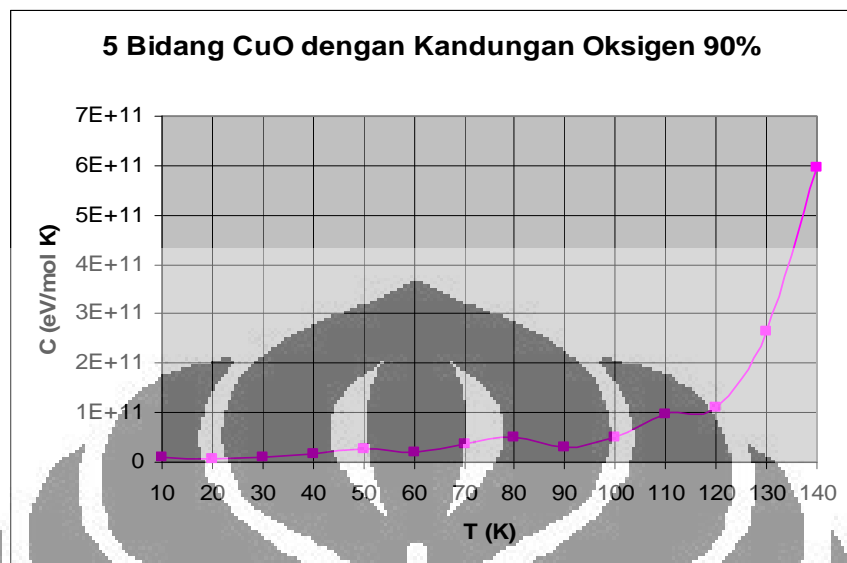
Gambar 5.19. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 110 K

Untuk TI -1234 , masukannya sama seperti pada TI -1212 .



Gambar 5.20. Grafik kalor jenis terhadap temperatur dengan temperatur transisi 122 K

Tetapi untuk TlO *monolayers* yang terdiri dari lima bidang CuO, temperatur transisinya turun sampai 90 K dengan kandungan oksigen 90% (Gambar 5.21).

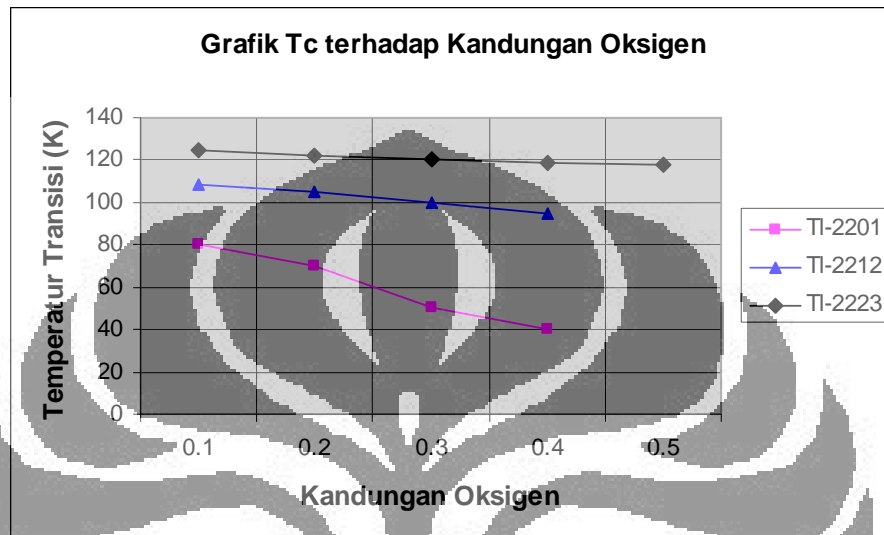


Gambar 5.21. Grafik kalor jenis terhadap temperatur untuk lima bidang CuO dengan temperatur transisi 90 K

Dari Gambar 5.21, lonjakan kalor jenis hampir tidak bisa ditentukan pada temperatur tertentu. Jadi sebenarnya temperatur transisi yang dihasilkan untuk senyawa ini sudah mulai tidak bersifat superkonduktor. Walaupun demikian jika diperhatikan maka bisa didapatkan temperatur transisinya yaitu sekitar 90 K. Jadi walaupun temperatur transisinya 90 K, tetapi nilai ini lebih kecil dibandingkan dengan temperatur transisi untuk senyawa Tl-1234. Pada keadaan ini interaksi fonon-elektron (fonon yang mengikat elektron) semakin berkurang sehingga menurunkan temperatur transisi. Jika interaksi fonon-elektron semakin berkurang, maka yang akan lebih dominan adalah interaksi antara elektron dengan elektron sehingga tidak terjadi pasangan elektron. Karena tidak terjadi pasangan elektron maka bahannya sudah mulai tidak bersifat superkonduktor. Keadaan ini juga memperkuat alasan bahwa untuk superkonduktor TlO *bilayers* yang terdiri dari empat bidang sudah tidak bersifat superkonduktor. Jadi titik ekstrim dimana terjadinya temperatur transisi maksimum terjadi pada superkonduktor TlO *bilayers* yang terdiri dari tiga bidang CuO. Temperatur transisi maksimum yang didapatkan adalah 125 K.

5.2 Pengaruh Kandungan Oksigen Terhadap Temperatur Transisi

Kandungan oksigen yang berbeda-beda akan mempengaruhi pada besarnya temperatur transisi. Grafik antara kandungan oksigen dan temperatur transisi dapat digambarkan dari grafik kalor jenis terhadap temperatur. Gambar 5.22 merupakan grafik untuk fasa Tl-2201, Tl-2212 dan Tl-2223.



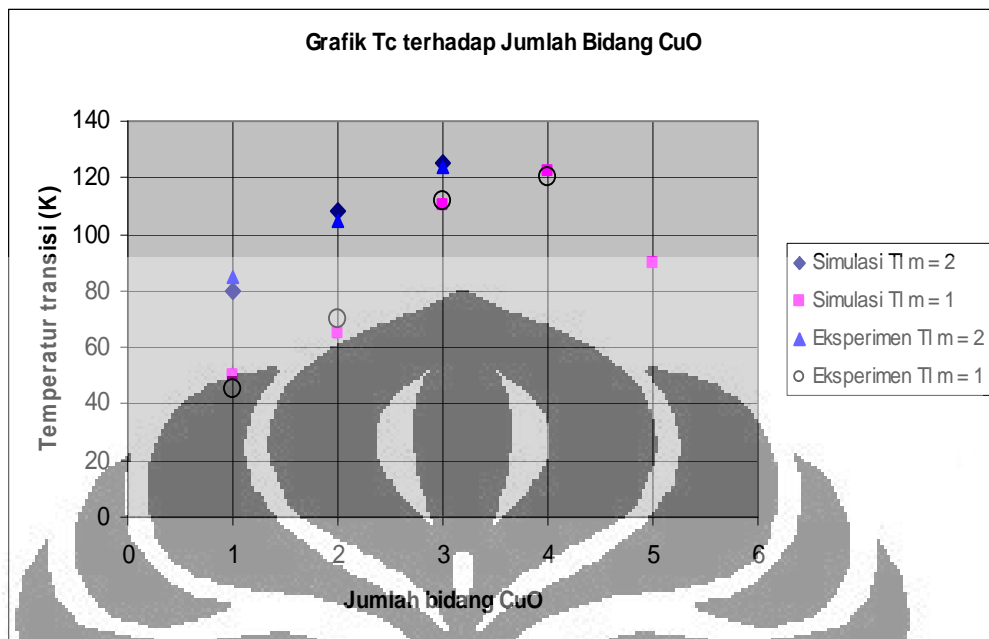
Gambar 5.22. Grafik temperatur transisi terhadap kandungan oksigen

Hasil simulasi menunjukkan bahwa ketiga senyawa Tl-2201, Tl-2212 dan Tl-2223 temperatur transisinya menurun dengan berkurangnya kandungan oksigen. Kandungan oksigen ini dapat dibandingkan dengan grafik antara temperatur transisi dengan jumlah bidang CuO. Dengan kandungan oksigen yang semakin berkurang seolah-olah akan mengurangi jumlah bidang CuO.

5.3 Temperatur Transisi Terhadap Jumlah Bidang CuO

Selain kandungan oksigen yang berbeda-beda, dapat digambarkan juga besarnya temperatur transisi terhadap jumlah bidang CuO. Gambar 5.23 merupakan grafik dari hasil simulasi dan eksperimen. Temperatur transisi maksimum diperoleh pada saat kandungan oksigen 90%. Pada Gambar 5.23 temperatur transisi hasil simulasi yang digambarkan adalah untuk kandungan oksigen 90%. Dari Gambar 5.23, temperatur transisi hasil simulasi untuk

$Tl_2Ba_2CuO_{6-x}$ ($n = 1$), $Tl_2Ba_2CaCu_2O_{8-x}$ ($n = 2$) dan $Tl_2Ba_2Ca_2Cu_3O_{10-x}$ ($n = 3$) sesuai dengan hasil eksperimen.

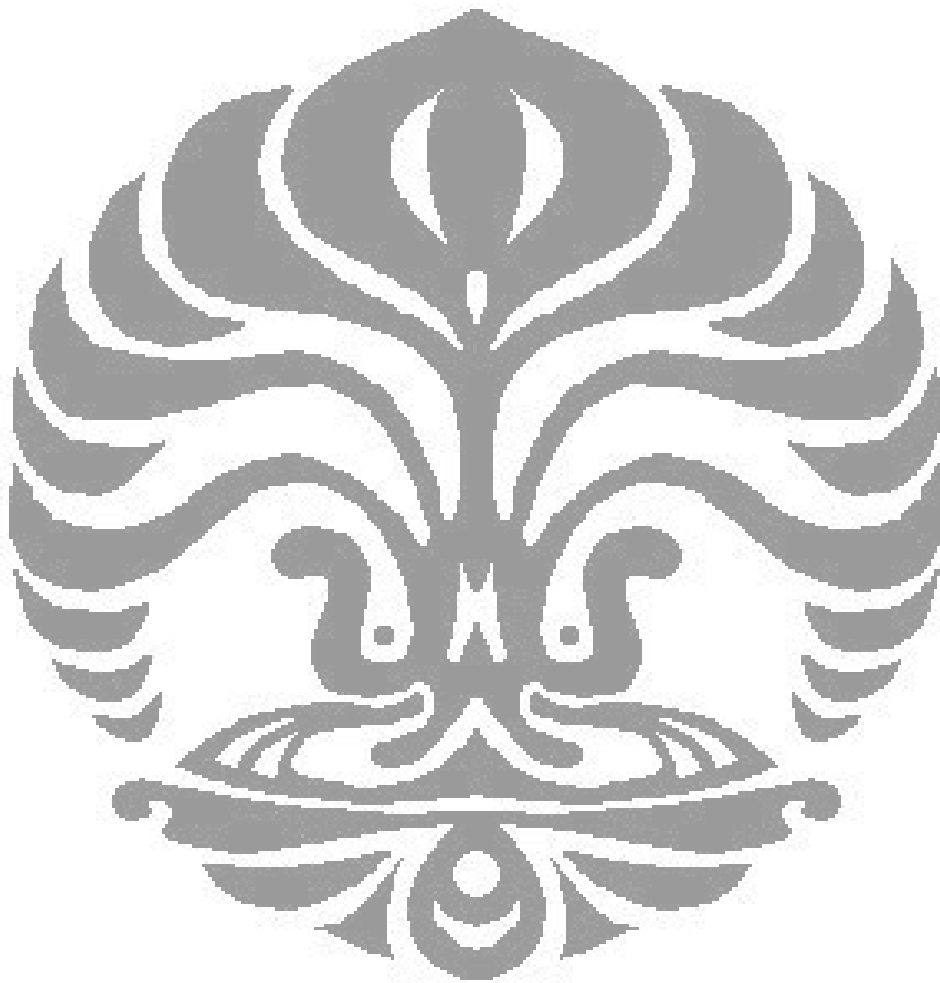


Gambar 5.23. Grafik hasil simulasi dan eksperimen pada TBCCO

Hasil simulasi TIO *bilayers* menunjukkan bahwa ketiga senyawa TBCCO yang mempunyai satu bidang CuO (TI-2201), dua bidang CuO (TI-2212) dan tiga bidang CuO (TI-2223) temperatur transisinya naik dengan bertambahnya jumlah bidang CuO. Pada saat kandungan oksigen 90 %, temperatur transisi untuk senyawa TI-2201, TI-2212 dan TI-2223 masing-masing adalah 80 K, 108 K dan 125 K. Atau sebaliknya semakin sedikit jumlah bidang CuO, maka temperatur transisinya semakin berkurang. Temperatur transisi juga berkurang dengan berkurangnya kandungan oksigen. Jika kandungan oksigen semakin berkurang, maka seolah-olah akan mengurangi jumlah bidang CuO, sehingga temperatur transisi turun dengan berkurangnya kandungan oksigen dan jumlah bidang CuO.

Simulasi pada senyawa TIO *bilayers* yang mempunyai empat lapis bidang CuO tidak menunjukkan adanya temperatur transisi, karena pada grafik kalor jenis tidak terjadi lonjakan. Keadaan ini tentu tidak bisa digambarkan pada grafik temperatur transisi terhadap jumlah bidang CuO yaitu pada Gambar 5.23. Hasil eksperimen pada Gambar 5.23 menunjukkan bahwa untuk nilai $m = 1$ masih mungkin dibuat superkonduktor yang mempunyai empat lapis bidang CuO, tetapi

temperatur transisi terbesar yang dihasilkan tetap terjadi pada Tl-2223 yang mempunyai tiga lapis bidang CuO. Simulasi juga menunjukkan untuk TlO *monolayers* yang terdiri dari lima bidang CuO menghasilkan temperatur transisi yang lebih kecil dari Tl-1234. Keadaan ini menunjukkan bahwa temperatur transisi maksimum yang masih bisa dicapai adalah pada senyawa TlO *bilayers* yang mempunyai tiga bidang CuO.



BAB VI KESIMPULAN DAN SARAN

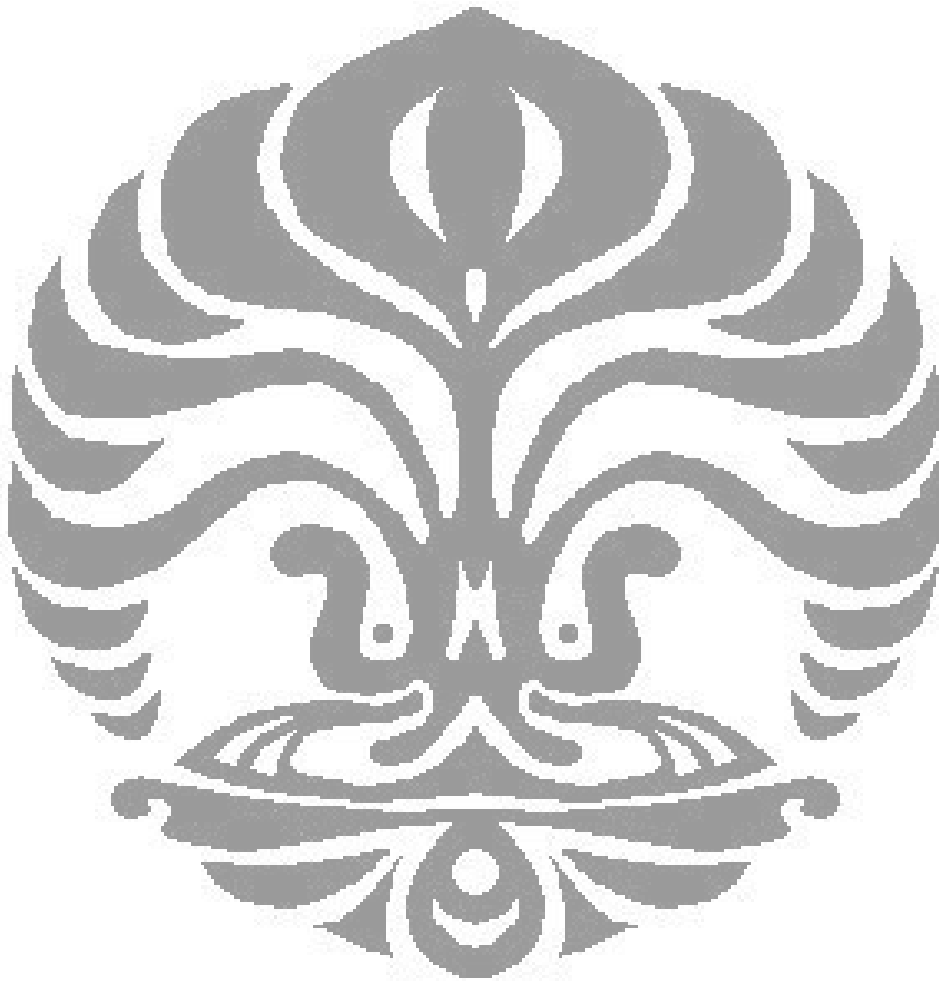
Dari pembahasan pada BAB V dapat disimpulkan bahwa:

1. Hasil simulasi menunjukkan bahwa temperatur transisi dipengaruhi oleh kandungan oksigen. Temperatur transisi naik dengan bertambahnya kandungan oksigen.
2. Temperatur transisi senyawa TBCCO naik dengan naiknya jumlah bidang CuO sampai pada batas tertentu. Untuk TlO *bilayers* temperatur transisi terbesar terjadi pada senyawa yang terdiri dari tiga bidang CuO, sedangkan untuk TlO *monolayers* temperatur transisi terbesar terjadi pada senyawa yang terdiri dari empat bidang CuO.
3. Pada TlO *monolayers* yang terdiri dari lima bidang CuO, temperatur transisinya lebih kecil dari pada TlO *monolayers* yang terdiri dari empat bidang. Temperatur transisi ini menurun karena interaksi fonon-elektron semakin kecil. Keadaan ini memperkuat alasan bahwa temperatur transisi tertinggi didapatkan pada TlO *bilayers* yang terdiri dari tiga bidang CuO.
4. Pada senyawa TlO *bilayers* yang terdiri dari empat lapis bidang CuO tidak terdapat temperatur transisi dan kalor jenis yang dihasilkan mendekati kalor jenis zat padat yang tidak bersifat superkonduktor.
5. Grafik kalor jenis untuk senyawa TlO *bilayers* yang mempunyai empat lapis bidang CuO pada temperatur di bawah 145 K mendekati kalor jenis beberapa bahan zat padat, tetapi masih belum tepat untuk temperatur di atas 145 K.

sehingga saya menyarankan:

1. Perlu dilakukan penelitian lebih lanjut untuk senyawa TlO *bilayers* yang terdiri dari empat bidang CuO yang dapat menunjukkan bahwa kalor jenisnya mengikuti kalor jenis bahan tertentu yang tidak bersifat superkonduktor khususnya pada temperatur di atas 145 K.
2. Perlu dilakukan penelitian lebih lanjut untuk senyawa TlO *monolayers* yang mempunyai lebih dari lima bidang CuO.
3. Perlu dilakukan penelitian lebih lanjut pengaruh dari energi interaksi antar bidang terhadap temperatur transisi.

4. Perlu dilakukan penelitian lebih lanjut dengan memasukkan besaran-besaran lain seperti potensial kimia.
5. Perlu dilakukan simulasi dengan kemampuan komputer yang lebih baik, sehingga ukuran kisi yang digunakan lebih besar.
6. Perlu dilakukan penelitian dengan menggunakan metode komputasi paralel untuk mempercepat komputasi.



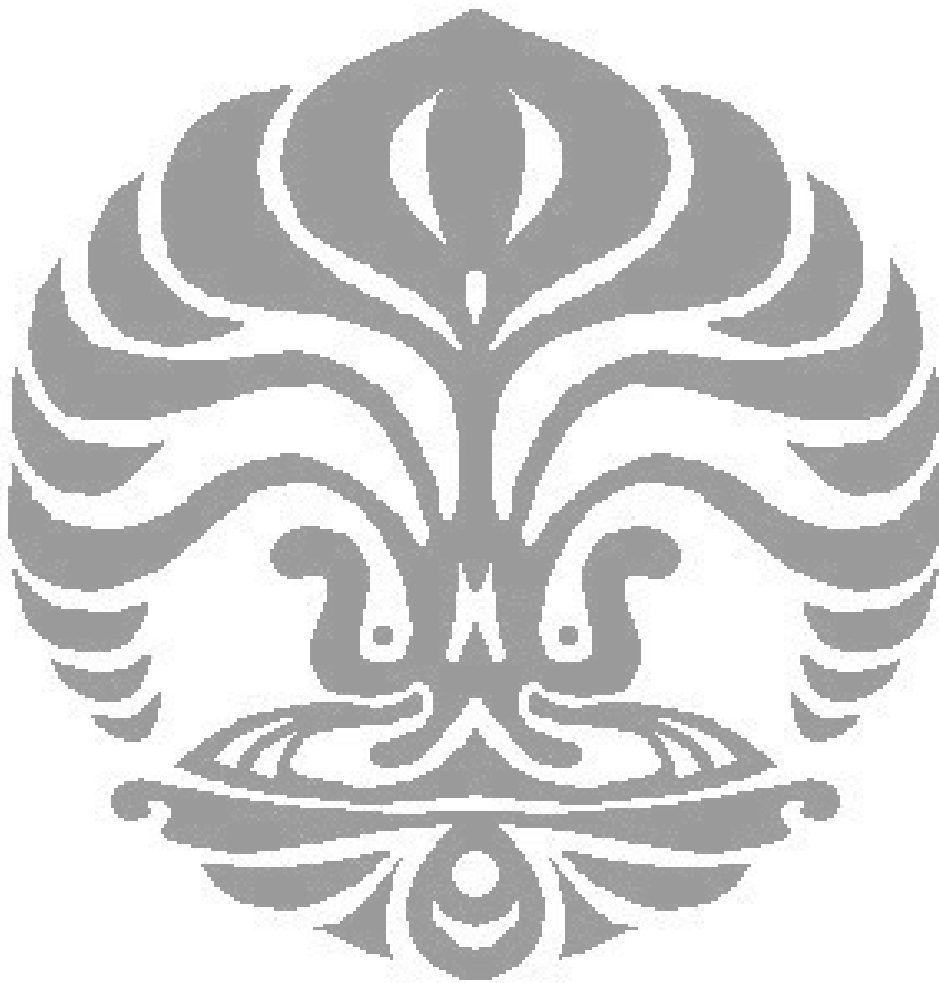
DAFTAR REFERENSI

- [1] Kamimura H, Ushio H, Matsuno S, Hamada Tsuyoshi. 2005. Theory of Copper Oxide Superconductors. Springer-Verlag Berlin Heidelberg.
- [2] Charles P. Poole Jr, Horacio A. Farach, Richard J. Creswick, Ruslan Prozorov. 2007. Superconductivity. Elsevier Ltd.
- [3] R. B. Beyers, S. S. P. Parkin, V. Y. Lee, A. I. Nazzal, R. J. Savoy, G. L. Gorman, T. C. Huang, S. J. La Placa. 1989. Tl-Ca-Ba-Cu-O Superconducting Oxides. IBM J. RES. Develop. Vol 33 No.3
- [4] J. M. Chen, R. G. Liu, R. S. Liu, H. C. Lin, T. M. Uen, J. Y. Juang, and Y. S. Gou. 1997. Polarized X-ray absorption Studies in Double-Thallium-Layer Superconducting. Chinese Journal of Physics. Vol. 36, No. 2-II.
- [5] R. Zalecki, A. Kolodziejczyk, O. Heiml and G. Gritzner. 2005. Temperature and Magnetic Field Dependences of Critical Current of Thallium-Based Superconducting Films. Actaphysica Polonica A Vol. 109 (2006).
- [6] F. Foong, B. Bedard, Q. L. Xu, Sy-Hwang Liou. 1996. C-axis Oriented (Hg,Ti)-based Superconducting Films with $T_c \geq 125$ K. Appl. Phys. Lett. 68(8).
- [7] P. P. Freitas and T. S. Plaskett. 1987. High-temperature order-disorder phase transition in the superconductor $Y_1Ba_2Cu_3O_{6+\delta}$ observed by electrical resistivity measurements. Phys. Rev. B Vol. 36, No. 10.
- [8] J. D. Jorgen, B. W. Veal, W. K. Kwok, G. W. Crabtree, A. Umezawa, L. J. Nowicki, and A. P. Paulikas. 1987. Phys. Rev. B Vol. 36, No. 10.
- [9] Z. Z. Wang, J. Clayhold, and N. P. Ong. 1987. Variation of Superconductivity with Carrier Concentration in Oxygen-doped $YBa_2Cu_3O_{7-y}$. Phys. Rev. B Vol. 36, No. 13.
- [10] T. J. Kistenmacher, W. A. Bryden, J. S. Morgan, and K. Moorjani. 1987. Stabilization of the tetragonal phase and superconducting behavior in $RBa_2(Cu_{1-x}Fe_x)_3O_y$ ($R = Y, Gd; 0 \leq x \leq 0.15$). Phys. Rev. B Vol. 36, No. 16.

- [11] C. Chaillout, M. A. Alario-Franco, J. J. Capponi, J. Chenavas, J. L. Hodeau, and M. Marezio. 1987. Oxygen-vacancy ordering in the $\text{Ba}_2\text{YCu}_3\text{O}_{7-x}$ ($0 \leq x \leq 1$) superconducting system. *Phys. Rev. B* Vol. 36, No. 13.
- [12] D. de Fontaine and L. T. Wille. 1987. Stability analysis of Special-point ordering in the basal plane in $\text{YBa}_2\text{Cu}_3\text{O}_{7-\delta}$. *Phys. Rev. B* Vol. 36, No. 10.
- [13] Pamungkas M. A. 2001. Perhitungan Kalor Jenis Superkonduktor $\text{YBa}_2\text{Cu}_3\text{O}_{7-x}$ dengan Model ASYNNNI. Universitas Indonesia: Thesis S2 Fisika Murni dan Terapan.
- [14] Suherman A. 2006. Perhitungan Kalor Jenis Superkonduktor BSCCO dengan Model ASYNNNI. Universitas Indonesia: Thesis S2 Fisika Murni dan Terapan.
- [15] Young Cheol KIM, Seok Kil HAN, Myung Woon KANG and Min Su JANG. 1990. Magnetic Properties of the High-Tc Superconductor $\text{Tl}_2\text{Ca}_2\text{Ba}_2\text{Cu}_3\text{O}_y$. *Journal of the Korean Physical Society*, Vol. 24, No. 2.
- [16] N. E. Alekseevskii, G. M. Kuzmicheva, A. V. Mitin, V. I. Nizhankovskii, E. G. Nikolaev, and E. P. Khlybov. 1989. NMR Spectra of ^{205}Tl in the high-temperature superconductors $\text{Tl}_2\text{Ba}_2\text{Ca}_n\text{Cu}_{n+1}\text{O}_{6+2n}$ with $n = 0, 1, 2$. *JETP Lett.*, Vol. 51, No. 1.
- [17] Charles P. Poole, Jr. Timir Datta, Horacio A. Farach., 1988. *Copper Oxide Superconductors*. Canada : John Wiley & Sons, Inc.
- [18] D. M. Newns. 1987. Two-particle resonance pairing mechanism of oxide superconductivity. *Phys. Rev. B* Vol. 36, No. 10.
- [19] F. Marsiglio, R. Akis, and J. P. Carbotte. 1987. Thermodynamics in very strong coupling: A possible model for the high-Tc oxides. *Phys. Rev. B* Vol. 36, No. 10.
- [20] T. Fiig, N.H. Andersen, P. A. Lingard, J. Berlin, O.G Mouritsen, *Phys Rev B* 54, 556.
- [21] Thomas Fiig. 1994. Ordering Phenomena and Non-Equilibrium Properties of Lattice Gas Models. Denmark : Riso National Laboratory, Roskilde.
- [22] K. Binder, D. W. Heermann. 1992. Monte Carlo Simulation in Statistical Physics. Springer-Verlag Berlin Heidelberg.

- [23] Betts, Davids, Roy E. Turner. *Introductory Statistical Mechanics*. New York: Addison Wesley.
- [24] Gould H, Tobochnik J. 1996. *An Introduction to Computer Simulation Methods. Application to Physical Systems*. Addison-Wesley Publishing Company, Inc.
- [25] Marouchkine A. 2004. *Room-Temperature Superconductivity*. England: Cambridge International Science Publishing.
- [26] M. E. Reeves, D. S. Citrin, B. G. Pazol, T. A. Friedmann, and D. M. Ginsberg. 1987. Specific heat of $GdBa_2Cu_3O_{7.8}$ in the normal and superconducting states. *Phys. Rev. B* Vol. 36, No. 13.
- [27] G. Nieva, E. N. Martinez, F. De la Cruz, D. A. Esparza, and C. A. D'Ovidio. 1987. Specific heat of $La_{1.8}Sr_{0.2}CuO_4$: Anomaly at the superconducting transition. *Phys. Rev. B* Vol. 36, No. 16.
- [28] Cardwell D. A, Ginley D. S. 2003. *Handbook of Superconducting Materials. Volume I: Superconductivity, Materials and Processes*. Philadelphia: IOP Publishing Ltd.
- [29] P. M. Horn, D. T. Keane, G. A. Held, J. E. Jordan-Sweet, D. L. Kaiser, and F. Holtzberg. 1987. Orthorhombic Distortion at the Superconducting Transition in $YBa_2Cu_3O_7$: Evidence for Anisotropic Pairing. *Phys. Rev. Lett.* Vol. 59, No. 24.
- [30] A. G. Khachatryan and J. W. Morris, Jr. 1987. Ordering and Decomposition in the High-Temperature Superconducting Compound $YBa_2Cu_3O_x$. *Phys. Rev. Lett.* Vol. 59, No. 24.
- [31] J. P. Andreea, H. C. Basso, E. E. Castellano, J. N. H. Gallo, and A. A. Martin. 1987. Crystallographic phase and transport properties in the superconducting Tm-Ba-Cu-O compound system. *Phys. Rev. B.* Vol. 36, No. 10.
- [32] L. R. Testardi, W. G. Moulton, H. Mathias, H. K. Ng, and C. M. Rey. 1987. Superconducting and nonsuperconducting phase of $YBa_2Cu_3O_x$: Modifications at the high-temperature phase transition. *Phys. Rev. B.* Vol. 36, No. 16.

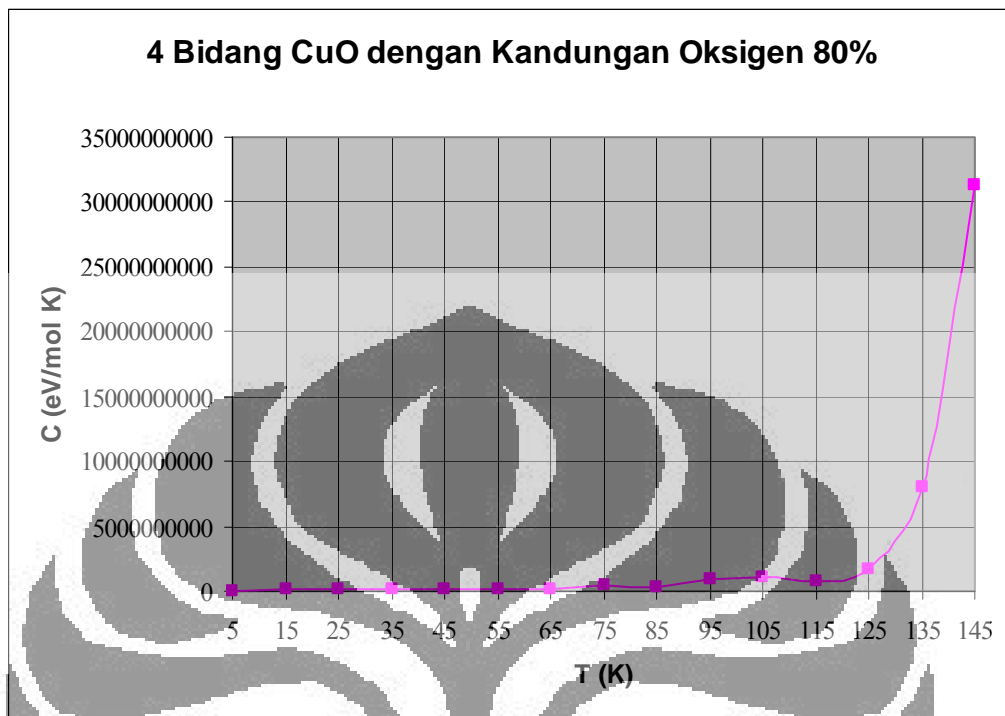
- [33] Maria Ronay. 1987. Hole formation in orthorombic and tetragonal $\text{YBa}_2\text{Cu}_3\text{O}_{7-x}$. Phys. Rev. B. Vol. 36, No. 16.
- [34] Francis W. Sears, Gerhard L. Salinger. 1975. Thermodynamics, Kinetic Theory, and Statistical Thermodynamics. Philippines: Addison-Wesley Publishing Company, Inc.



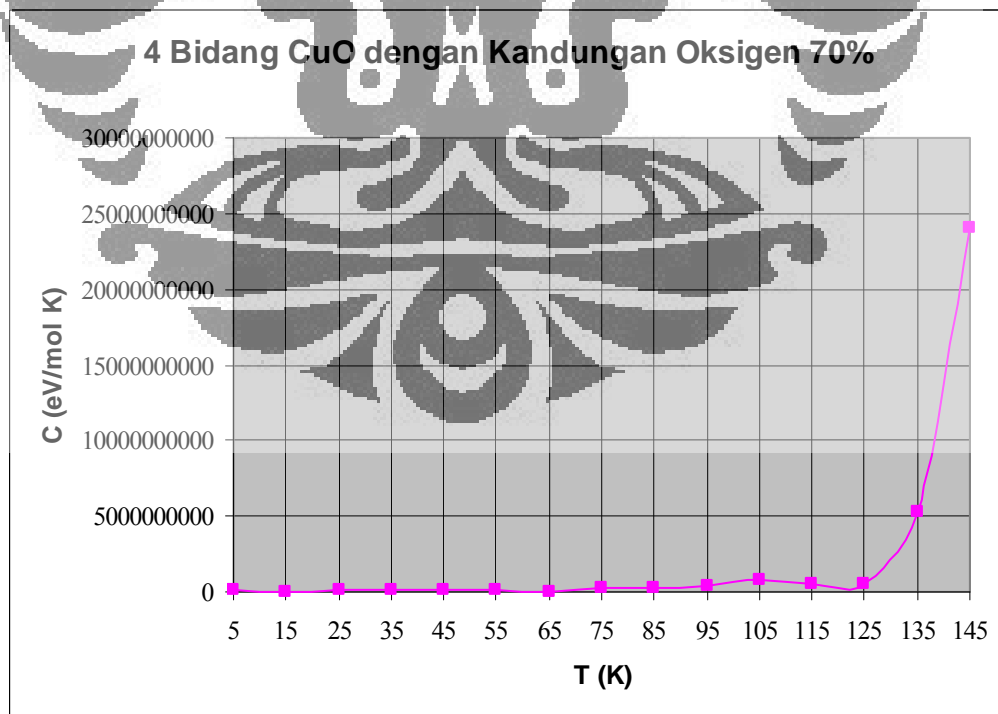


LAMPIRAN 1
Grafik Hasil Simulasi

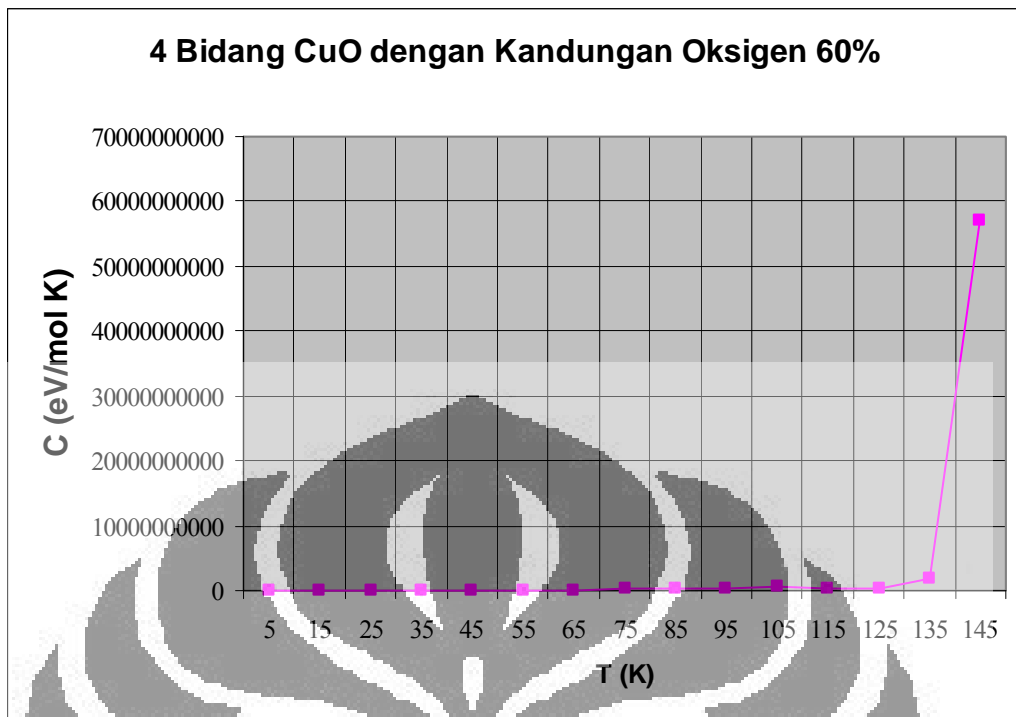
Berikut ini adalah grafik hasil simulasi untuk senyawa TlO *bilayers* yang terdiri dari empat bidang CuO dengan kandungan oksigen yang berbeda-beda.



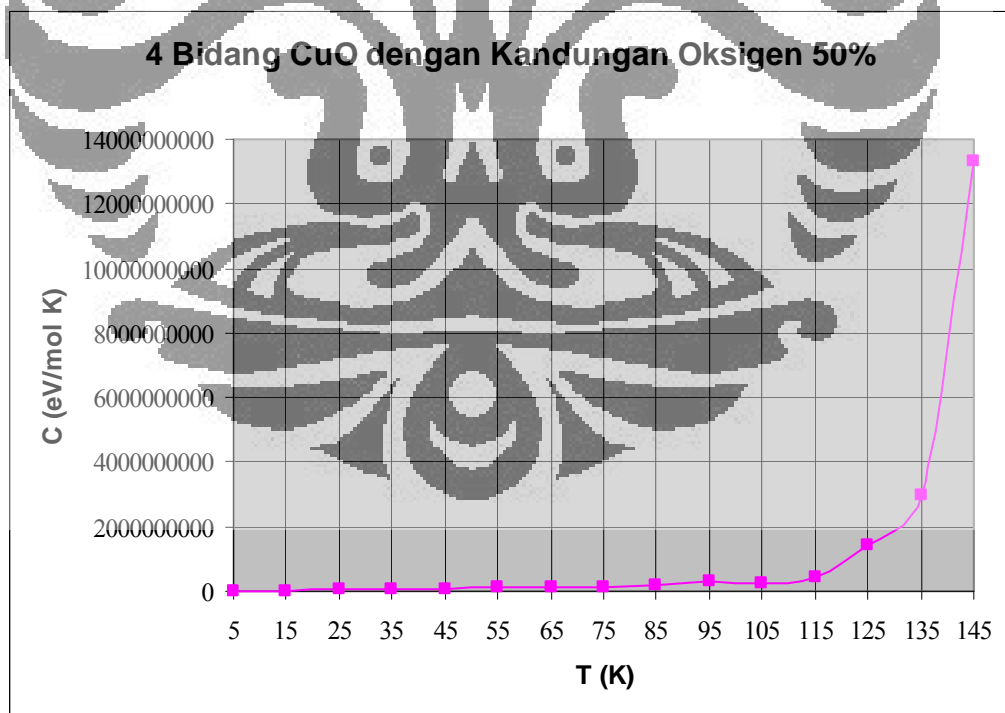
Gambar 1.a. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO



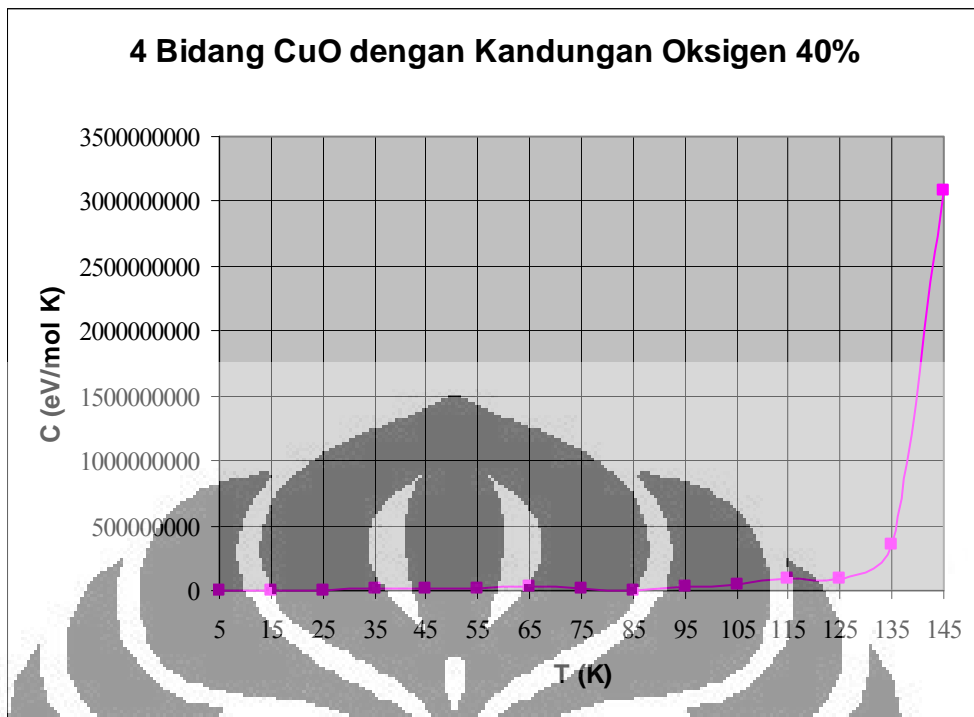
Gambar 1.b. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO



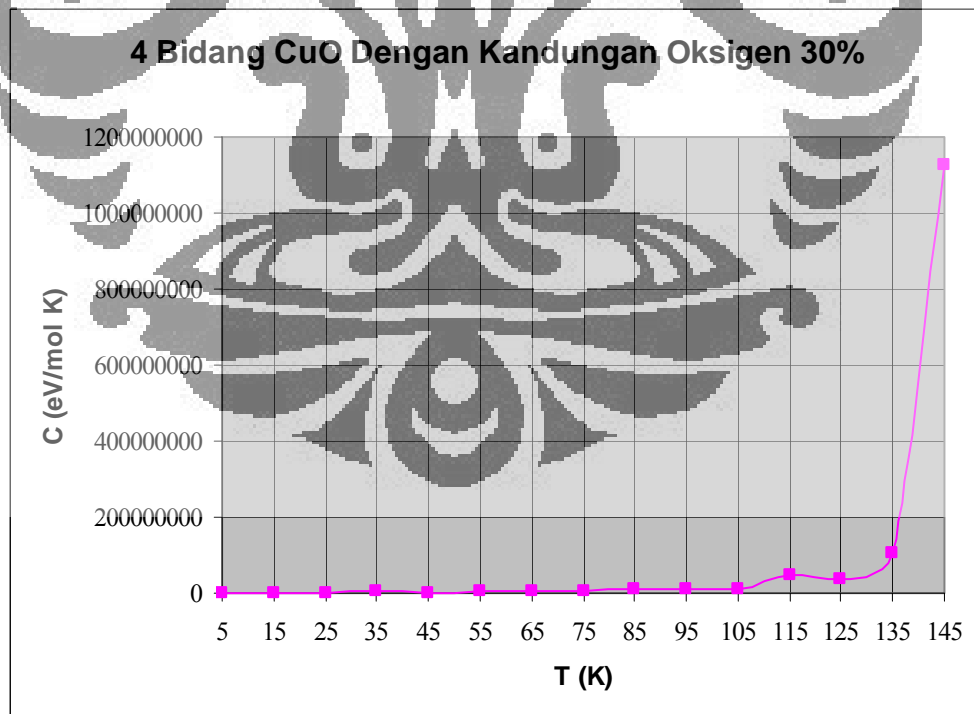
Gambar 1.e. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO



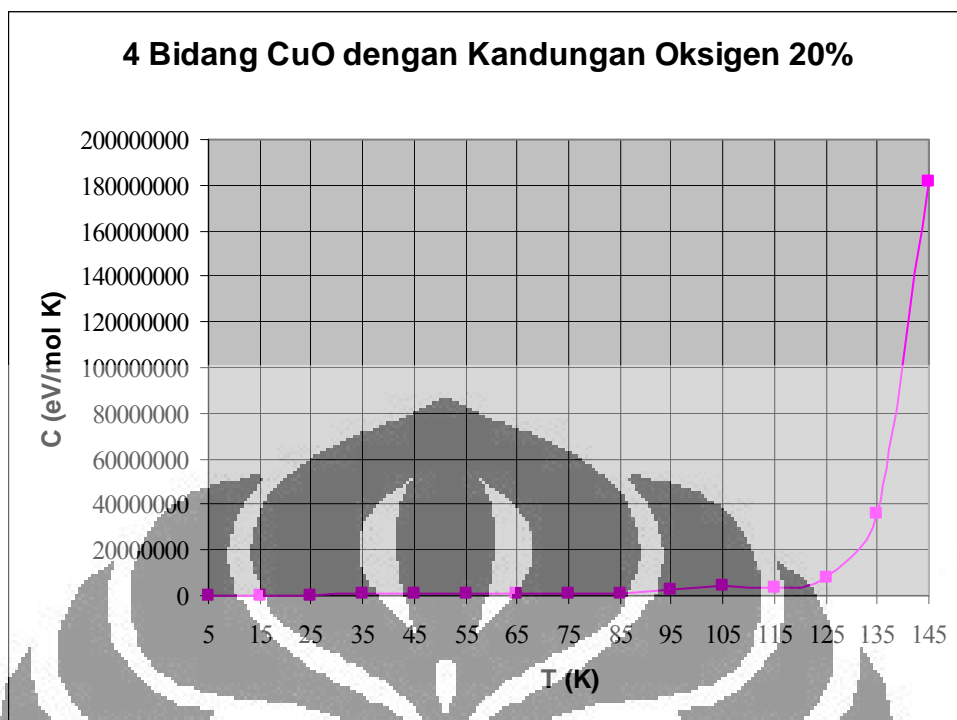
Gambar 1.d. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO



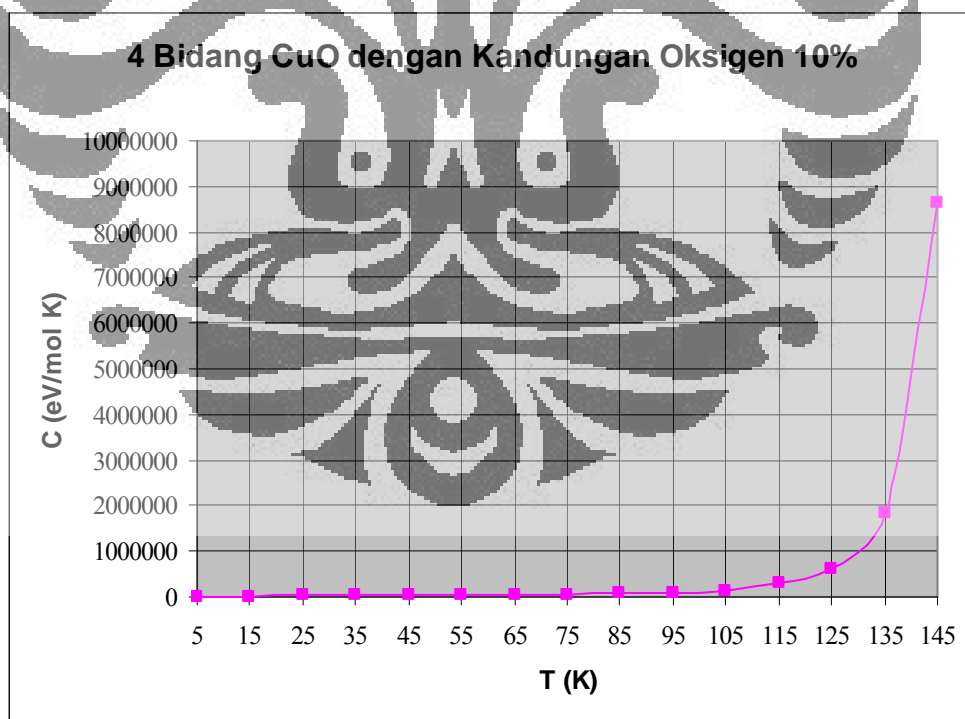
Gambar 1.e. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO



Gambar 1.f. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO



Gambar 1.g. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO

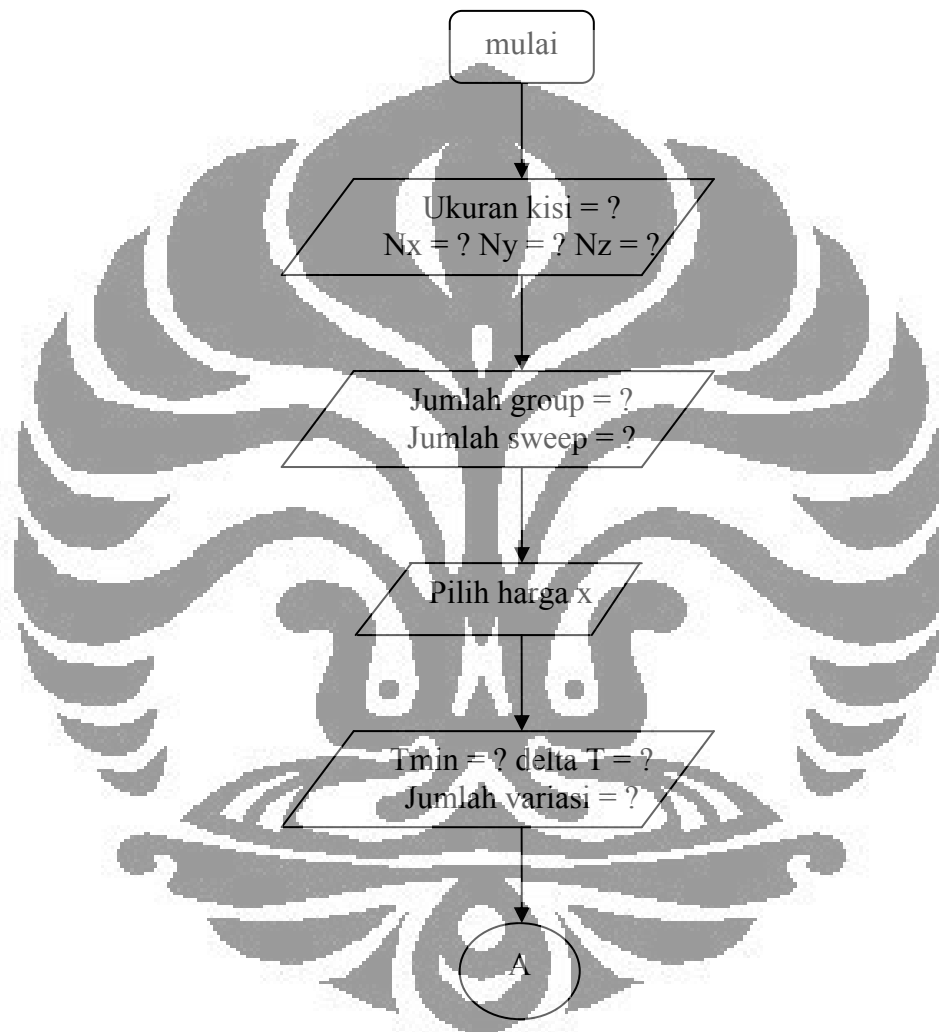


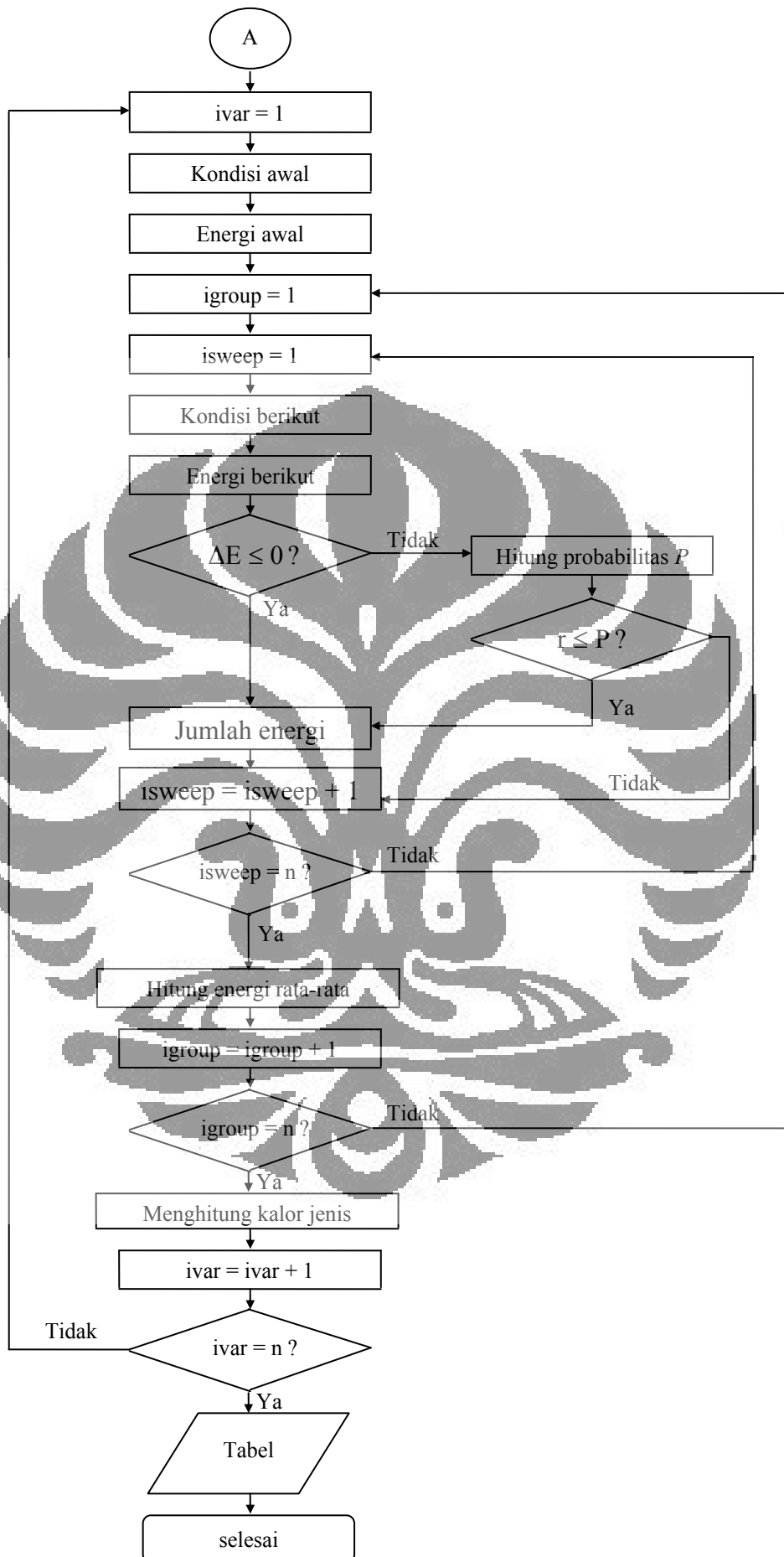
Gambar 1.h. Grafik kalor jenis terhadap temperatur untuk empat bidang CuO

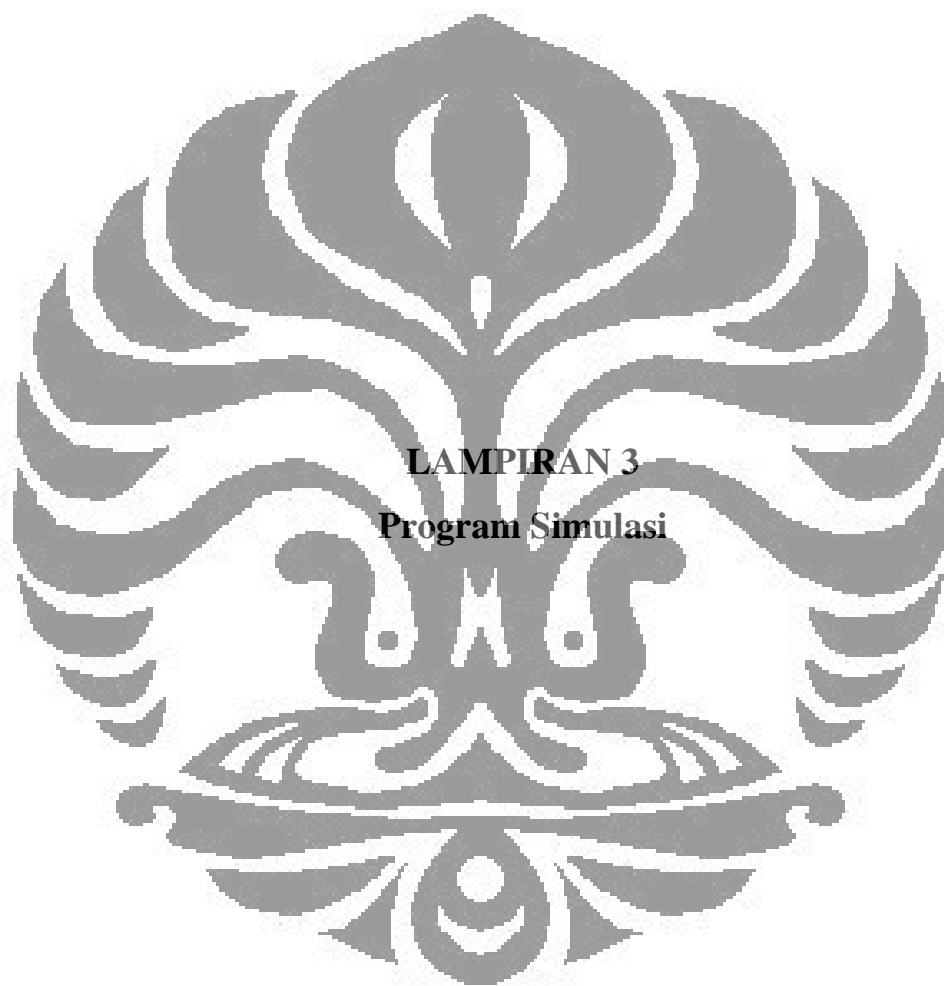


LAMPIRAN 2

Flow Chart Program Simulasi







LAMPIRAN 3
Program Simulasi

```

/*-----*/
/* PROGRAM MODEL ASYNNNI TI 2201 */
/* Saepul Qodar S2 Fisika UI */
/* OS : Microsoft Windows XP SP2 */
/* Compiler : Turbo C++ 4.5 */
/* Memory Model : Large */
/*-----*/
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <dos.h>
#include <conio.h>
#include <stdio.h>
int pil=0,
    x=0,
    w=0,
    y=0,
    z=0,
    i1=0,
    j1=0,
    k1=0,
    i2=0,
    j2=0,
    k2=0,
    i5=0,
    j5=0,
    k5=0,
    i6=0,
    j6=0,
    k6=0,
    Besarx=0;
long w1,wt;
int ngroup=0,
    nsweep=0;
double nspin=0;
double probabilitas=0;
double Etotal=0,Ekkuadrat=0,
    rata_group_energi=0,
    rata_group_kuadrat=0;
int jumlahnol=0,jumlahsatu=0;
double groupe=0,groupe2=0;
double energi_total=0,
    E1=0,
    E1_rata_rata=0,
    E2=0,
    E2_rata_rata=0,
    delta_energi=0,
    E6=0,
    E8_rata_rata=0;
double rata_rata=0,
    rata_rata_kuadrat=0,
    kuadrat=0,
    kuadrat_rata_rata=0,
    rata_sweep_energi=0,
    rata_sweep_kuadrat=0,
    E_hitung=0,
    delta=0,

```

```

        absolut=0,
        pertemperatur=0,
        sum_sweep_kuadrat=0,
        sum_sweep_energi=0,
        sume=0,
        sume2=0;

int nvar=0;
double delt=0,Tawal=0,d=0;
double huge spesifik_heat[2400],E3[4000],T[25];
double kalor_jenis=0;
int jumlah_random=0,
    nilai_random=0;
int nilai_random2=0;
int sweepke=0,grupke=0;

long huge S[100L][100L][100L];/*jumlah posisi atom oksigen maksimum 100x100x100*/
double cv=0,
    cvc=0;
double v1=0,
    v2=0,
    v3=0;

int hitung2=0,hitung1=0,n=0;
double kb=0,E=0;
char pil1='Y';
void tampilan(void),input_parameter(void),energi_pembanding(void),kisi(void);
void kisi_awal(void),energi_awal(void),kisi_berikut(void),energi_berikut(void);
int hitungE_M(void);
void hasil(void),temperatur(void),kalor(void),tabel(void);
extern int _directvideo=0;
int main(void)
{
    register int isweep,igroup,test,ivar;

    tampilan();
    input_parameter();
    temperatur();
    w1=time('\0');
    randomize();
    /*Perhitungan dengan temperatur yang berbeda-beda*/
    for (ivar=1;ivar<=nvar;++ivar)
    {
        T[1]=Tawal;
        T[ivar+1]=T[ivar]+delt;
        kalor_jenis=0;
        sume=0;
        sume2=0;

        /*Menentukan posisi awal atom dan menghitung energinya*/
        kisi();
        energi_awal();
        gotoxy(20,12);
        printf("Nx=%d Ny=%d Nz=%d \r\n",x,y,z);
        gotoxy(20,20);printf("wait...");
        sum_sweep_kuadrat=0;sum_sweep_energi=0,grupke=0;
        hitung2=0;hitung1=0;E_hitung=0;

        n=0;

```



```

for (igroup=1;igroup<=ngroup;++igroup)
{
groupe=0;groupe2=0;Etotal=0;
Ekuadrat=0;sweepke=0;
for (isweep=1;isweep<=nsweep;++isweep)
{
hitung2=hitung2+1;
hitung1=hitung2-1;
/*Menentukan posisi kedua,ketiga dst atom dan menghitung energinya*/
kisi_berikut();
energi_berikut();
E3[hitung2]=E6;
/*Mencari delta energi,E2-E1,E3-E2,E4-E3 dst*/
if (hitung2==1)
delta=E3[hitung2]-E1;
else
delta=E3[hitung2]-E3[hitung1];
kb=0.00008125;
if (delta <= 0)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
pertemperatur=delta/(2*T[ivar]*kb);
probabilitas=0.5*(1-tanh(pertemperatur));
nilai_random=random(32760);
d=nilai_random/(double)32760;
/*Membandingkan bilangan random dengan probabilitas
untuk menentukan apakah kondisi tersebut bisa diterima
sebagai kondisi berikutnya*/
if (d<probabilitas)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
if (hitung2==1)
{
E3[hitung1+1]=E1;
E_hitung=0;
}
else
{
n=n+1;
E3[hitung1+1]=E3[hitung1-(n-1)];
E_hitung=0;
}
}
}
}
Etotal=Etotal+E_hitung;
Ekuadrat=Ekuadrat+(E_hitung*E_hitung);
}

rata_sweep_energi=Etotal/nsweep;
rata_sweep_kuadrat=Ekuadrat/nsweep;
sum_sweep_energi=sum_sweep_energi+rata_sweep_energi;

```

```

        sum_sweep_kuadrat=sum_sweep_kuadrat+rata_sweep_kuadrat;
    }
    rata_group_energi=sum_sweep_energi/ngroup;
    rata_rata_kuadrat=sum_sweep_kuadrat/ngroup;
    kuadrat_rata_rata=rata_group_energi*rata_group_energi;
/* c=<math>E^2</math>-<math>E^2/kb*T*T </math> */
specific_heat[ivar]=(rata_rata_kuadrat-kuadrat_rata_rata)/(kb*T[ivar]*T[ivar]);
    }
    while (pil1=='Y' || pil1=='y')
    {

        tabel();
        printf("press any key ");getch();
        getch();

        printf("Ingin melihat tabel lagi<Y/T>?");
        pil1=getch();
    }
    return 0;
}
void tampilan()
{
    register int i,j;
    clrscr();
    gotoxy(30,2);printf("COMPUTATIONAL PHYSICS\r\n");

    gotoxy(18,5);printf("Departement of Physics,University of Indonesia\r\n");
    gotoxy(10,6);printf("\xC9");
    for (i=1;i<=59;++i)
        printf("\xCD");
        printf("\xBB\r\n");
        for (j=7;j<=20;++j)
        {
            gotoxy(10,j);printf("\xBA");
            gotoxy(70,j);printf("\xBA\r\n");
        }
        gotoxy(10,21);
        printf("\xCB");
        for (i=1;i<=59;++i)
            printf("\xCD");
            printf("\xBC\r\n");
            gotoxy(33,8);printf("Copyright 2009\r\n");
            gotoxy(30,10);printf("ASYNNNI MODEL\r\n");
            gotoxy(22,11);printf("Using The Metropolis Algorithm\r\n");
            gotoxy(36,14);printf("*****\r\n");
            gotoxy(22,17);printf("Type ctrl-break to stop at a promt.\r\n");
            gotoxy(28,19);
            printf("Press any key to comtinue...");
            getch();
    }
    void input_parameter()
    {
        clrscr();
        gotoxy(20,10);
        printf("Arah_x,Nx (<=95) = ");
        scanf("%d",&x);
        while(x<=0 || x>95)
        {

```

```

        gotoxy(20,10);
        printf("Arah_x, Nx (<=95) = ");
        scanf("%d",&x);
    }
    gotoxy(20,12);
    printf("Arah_y , Ny (<=95) = ");
    scanf("%d",&y);
    while (y<=0 || y>95)
    {
        gotoxy(20,12);
        printf("Arah_y, Ny (<=95) = ");
        scanf("%d",&y);
    }
    gotoxy(20,14);
    printf("Arah_z , Nz (<=95) = ");
    scanf("%d",&z);
    while (z<=0 || z>95)
    {
        gotoxy(29,14);
        printf("Arah_z , Nz (<=95) = ");
        scanf("%d",&z);
    }
    gotoxy(20,16);
    printf("v1 = ");
    scanf("%lf",&v1);
    gotoxy(20,18);
    printf("v2 = ");
    scanf("%lf",&v2);
    gotoxy(20,20);
    printf("v3 = ");
    scanf("%lf",&v3);

    clrscr();
    gotoxy(20,12);
    printf("Jumlah group, ngroup (>0) = ");
    scanf("%d",&ngroup);
    while(ngroup<=0)
    {
        gotoxy(20,12);
        printf("Jumlah group, ngroup (>0) = ");
        scanf("%d",&ngroup);
    }
    gotoxy(20,14);
    printf("Jumlah sweep , nsweep(>0) = ");
    scanf("%d",&nsweep);
    while(nsweep<=0)
    {
        gotoxy(20,14);
        printf("Jumlah sweep , nsweep(>0) = ");
        scanf("%d",&nsweep);
    }
    printf("Besarx = ");scanf("%d",&Besarx);
    }
    void temperatur()
    {
        clrscr();
        gotoxy(27,9);printf("Temp awal =");scanf("%lf",&Tawal);
        gotoxy(27,11);printf("delta temp=");scanf("%lf",&delt);

```

```

gotoxy(27,13);printf("Jumlah variasi=");scanf("%d",&nvar);
}
void kisi()
{
for (i2=0;i2<=x;++i2)
{
for (j2=0;j2<=y;++j2)
{
for( k2=0;k2<=z;++k2)
{
nilai_random = random(32760);
switch(Besarx)
{
case 0:
if (nilai_random<32766)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;

/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/
case 6:
if(nilai_random<13106)

```



```

sumF=sumF+F;
sumG=sumG+G;
sumP=sumP+P;

E1=v1*sumF-v2*sumG+v3*sumP;
}
}
}
void kisi_berikut()
{
for (i5=0;i5<=x;++i5)
{
for (j5=0;j5<=y;++j5)
{
for( k5=0;k5<=z;++k5)
{
nilai_random = random(32760);
switch(Besarx)
{
case 0:
if (nilai_random<32766)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;

/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:

```



```

/* Interaksi atom oksigen dengan tetangganya*/
F6=S[i6][j6][k6]*(S[i6p][j6][k6]+S[i6m][j6][k6]+S[i6][j6p][k6]+S[i6][j6m][k6]);

G6=S[i6][j6][k6]*(S[i6m][j6m][k6]+S[i6p][j6p][k6]);

P6=S[i6][j6][k6]*(S[i6p][j6m][k6]+S[i6m][j6p][k6]);

    sumF6=sumF6+F6;
    sumG6=sumG6+G6;
    sumP6=sumP6+P6;

    E6=v1*sumF6-v2*sumG6+v3*sumP6;
}
}
}
void tabel()
{
    register int i;
    clrscr();
    gotoxy(1,5);
    printf("Nx=%2d Ny=%2d Nz=%2d \r\n",x,y,z);
    printf("v1=%lf\r\n",v1);
    printf("v2=%lf\r\n",v2);
    printf("v3=%lf\r\n",v3);

    printf("ngroup=%d\r\n",ngroup);
    printf("nsweep=%d\r\n",nsweep);
    printf("kb=%lg\r\n",kb);
    printf("Besarx=%d\r\n",Besarx);
    wt=time('\0)-w1;
    printf("\r\nwaktu=%d detik\r\n",wt);

    gotoxy(19,1);printf("\xC9");
    for(i=1;i<=52;++i)
    printf("\xCD");
    gotoxy(22,1);printf("\xCB");
    gotoxy(35,1);printf("\xCB");
    gotoxy(55,1);printf("\xCB");
    gotoxy(72,1);printf("\xBB\r\n");
    gotoxy(19,2);printf("\xBA");printf("\NO");printf("\xBA");
    gotoxy(26,2);printf("T");gotoxy(35,2);printf("\xBA");
    gotoxy(46,2);printf("Cv");gotoxy(55,2);printf("\xBA");
    gotoxy(64,2);printf("E");gotoxy(72,2);printf("\xBA");
    gotoxy(19,3);printf("\xCC");
    for(i=1;i<=52;++i)
    printf("\xCD");
    gotoxy(22,3);printf("\xCE");
    gotoxy(35,3);printf("\xCE");
    gotoxy(55,3);printf("\xCE");
    gotoxy(72,3);printf("\xB9\r\n");
    for(i=4;i<=(nvar+3);++i)
    {
    gotoxy(19,i);printf("\xBA");printf("%d",i-3);
    gotoxy(22,i);printf("\xBA");
    if(i>4)

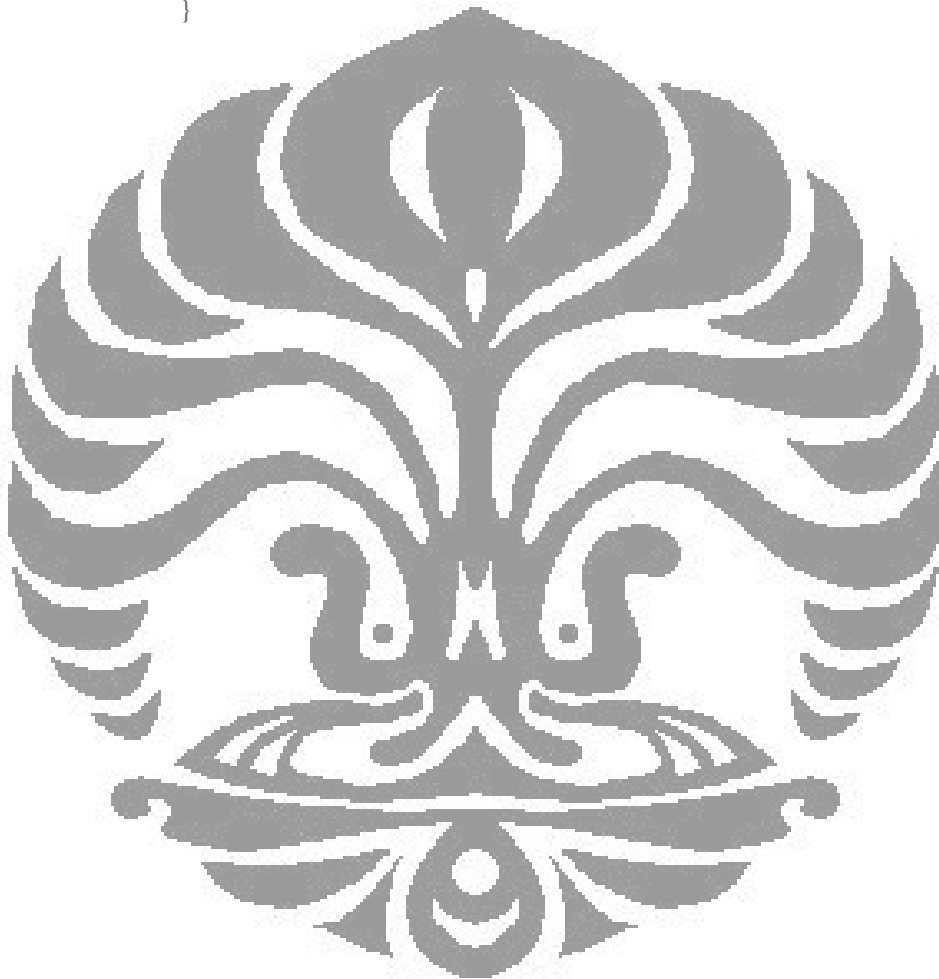
```



```

gotoxy(23,(nvar+7-i));printf("%+1.41f",T[nvar+4-i]);
gotoxy(35,(nvar+7-i));printf("\xBA");
gotoxy(40,(nvar+7-i));printf("%+1.6f",specific_heat[i-4]);
gotoxy(55,(nvar+7-i));printf("\xBA");
gotoxy(72,i);printf("\xBA");
}
gotoxy(19,nvar+4);printf("\xC8");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,nvar+4);printf("\xCA");
gotoxy(35,nvar+4);printf("\xCA");
gotoxy(55,nvar+4);printf("\xCA");
gotoxy(72,nvar+4);printf("\xBC\r\n");
}

```



```

/*-----*/
/* PROGRAM MODEL ASYNNNI TI 2212 */
/* Saepul Qodar S2 Fisika UI */
/* OS : Microsoft Windows XP SP2 */
/* Compiler : Turbo C++ 4.5 */
/* Memory Model : Large */
/*-----*/
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <dos.h>
#include <conio.h>
#include <stdio.h>
int pil=0,
    x=0,
    w=0,
    y=0,
    z=0,
    i1=0,
    j1=0,
    k1=0,
    i2=0,
    j2=0,
    k2=0,
    i5=0,
    j5=0,
    k5=0,
    i6=0,
    j6=0,
    k6=0,
    Besarx=0;
long w1,wt;
int ngroup=0,
    nsweep=0;
double nspin=0,
    probabilitas=0;
double Etotal=0,Ekkuadrat=0,
    rata_group_energi=0,
    rata_group_kuadrat=0;
int jumlahtotal=0,jumlahtatu=0;
double groupe=0,groupe2=0;
double energi_total=0,
    E1=0,
    E1_rata_rata=0,
    E2=0,
    E2_rata_rata=0,
    delta_energi=0,
    E6=0,
    E8_rata_rata=0;
double rata_rata=0,
    rata_rata_kuadrat=0,
    kuadrat=0,
    kuadrat_rata_rata=0,
    rata_sweep_energi=0,
    rata_sweep_kuadrat=0,
    E_hitung=0,
    delta=0,

```

```

        absolut=0,
        pertemperatur=0,
        sum_sweep_kuadrat=0,
        sum_sweep_energi=0,
        sume=0,
        sume2=0;

int nvar=0;
double delt=0,Tawal=0,d=0;
double huge spesifik_heat[2400],E3[4000],T[25];
double kalor_jenis=0;
int jumlah_random=0,
    nilai_random=0;
int nilai_random2=0;
int sweepke=0,grupke=0;

long huge S[100L][100L][100L];/*jumlah posisi atom oksigen maksimum100x100x100*/
double cv=0,
    cvc=0;
double v1=0,
    v2=0,
    v3=0,
    v4=0;
int hitung2=0,hitung1=0,n=0;
double kb=0,E=0;
char pil1='Y';
void tampilan(void),input_parameter(void),energi_pembanding(void),kisi(void);
void kisi_awal(void),energi_awal(void),kisi_berikut(void),energi_berikut(void);
int hitungE_M(void);
void hasil(void),temperatur(void),kalor(void),tabel(void);
extern int _directvideo=0;
int main(void)
{
    register int isweep,igroup,test,ivar;

    tampilan();
    input_parameter();
    temperatur();
    w1=time('\0');
    randomize();
    /*Perhitungan dengan temperatur yang berbeda-beda*/
    for (ivar=1;ivar<=nvar;++ivar)
    {
        T[1]=Tawal;
        T[ivar+1]=T[ivar]+delt;
        kalor_jenis=0;
        sume=0;
        sume2=0;
        /*Menentukan posisi awal atom dan menghitung energinya*/
        kisi();
        energi_awal();
        gotoxy(20,12);
        printf("Nx=%d Ny=%d Nz=%d \r\n",x,y,z);
        gotoxy(20,20);printf("wait...");
        sum_sweep_kuadrat=0;sum_sweep_energi=0,grupke=0;
        hitung2=0;hitung1=0;E_hitung=0;

        n=0;

```

```

for (igroup=1;igroup<=ngroup;++igroup)
{
groupe=0;groupe2=0;Etotal=0;
Ekuadrat=0;sweepke=0;
for (isweep=1;isweep<=nsweep;++isweep)
{
hitung2=hitung2+1;
hitung1=hitung2-1;
/*Menentukan posisi kedua,ketiga dst atom dan menghitung energinya*/
kisi_berikut();
energi_berikut();
E3[hitung2]=E6;
/*Mencari delta energi,E2-E1,E3-E2,E4-E3 dst*/
if (hitung2==1)
delta=E3[hitung2]-E1;
else
delta=E3[hitung2]-E3[hitung1];
kb=0.00008125;
if (delta <= 0)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
pertemperatur=delta/(2*T[ivar]*kb);
probabilitas=0.5*(1-tanh(pertemperatur));
nilai_random=random(32760);
d=nilai_random/(double)32760;
/*Membandingkan bilangan random dengan probabilitas
untuk menentukan apakah kondisi tersebut bisa diterima
sebagai kondisi berikutnya*/
if (d<probabilitas)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
if (hitung2==1)
{
E3[hitung1+1]=E1;
E_hitung=0;
}
else
{
n=n+1;
E3[hitung1+1]=E3[hitung1-(n-1)];
E_hitung=0;
}
}
}
}
Etotal=Etotal+E_hitung;
Ekuadrat=Ekuadrat+(E_hitung*E_hitung);
}

rata_sweep_energi=Etotal/nsweep;
rata_sweep_kuadrat=Ekuadrat/nsweep;
sum_sweep_energi=sum_sweep_energi+rata_sweep_energi;

```

```

        sum_sweep_kuadrat=sum_sweep_kuadrat+rata_sweep_kuadrat;
    }
    rata_group_energi=sum_sweep_energi/ngroup;
    rata_rata_kuadrat=sum_sweep_kuadrat/ngroup;
    kuadrat_rata_rata=rata_group_energi*rata_group_energi;
/* c=<E^2>-<E>^2/k*T */
specific_heat[ivar]=(rata_rata_kuadrat-kuadrat_rata_rata)/(kb*T[ivar]*T[ivar]);
    }
    while (pil1=='Y' || pil1=='y')
    {

        tabel();
        printf("press any key ");getch();
        getch();

        printf("Ingin melihat tabel lagi<Y/T>?");
        pil1=getch();
    }
    return 0;
}
void tampilan()
{
    register int i,j;
    clrscr();
    gotoxy(30,2);printf("COMPUTATIONAL PHYSICS\r\n");

    gotoxy(18,5);printf("Departement of Physics,University of Indonesia\r\n");
    gotoxy(10,6);printf("\xC9");
    for (i=1;i<=59;++i)
        printf("\xCD");
        printf("\xBB\r\n");
        for (j=7;j<=20;++j)
        {
            gotoxy(10,j);printf("\xBA");
            gotoxy(70,j);printf("\xBA\r\n");
        }
        gotoxy(10,21);
        printf("\xCB");
        for (i=1;i<=59;++i)
            printf("\xCD");
            printf("\xBC\r\n");
            gotoxy(33,8);printf("Copyright 2009\r\n");
            gotoxy(30,10);printf("ASYNNNI MODEL\r\n");
            gotoxy(22,11);printf("Using The Metropolis Algorithm\r\n");
            gotoxy(36,14);printf("*****\r\n");
            gotoxy(22,17);printf("Type ctrl-break to stop at a promt.\r\n");
            gotoxy(28,19);
            printf("Press any key to comtinue...");
            getch();
    }
void input_parameter()
{
    clrscr();
    gotoxy(20,10);
    printf("Arah_x,Nx (<=95) = ");
    scanf("%d",&x);
    while(x<=0 || x>95)
    {

```

```

        gotoxy(20,10);
        printf("Arah_x, Nx (<=95) = ");
        scanf("%d",&x);
    }
    gotoxy(20,12);
    printf("Arah_y , Ny (<=95) = ");
    scanf("%d",&y);
    while (y<=0 || y>95)
    {
        gotoxy(20,12);
        printf("Arah_y, Ny (<=95) = ");
        scanf("%d",&y);
    }
    gotoxy(20,14);
    printf("Arah_z , Nz (<=95) = ");
    scanf("%d",&z);
    while (z<=0 || z>95)
    {
        gotoxy(29,14);
        printf("Arah_z , Nz (<=95) = ");
        scanf("%d",&z);
    }
    gotoxy(20,16);
    printf("v1 = ");
    scanf("%lf",&v1);
    gotoxy(20,18);
    printf("v2 = ");
    scanf("%lf",&v2);
    gotoxy(20,20);
    printf("v3 = ");
    scanf("%lf",&v3);
    gotoxy(20,22);
    printf("v4 = ");
    scanf("%lf",&v4);
    clrscr();
    gotoxy(20,12);
    printf("Jumlah group, ngroup (>0) = ");
    scanf("%d",&ngroup);
    while(ngroup<=0)
    {
        gotoxy(20,12);
        printf("Jumlah group, ngroup (>0) = ");
        scanf("%d",&ngroup);
    }
    gotoxy(20,14);
    printf("Jumlah sweep , nsweep(>0) = ");
    scanf("%d",&nsweep);
    while(nsweep<=0)
    {
        gotoxy(20,14);
        printf("Jumlah sweep , nsweep(>0) = ");
        scanf("%d",&nsweep);
    }
    printf("Besarx = ");scanf("%d",&Besarx);
    }
    void temperatur()
    {
        clrscr();

```

```

gotoxy(27,9);printf("Temp awal =");scanf("%lf",&Tawal);
gotoxy(27,11);printf("delta temp=");scanf("%lf",&delt);
gotoxy(27,13);printf("Jumlah variasi=");scanf("%d",&nvar);
}
void kisi()
{
for (i2=0;i2<=x;++i2)
{
for (j2=0;j2<=y;++j2)
{
for( k2=0;k2<=z;++k2)
{
nilai_random = random(32760);
switch(Besarx)
{
case 0:
if (nilai_random<32766)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/

```

```

case 6:
if(nilai_random<13106)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 30% posisi atom oksigen terisi (x=0.7)*/
case 7:
if(nilai_random<9829)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 20% posisi atom oksigen terisi (x=0.8)*/
case 8:
if(nilai_random<6553)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 10% posisi atom oksigen terisi (x=0.9)*/
case 9:
if(nilai_random<3277)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
}
}
}
}
void energi_awal()
{
double F=0,G=0,P=0,GP=0;
double sumF=0,sumG=0,sumP=0,sumGP=0;
int i1m=0,i1p=0,
j1m=0,j1p=0,
k1m=0,k1p=0;
int loop;
for(i1=0;i1<=x;++i1)
{
if(i1<x) i1p=i1+1;else i1p=1;
if(i1>1) i1m=i1-1;else i1m=x;
for(j1=0;j1<=y;++j1)
{
if(j1<y) j1p=j1+1;else j1p=1;
if(j1>1) j1m=j1-1;else j1m=y;
for(k1=0;k1<=z;++k1)
{
if(k1<z) k1p=k1+1;else k1p=1;
if(k1>1) k1m=k1-1;else k1m=z;
}
}
}
}
}
}
/* Interaksi atom oksigen dengan tetangganya*/

F=S[i1][j1][k1]*(S[i1p][j1][k1]+S[i1m][j1][k1]+S[i1][j1p][k1]+S[i1][j1m][k1])
+S[i1][j1][k1p]*(S[i1p][j1][k1p]+S[i1m][j1][k1p]+S[i1][j1p][k1p]+S[i1][j1m][k1p]);

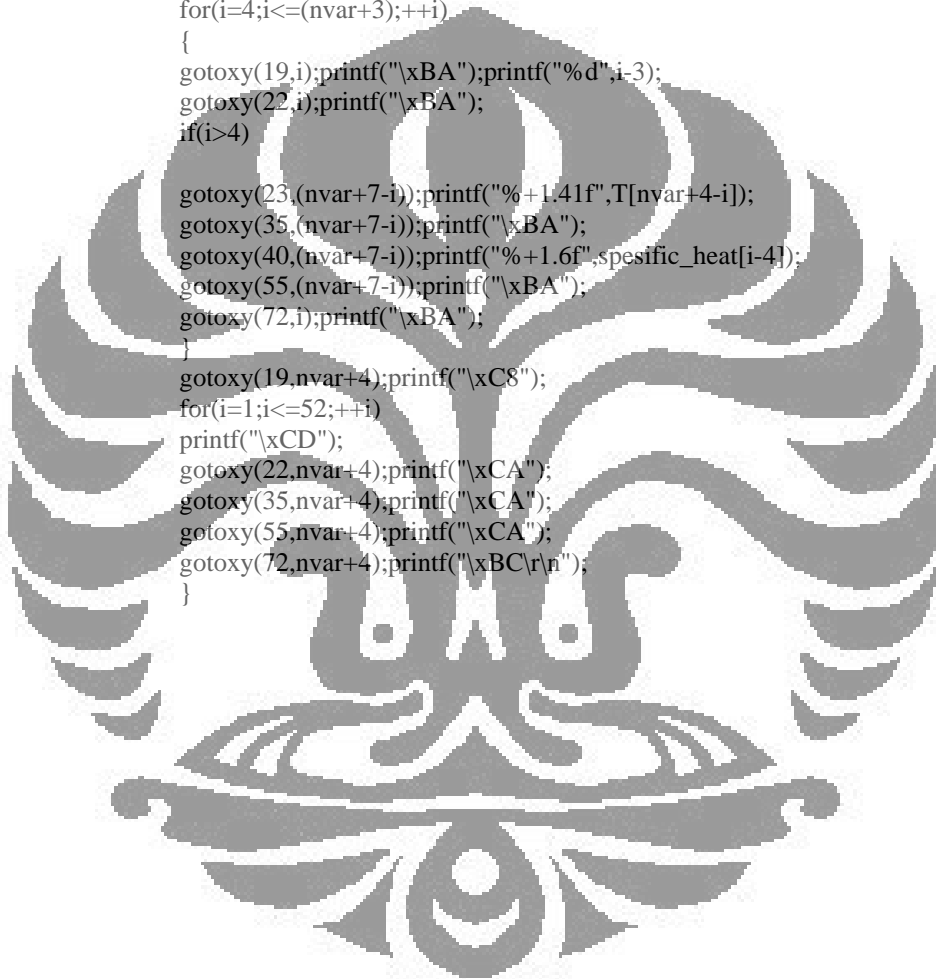
```



```

gotoxy(55,1);printf("\xCB");
gotoxy(72,1);printf("\xBB\r\n");
gotoxy(19,2);printf("\xBA");printf("\NO");printf("\xBA");
gotoxy(26,2);printf("T");gotoxy(35,2);printf("\xBA");
gotoxy(46,2);printf("Cv");gotoxy(55,2);printf("\xBA");
gotoxy(64,2);printf("E");gotoxy(72,2);printf("\xBA");
gotoxy(19,3);printf("\xCC");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,3);printf("\xCE");
gotoxy(35,3);printf("\xCE");
gotoxy(55,3);printf("\xCE");
gotoxy(72,3);printf("\xB9\r\n");
for(i=4;i<=(nvar+3;++i)
{
gotoxy(19,i);printf("\xBA");printf("%d",i-3);
gotoxy(22,i);printf("\xBA");
if(i>4)
gotoxy(23,(nvar+7-i));printf("%+1.4f",T[nvar+4-i]);
gotoxy(35,(nvar+7-i));printf("\xBA");
gotoxy(40,(nvar+7-i));printf("%+1.6f",specific_heat[i-4]);
gotoxy(55,(nvar+7-i));printf("\xBA");
gotoxy(72,i);printf("\xBA");
}
gotoxy(19,nvar+4);printf("\xC8");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,nvar+4);printf("\xCA");
gotoxy(35,nvar+4);printf("\xCA");
gotoxy(55,nvar+4);printf("\xCA");
gotoxy(72,nvar+4);printf("\xBC\r\n");
}

```



```

/*-----*/
/* PROGRAM MODEL ASYNNNI TI 2223 */
/* Saepul Qodar S2 Fisika UI */
/* OS : Microsoft Windows XP SP2 */
/* Compiler : Turbo C++ 4.5 */
/* Memory Model : Large */
/*-----*/
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <dos.h>
#include <conio.h>
#include <stdio.h>
int pil=0,
    x=0,
    w=0,
    y=0,
    z=0,
    i1=0,
    j1=0,
    k1=0,
    i2=0,
    j2=0,
    k2=0,
    i5=0,
    j5=0,
    k5=0,
    i6=0,
    j6=0,
    k6=0,
    Besarx=0;
long w1,wt;
int ngroup=0,
    nsweep=0;
double nspin=0;
double probabilitas=0;
double Etotal=0,Ekuadrat=0,
    rata_group_energi=0,
    rata_group_kuadrat=0;
int jumlahnol=0,jumlahsatu=0;
double groupe=0,groupe2=0;
double energi_total=0,
    E1=0,
    E1_rata_rata=0,
    E2=0,
    E2_rata_rata=0,
    delta_energi=0,
    E6=0,
    E8_rata_rata=0;
double rata_rata=0,
    rata_rata_kuadrat=0,
    kuadrat=0,
    kuadrat_rata_rata=0,
    rata_sweep_energi=0,
    rata_sweep_kuadrat=0,
    E_hitung=0,
    delta=0,

```

```

        absolut=0,
        pertemperatur=0,
        sum_sweep_kuadrat=0,
        sum_sweep_energi=0,
        sume=0,
        sume2=0;

int nvar=0;
double delt=0,Tawal=0,d=0;
double huge spesifik_heat[2400],E3[4000],T[25];
double kalor_jenis=0;
int jumlah_random=0,
    nilai_random=0;
int nilai_random2=0;
int sweepke=0,grupke=0;

long huge S[100L][100L][100L];/*jumlah posisi atom oksigen maksimum 100x100x100*/
double cv=0,
    cvc=0;
double v1=0,
    v2=0,
    v3=0,
    v4=0;
int hitung2=0,hitung1=0,n=0;
double kb=0,E=0;
char pil1='Y';
void tampilan(void),input_parameter(void),energi_pembanding(void),kisi(void);
void kisi_awal(void),energi_awal(void),kisi_berikut(void),energi_berikut(void);
int hitungE_M(void);
void hasil(void),temperatur(void),kalor(void),tabel(void);
extern int _directvideo=0;
int main(void)
{
    register int isweep,igroup,test,ivar;

    tampilan();
    input_parameter();
    temperatur();
    w1=time('\0');
    randomize();
    /*Perhitungan dengan temperatur yang berbeda-beda*/
    for (ivar=1;ivar<=nvar;++ivar)
    {
        T[1]=Tawal;
        T[ivar+1]=T[ivar]+delt;
        kalor_jenis=0;
        sume=0;
        sume2=0;

        /*Menentukan posisi awal atom dan menghitung energinya*/
        kisi();
        energi_awal();
        gotoxy(20,12);
        printf("Nx=%d Ny=%d Nz=%d \r\n",x,y,z);
        gotoxy(20,20);printf("wait...");
        sum_sweep_kuadrat=0;sum_sweep_energi=0,grupke=0;
        hitung2=0;hitung1=0;E_hitung=0;

        n=0;

```

```

for (igroup=1;igroup<=ngroup;++igroup)
{
groupe=0;groupe2=0;Etotal=0;
Ekuadrat=0;sweepke=0;
for (isweep=1;isweep<=nsweep;++isweep)
{
hitung2=hitung2+1;
hitung1=hitung2-1;
/*Menentukan posisi kedua,ketiga dst atom dan menghitung energinya*/
kisi_berikut();
energi_berikut();
E3[hitung2]=E6;
/*Mencari delta energi,E2-E1,E3-E2,E4-E3 dst*/
if (hitung2==1)
delta=E3[hitung2]-E1;
else
delta=E3[hitung2]-E3[hitung1];
kb=0.00008125;
if (delta <= 0)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
pertemperatur=delta/(2*T[ivar]*kb);
probabilitas=0.5*(1-tanh(pertemperatur));
nilai_random=random(32760);
d=nilai_random/(double)32760;
/*Membandingkan bilangan random dengan probabilitas
untuk menentukan apakah kondisi tersebut bisa diterima
sebagai kondisi berikutnya*/
if (d<probabilitas)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
if (hitung2==1)
{
E3[hitung1+1]=E1;
E_hitung=0;
}
else
{
n=n+1;
E3[hitung1+1]=E3[hitung1-(n-1)];
E_hitung=0;
}
}
}
}
Etotal=Etotal+E_hitung;
Ekuadrat=Ekuadrat+(E_hitung*E_hitung);
}

rata_sweep_energi=Etotal/nsweep;
rata_sweep_kuadrat=Ekuadrat/nsweep;
sum_sweep_energi=sum_sweep_energi+rata_sweep_energi;

```

```

sum_sweep_kuadrat=sum_sweep_kuadrat+rata_sweep_kuadrat;
}
rata_group_energi=sum_sweep_energi/ngroup;
rata_rata_kuadrat=sum_sweep_kuadrat/ngroup;
kuadrat_rata_rata=rata_group_energi*rata_group_energi;
/* c=<E^2>-<E>^2/k*T */
specific_heat[ivar]=(rata_rata_kuadrat-kuadrat_rata_rata)/(kb*T[ivar]*T[ivar]);
}
while (pil1=='Y' || pil1=='y')
{

    tabel();
    printf("press any key ");getch();
    getch();

    printf("Ingin melihat tabel lagi<Y/T>?");
    pil1=getch();
}
return 0;
}
void tampilan()
{
    register int i,j;
    clrscr();
    gotoxy(30,2);printf("COMPUTATIONAL PHYSICS\r\n");

    gotoxy(18,5);printf("Departement of Physics, University of Indonesia\r\n");
    gotoxy(10,6);printf("\xC9");
    for (i=1;i<=59;++i)
        printf("\xCD");
        printf("\xBB\r\n");
        for (j=7;j<=20;++j)
        {
            gotoxy(10,j);printf("\xBA");
            gotoxy(70,j);printf("\xBA\r\n");
        }
        gotoxy(10,21);
        printf("\xCB");
        for (i=1;i<=59;++i)
            printf("\xCD");
            printf("\xBC\r\n");
            gotoxy(33,8);printf("Copyright 2009\r\n");
            gotoxy(30,10);printf("ASYNNNI MODEL\r\n");
            gotoxy(22,11);printf("Using The Metropolis Algorithm\r\n");
            gotoxy(36,14);printf("*****\r\n");
            gotoxy(22,17);printf("Type ctrl-break to stop at a promt.\r\n");
            gotoxy(28,19);
            printf("Press any key to comtinue...");
            getch();
}
void input_parameter()
{
    clrscr();
    gotoxy(20,10);
    printf("Arah_x,Nx (<=95) = ");
    scanf("%d",&x);
    while(x<=0 || x>95)
    {

```



```

        gotoxy(20,10);
        printf("Arah_x, Nx (<=95) = ");
        scanf("%d",&x);
    }
    gotoxy(20,12);
    printf("Arah_y, Ny (<=95) = ");
    scanf("%d",&y);
    while (y<=0 || y>95)
    {
        gotoxy(20,12);
        printf("Arah_y, Ny (<=95) = ");
        scanf("%d",&y);
    }
    gotoxy(20,14);
    printf("Arah_z, Nz (<=95) = ");
    scanf("%d",&z);
    while (z<=0 || z>95)
    {
        gotoxy(29,14);
        printf("Arah_z, Nz (<=95) = ");
        scanf("%d",&z);
    }
    gotoxy(20,16);
    printf("v1= ");
    scanf("%lf",&v1);
    gotoxy(20,18);
    printf("v2= ");
    scanf("%lf",&v2);
    gotoxy(20,20);
    printf("v3= ");
    scanf("%lf",&v3);
    gotoxy(20,22);
    printf("v4= ");
    scanf("%lf",&v4);
    clrscr();
    gotoxy(20,12);
    printf("Jumlah group, ngroup (>0) = ");
    scanf("%d",&ngroup);
    while(ngroup<=0)
    {
        gotoxy(20,12);
        printf("Jumlah group, ngroup (>0) = ");
        scanf("%d",&ngroup);
    }
    gotoxy(20,14);
    printf("Jumlah sweep , nsweep(>0) = ");
    scanf("%d",&nsweep);
    while(nsweep<=0)
    {
        gotoxy(20,14);
        printf("Jumlah sweep , nsweep(>0) = ");
        scanf("%d",&nsweep);
    }
    printf("Besarx = ");scanf("%d",&Besarx);
    }
    void temperatur()
    {
        clrscr();

```

```

gotoxy(27,9);printf("Temp awal =");scanf("%lf",&Tawal);
gotoxy(27,11);printf("delta temp=");scanf("%lf",&delt);
gotoxy(27,13);printf("Jumlah variasi=");scanf("%d",&nvar);
}
void kisi()
{
for (i2=0;i2<=x;++i2)
{
for (j2=0;j2<=y;++j2)
{
for( k2=0;k2<=z;++k2)
{
nilai_random = random(32760);
switch(Besarx)
{
case 0:
if (nilai_random<32766)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/

```



```
G=S[i1][j1][k1]*(S[i1m][j1m][k1]+S[i1p][j1p][k1])
+S[i1][j1][k1p]*(S[i1m][j1m][k1p]+S[i1p][j1p][k1p])
+S[i1][j1][k1m]*(S[i1m][j1m][k1m]+S[i1p][j1p][k1m]);
```

```
P=S[i1][j1][k1]*(S[i1p][j1p][k1]+S[i1m][j1p][k1])
+S[i1][j1][k1p]*(S[i1p][j1p][k1p]+S[i1m][j1p][k1p])
+S[i1][j1][k1m]*(S[i1p][j1p][k1m]+S[i1m][j1p][k1m]);
```

```
GP=S[i1][j1][k1]*(S[i1m][j1m][k1]+S[i1p][j1p][k1])
+S[i1][j1][k1p]*(S[i1m][j1m][k1p]+S[i1p][j1p][k1p])
+S[i1][j1][k1m]*(S[i1m][j1m][k1m]+S[i1p][j1p][k1m])
+S[i1][j1][k1]*(S[i1p][j1p][k1]+S[i1m][j1p][k1])
+S[i1][j1][k1p]*(S[i1p][j1p][k1p]+S[i1m][j1p][k1p])
+S[i1][j1][k1m]*(S[i1p][j1p][k1m]+S[i1m][j1p][k1m]);
```

```

sumF=sumF+F;
sumG=sumG+G;
sumP=sumP+P;
sumGP=sumGP+GP;
E1=v1*sumF-v2*sumG+v3*sumP-v4*sumGP-0.4012;
}
}
}
void kisi_berikut()
{
for (i5=0;i5<=x;++i5)
{
for (j5=0;j5<=y;++j5)
{
for( k5=0;k5<=z;++k5)
{
nilai_random = random(32760);
switch(Besarx)
{
case 0:
if (nilai_random<32766)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;

/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/

```

```

case 3:
if(nilai_random<22936)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/
case 6:
if(nilai_random<13106)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 30% posisi atom oksigen terisi (x=0.7)*/
case 7:
if(nilai_random<9829)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 20% posisi atom oksigen terisi (x=0.8)*/
case 8:
if(nilai_random<6553)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 10% posisi atom oksigen terisi (x=0.9)*/
case 9:
if(nilai_random<3277)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
}
}
}
}
}
void energi_berikut()
{
double F6=0,G6=0,P6=0,GP6=0;
double sumF6=0,sumG6=0,sumP6=0,sumGP6=0;
int i6m=0,i6p=0,

```

```

        j6m=0,j6p=0,
        k6m=0,k6p=0;
int loop;
for(i6=0;i6<=x;++i6)
{
    if(i6<x) i6p=i6+1;else i6p=1;
    if(i6>1) i6m=i6-1;else i6m=x;
    for(j6=0;j6<=y;++j6)
    {
        if(j6<y) j6p=j6+1;else j6p=1;
        if(j6>1) j6m=j6-1;else j6m=y;
        for(k6=0;k6<=z;++k6)
        {
            if(k6<z) k6p=k6+1;else k6p=1;
            if(k6>1) k6m=k6-1;else k6m=z;

/* Interaksi atom oksigen dengan tetangganya*/

F6=S[i6][j6][k6]*(S[i6p][j6][k6]+S[i6m][j6][k6]+S[i6][j6p][k6]+S[i6][j6m][k6])
+S[i6][j6][k6p]*(S[i6p][j6][k6p]+S[i6m][j6][k6p]+S[i6][j6p][k6p]+S[i6][j6m][k6p])
+S[i6][j6][k6m]*(S[i6p][j6][k6m]+S[i6m][j6][k6m]+S[i6][j6p][k6m]+S[i6][j6m][k6m]);

G6=S[i6][j6][k6]*(S[i6m][j6m][k6]+S[i6p][j6p][k6])
+S[i6][j6][k6p]*(S[i6m][j6m][k6p]+S[i6p][j6p][k6p])
+S[i6][j6][k6m]*(S[i6m][j6m][k6m]+S[i6p][j6p][k6m]);

P6=S[i6][j6][k6]*(S[i6p][j6m][k6]+S[i6m][j6p][k6])
+S[i6][j6][k6p]*(S[i6p][j6m][k6p]+S[i6m][j6p][k6p])
+S[i6][j6][k6m]*(S[i6p][j6m][k6m]+S[i6m][j6p][k6m]);

GP6=S[i6][j6][k6]*(S[i6m][j6m][k6]+S[i6p][j6p][k6])
+S[i6][j6][k6p]*(S[i6m][j6m][k6p]+S[i6p][j6p][k6p])
+S[i6][j6][k6m]*(S[i6m][j6m][k6m]+S[i6p][j6p][k6m])
+S[i6][j6][k6]*(S[i6p][j6m][k6]+S[i6m][j6p][k6])
+S[i6][j6][k6p]*(S[i6p][j6m][k6p]+S[i6m][j6p][k6p])
+S[i6][j6][k6m]*(S[i6p][j6m][k6m]+S[i6m][j6p][k6m]);

        sumF6=sumF6+F6;
        sumG6=sumG6+G6;
        sumP6=sumP6+P6;
        sumGP6=sumGP6+GP6;
        E6=v1*sumF6-v2*sumG6+v3*sumP6-v4*sumGP6-0.4012;
    }
}
}
void tabel()
{
    register int i;
    clrscr();
    gotoxy(1,5);
    printf("Nx=%2d Ny=%2d Nz=%2d \r\n" ,x,y,z);
    printf("v1=%lf\r\n",v1);
    printf("v2=%lf\r\n",v2);
    printf("v3=%lf\r\n",v3);
    printf("v4=%lf\r\n",v4);
    printf("ngroup=%d\r\n",ngroup);
    printf("nsweep=%d\r\n",nsweep);
}

```

```

printf("kb=%lg\r\n",kb);
printf("Besarx=%d\r\n",Besarx);
wt=time('\0)-w1;
printf("\r\nwaktu=%d detik\r\n",wt);

gotoxy(19,1);printf("\xC9");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,1);printf("\xCB");
gotoxy(35,1);printf("\xCB");
gotoxy(55,1);printf("\xCB");
gotoxy(72,1);printf("\xBB\r\n");
gotoxy(19,2);printf("\xBA");printf("\NO");printf("\xBA");
gotoxy(26,2);printf("T");gotoxy(35,2);printf("\xBA");
gotoxy(46,2);printf("Cv");gotoxy(55,2);printf("\xBA");
gotoxy(64,2);printf("E");gotoxy(72,2);printf("\xBA");
gotoxy(19,3);printf("\xCC");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,3);printf("\xCE");
gotoxy(35,3);printf("\xCE");
gotoxy(55,3);printf("\xCE");
gotoxy(72,3);printf("\xB9\r\n");
for(i=4;i<=(nvar+3);++i)
{
gotoxy(19,i);printf("\xBA");printf("%d",i-3);
gotoxy(22,i);printf("\xBA");
if(i>4)

gotoxy(23,(nvar+7-i));printf("%+1.4f",T[nvar+4-i]);
gotoxy(35,(nvar+7-i));printf("\xBA");
gotoxy(40,(nvar+7-i));printf("%+1.6f",specific_heat[i-4]);
gotoxy(55,(nvar+7-i));printf("\xBA");
gotoxy(72,i);printf("\xBA");
}
gotoxy(19,nvar+4);printf("\xC8");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,nvar+4);printf("\xCA");
gotoxy(35,nvar+4);printf("\xCA");
gotoxy(55,nvar+4);printf("\xCA");
gotoxy(72,nvar+4);printf("\xBC\r\n");
}

```

```

/*-----*/
/* PROGRAM MODEL ASYNNNI TIO bilayers 4 bidang CuO */
/* Saepul Qodar S2 Fisika UI */
/* OS : Microsoft Windows XP SP2 */
/* Compiler : Turbo C++ 4.5 */
/* Memory Model : Large */
/*-----*/
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <dos.h>
#include <conio.h>
#include <stdio.h>
int pil=0,
    x=0,
    w=0,
    y=0,
    z=0,
    i1=0,
    j1=0,
    k1=0,
    i2=0,
    j2=0,
    k2=0,
    i5=0,
    j5=0,
    k5=0,
    i6=0,
    j6=0,
    k6=0,
    Besarx=0;
long w1,wt;
int ngroup=0,
    nsweep=0;
double nspin=0;
double probabilitas=0;
double Etotal=0,Ekuadrat=0,
    rata_group_energi=0,
    rata_group_kuadrat=0;
int jumlahnol=0,jumlahsatu=0;
double groupe=0,groupe2=0;
double energi_total=0,
    E1=0,
    E1_rata_rata=0,
    E2=0,
    E2_rata_rata=0,
    delta_energi=0,
    E6=0,
    E8_rata_rata=0;
double rata_rata=0,
    rata_rata_kuadrat=0,
    kuadrat=0,
    kuadrat_rata_rata=0,
    rata_sweep_energi=0,
    rata_sweep_kuadrat=0,
    E_hitung=0,
    delta=0,

```



```

        absolut=0,
        pertemperatur=0,
        sum_sweep_kuadrat=0,
        sum_sweep_energi=0,
        sume=0,
        sume2=0;

int nvar=0;
double delt=0,Tawal=0,d=0;
double huge spesifik_heat[2400],E3[4000],T[25];
double kalor_jenis=0;
int jumlah_random=0,
    nilai_random=0;
int nilai_random2=0;
int sweepke=0,grupke=0;

long huge S[100L][100L][100L];/*jumlah posisi atom oksigen maksimum 64x64x3*/
double cv=0,
    cvc=0;
double v1=0,
    v2=0,
    v3=0,
    v4=0;
int hitung2=0,hitung1=0,n=0;
double kb=0,E=0;
char pil1='Y';
void tampilan(void),input_parameter(void),energi_pembanding(void),kisi(void);
void kisi_awal(void),energi_awal(void),kisi_berikut(void),energi_berikut(void);
int hitungE_M(void);
void hasil(void),temperatur(void),kalor(void),tabel(void);
extern int _directvideo=0;
int main(void)
{
    register int isweep,igroup,test,ivar;

    tampilan();
    input_parameter();
    temperatur();
    w1=time('\0');
    randomize();
    /*Perhitungan dengan temperatur yang berbeda-beda*/
    for (ivar=1;ivar<=nvar;++ivar)
    {
        T[1]=Tawal;
        T[ivar+1]=T[ivar]+delt;
        kalor_jenis=0;
        sume=0;
        sume2=0;

        /*Menentukan posisi awal atom dan menghitung energinya*/
        kisi();
        energi_awal();
        gotoxy(20,12);
        printf("Nx=%d Ny=%d Nz=%d \r\n",x,y,z);
        gotoxy(20,20);printf("wait...");
        sum_sweep_kuadrat=0;sum_sweep_energi=0,grupke=0;
        hitung2=0;hitung1=0;E_hitung=0;

        n=0;

```

```

for (igroup=1;igroup<=ngroup;++igroup)
{
groupe=0;groupe2=0;Etotal=0;
Ekuadrat=0;sweepke=0;
for (isweep=1;isweep<=nsweep;++isweep)
{
hitung2=hitung2+1;
hitung1=hitung2-1;
/*Menentukan posisi kedua,ketiga dst atom dan menghitung energinya*/
kisi_berikut();
energi_berikut();
E3[hitung2]=E6;
/*Mencari delta energi,E2-E1,E3-E2,E4-E3 dst*/
if (hitung2==1)
delta=E3[hitung2]-E1;
else
delta=E3[hitung2]-E3[hitung1];
kb=0.00008125;
if (delta <= 0)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
pertemperatur=delta/(2*T[ivar]*kb);
probabilitas=0.5*(1-tanh(pertemperatur));
nilai_random=random(32760);
d=nilai_random/(double)32760;
/*Membandingkan bilangan random dengan probabilitas
untuk menentukan apakah kondisi tersebut bisa diterima
sebagai kondisi berikutnya*/
if (d<probabilitas)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
if (hitung2==1)
{
E3[hitung1+1]=E1;
E_hitung=0;
}
else
{
n=n+1;
E3[hitung1+1]=E3[hitung1-(n-1)];
E_hitung=0;
}
}
}
}
Etotal=Etotal+E_hitung;
Ekuadrat=Ekuadrat+(E_hitung*E_hitung);
}

rata_sweep_energi=Etotal/nsweep;
rata_sweep_kuadrat=Ekuadrat/nsweep;
sum_sweep_energi=sum_sweep_energi+rata_sweep_energi;

```

```

        sum_sweep_kuadrat=sum_sweep_kuadrat+rata_sweep_kuadrat;
    }
    rata_group_energi=sum_sweep_energi/ngroup;
    rata_rata_kuadrat=sum_sweep_kuadrat/ngroup;
    kuadrat_rata_rata=rata_group_energi*rata_group_energi;
/* c=<E^2>-<E>^2/k*T */
specific_heat[ivar]=(rata_rata_kuadrat-kuadrat_rata_rata)/(kb*T[ivar]*T[ivar]);
    }
    while (pil1=='Y' || pil1=='y')
    {

        tabel();
        printf("press any key ");getch();
        getch();

        printf("Ingin melihat tabel lagi<Y/T>?");
        pil1=getch();
    }
    return 0;
}
void tampilan()
{
    register int i,j;
    clrscr();
    gotoxy(30,2);printf("COMPUTATIONAL PHYSICS\r\n");

    gotoxy(18,5);printf("Departement of Physics,University of Indonesia\r\n");
    gotoxy(10,6);printf("\xC9");
    for (i=1;i<=59;++i)
        printf("\xCD");
        printf("\xBB\r\n");
        for (j=7;j<=20;++j)
        {
            gotoxy(10,j);printf("\xBA");
            gotoxy(70,j);printf("\xBA\r\n");
        }
        gotoxy(10,21);
        printf("\xCB");
        for (i=1;i<=59;++i)
            printf("\xCD");
            printf("\xBC\r\n");
            gotoxy(33,8);printf("Copyright 2006\r\n");
            gotoxy(30,10);printf("ASYNNNI MODEL\r\n");
            gotoxy(22,11);printf("Using The Metropolis Algorithm\r\n");
            gotoxy(36,14);printf("*****\r\n");
            gotoxy(22,17);printf("Type ctrl-break to stop at a promt.\r\n");
            gotoxy(28,19);
            printf("Press any key to comtinue...");
            getch();
    }
void input_parameter()
{
    clrscr();
    gotoxy(20,10);
    printf("Arah_x,Nx (<=95) = ");
    scanf("%d",&x);
    while(x<=0 || x>95)
    {

```

```

        gotoxy(20,10);
        printf("Arah_x, Nx (<=95) = ");
        scanf("%d",&x);
    }
    gotoxy(20,12);
    printf("Arah_y , Ny (<=95) = ");
    scanf("%d",&y);
    while (y<=0 || y>95)
    {
        gotoxy(20,12);
        printf("Arah_y, Ny (<=95) = ");
        scanf("%d",&y);
    }
    gotoxy(20,14);
    printf("Arah_z , Nz (<=95) = ");
    scanf("%d",&z);
    while (z<=0 || z>95)
    {
        gotoxy(29,14);
        printf("Arah_z , Nz (<=95) = ");
        scanf("%d",&z);
    }
    gotoxy(20,16);
    printf("v1= ");
    scanf("%lf",&v1);
    gotoxy(20,18);
    printf("v2= ");
    scanf("%lf",&v2);
    gotoxy(20,20);
    printf("v3= ");
    scanf("%lf",&v3);
    gotoxy(20,22);
    printf("v4= ");
    scanf("%lf",&v4);
    clrscr();
    gotoxy(20,12);
    printf("Jumlah group, ngroup (>0) = ");
    scanf("%d",&ngroup);
    while(ngroup<=0)
    {
        gotoxy(20,12);
        printf("Jumlah group, ngroup (>0) = ");
        scanf("%d",&ngroup);
    }
    gotoxy(20,14);
    printf("Jumlah sweep , nsweep(>0) = ");
    scanf("%d",&nsweep);
    while(nsweep<=0)
    {
        gotoxy(20,14);
        printf("Jumlah sweep , nsweep(>0) = ");
        scanf("%d",&nsweep);
    }
    printf("Besarx = ");scanf("%d",&Besarx);
    }
    void temperatur()
    {
        clrscr();

```

```

gotoxy(27,9);printf("Temp awal =");scanf("%lf",&Tawal);
gotoxy(27,11);printf("delta temp=");scanf("%lf",&delt);
gotoxy(27,13);printf("Jumlah variasi=");scanf("%d",&nvar);
}
void kisi()
{
for (i2=0;i2<=x;++i2)
{
for (j2=0;j2<=y;++j2)
{
for( k2=0;k2<=z;++k2)
{
nilai_random = random(32760);
switch(Besarx)
{
case 0:
if (nilai_random<32766)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/

```



```
/* Interaksi atom oksigen dengan tetangganya*/
```

```
F=S[i1][j1][k1]*(S[i1p][j1][k1]+S[i1m][j1][k1]+S[i1][j1p][k1]+S[i1][j1m][k1])  
+S[i1][j1][k1p]*(S[i1p][j1][k1p]+S[i1m][j1][k1p]+S[i1][j1p][k1p]+S[i1][j1m][k1p])  
+S[i1][j1][k1m]*(S[i1p][j1][k1m]+S[i1m][j1][k1m]+S[i1][j1p][k1m]+S[i1][j1m][k1m])  
+S[i1][j1][k11p]*(S[i1p][j1][k11p]+S[i1m][j1][k11p]+S[i1][j1p][k11p]+S[i1][j1m][k11p]);
```

```
G=S[i1][j1][k1]*(S[i1m][j1m][k1]+S[i1p][j1p][k1])  
+S[i1][j1][k1p]*(S[i1m][j1m][k1p]+S[i1p][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1m][j1m][k1m]+S[i1p][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1m][j1m][k11p]+S[i1p][j1p][k11p]);
```

```
P=S[i1][j1][k1]*(S[i1p][j1m][k1]+S[i1m][j1p][k1])  
+S[i1][j1][k1p]*(S[i1p][j1m][k1p]+S[i1m][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1p][j1m][k1m]+S[i1m][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1p][j1m][k11p]+S[i1m][j1p][k11p]);
```

```
GP=S[i1][j1][k1]*(S[i1m][j1m][k1]+S[i1p][j1p][k1])  
+S[i1][j1][k1p]*(S[i1m][j1m][k1p]+S[i1p][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1m][j1m][k1m]+S[i1p][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1m][j1m][k11p]+S[i1p][j1p][k11p])  
+S[i1][j1][k1]*(S[i1p][j1m][k1]+S[i1m][j1p][k1])  
+S[i1][j1][k1p]*(S[i1p][j1m][k1p]+S[i1m][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1p][j1m][k1m]+S[i1m][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1p][j1m][k11p]+S[i1m][j1p][k11p]);
```

```
sumF=sumF+F;  
sumG=sumG+G;  
sumP=sumP+P;  
sumGP=sumGP+GP;  
E1=v1*sumF-v2*sumG+v3*sumP-v4*sumGP-0.2765;  
}
```

```
void kisi_berikut()
```

```
{  
for (i5=0;i5<=x;++i5)  
{  
for (j5=0;j5<=y;++j5)  
{  
for( k5=0;k5<=z;++k5)  
{  
nilai_random = random(32760);  
switch(Besarx)  
{  
case 0:  
if (nilai_random<32766)  
S[i5][j5][k5]=1;  
else  
S[i5][j5][k5]=0;  
break;  
/* 90% posisi atom oksigen terisi(x=0.1)*/  
case 1:  
if(nilai_random<29489)  
S[i5][j5][k5]=1;  

```

```

else
S[i5][j5][k5]=0;
break;

/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/
case 6:
if(nilai_random<13106)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 30% posisi atom oksigen terisi (x=0.7)*/
case 7:
if(nilai_random<9829)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 20% posisi atom oksigen terisi (x=0.8)*/
case 8:
if(nilai_random<6553)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 10% posisi atom oksigen terisi (x=0.9)*/
case 9:
if(nilai_random<3277)
S[i5][j5][k5]=1;
else

```



```

        S[i5][j5][k5]=0;
        break;
    }
}
}
}
}
void energi_berikut()
{
    double F6=0,G6=0,P6=0,GP6=0;
    double sumF6=0,sumG6=0,sumP6=0,sumGP6=0;
    int i6m=0,i6p=0,
        j6m=0,j6p=0,
        k6m=0,k6p=0,
        k66m=0,k66p=0;

    int loop;
    for(i6=0;i6<=x;++i6)
    {
        if(i6<x) i6p=i6+1;else i6p=1;
        if(i6>1) i6m=i6-1;else i6m=x;
        for(j6=0;j6<=y;++j6)
        {
            if(j6<y) j6p=j6+1;else j6p=1;
            if(j6>1) j6m=j6-1;else j6m=y;
            for(k6=0;k6<=z;++k6)
            {
                if(k6<z) k6p=k6+1;else k6p=1;
                if(k6>1) k6m=k6-1;else k6m=z;
                for(k6=0;k6<=z;++k6)
                {
                    if(k6<z) k66p=k6+1;else k66p=1;
                    if(k6>1) k66m=k6-1;else k66m=z;

/* Interaksi atom oksigen dengan tetangganya*/

F6=S[i6][j6][k6]*(S[i6p][j6][k6]+S[i6m][j6][k6]+S[i6][j6p][k6]+S[i6][j6m][k6])
+S[i6][j6][k6p]*(S[i6p][j6][k6p]+S[i6m][j6][k6p]+S[i6][j6p][k6p]+S[i6][j6m][k6p])
+S[i6][j6][k6m]*(S[i6p][j6][k6m]+S[i6m][j6][k6m]+S[i6][j6p][k6m]+S[i6][j6m][k6m])
+S[i6][j6][k66p]*(S[i6p][j6][k66p]+S[i6m][j6][k66p]+S[i6][j6p][k66p]+S[i6][j6m][k66p]);

G6=S[i6][j6][k6]*(S[i6m][j6m][k6]+S[i6p][j6p][k6])
+S[i6][j6][k6p]*(S[i6m][j6m][k6p]+S[i6p][j6p][k6p])
+S[i6][j6][k6m]*(S[i6m][j6m][k6m]+S[i6p][j6p][k6m])
+S[i6][j6][k66p]*(S[i6m][j6m][k66p]+S[i6p][j6p][k66p]);

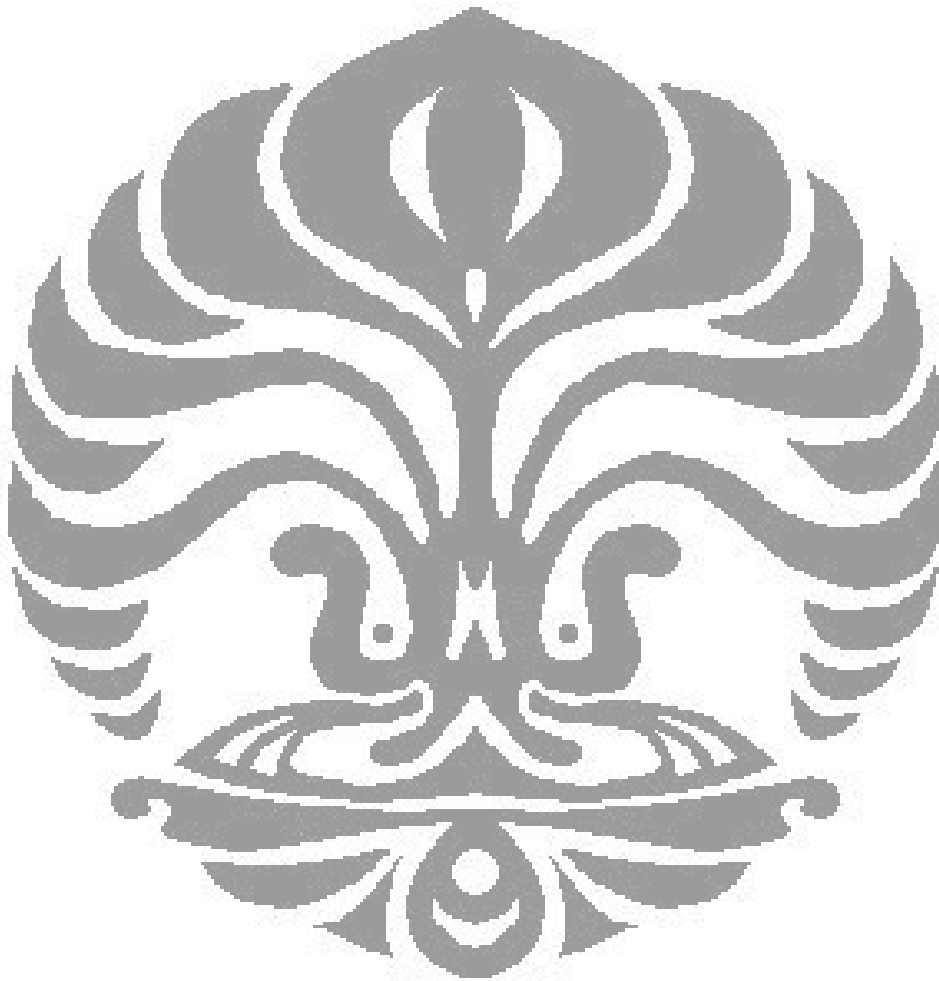
P6=S[i6][j6][k6]*(S[i6p][j6m][k6]+S[i6m][j6p][k6])
+S[i6][j6][k6p]*(S[i6p][j6m][k6p]+S[i6m][j6p][k6p])
+S[i6][j6][k6m]*(S[i6p][j6m][k6m]+S[i6m][j6p][k6m])
+S[i6][j6][k66p]*(S[i6p][j6m][k66p]+S[i6m][j6p][k66p]);

GP6=S[i6][j6][k6]*(S[i6m][j6m][k6]+S[i6p][j6p][k6])
+S[i6][j6][k6p]*(S[i6m][j6m][k6p]+S[i6p][j6p][k6p])
+S[i6][j6][k6m]*(S[i6m][j6m][k6m]+S[i6p][j6p][k6m])
+S[i6][j6][k66p]*(S[i6m][j6m][k66p]+S[i6p][j6p][k66p])
+S[i6][j6][k6]*(S[i6p][j6m][k6]+S[i6m][j6p][k6])
+S[i6][j6][k6p]*(S[i6p][j6m][k6p]+S[i6m][j6p][k6p])
+S[i6][j6][k6m]*(S[i6p][j6m][k6m]+S[i6m][j6p][k6m])

```



```
}
gotoxy(19,nvar+4);printf("\xC8");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,nvar+4);printf("\xCA");
gotoxy(35,nvar+4);printf("\xCA");
gotoxy(55,nvar+4);printf("\xCA");
gotoxy(72,nvar+4);printf("\xBC\r\n");
}
```



```

/*-----*/
/* PROGRAM MODEL ASYNNNI TIO monolayers 5 bidang CuO */
/* Saepul Qodar S2 Fisika UI */
/* OS : Microsoft Windows XP SP2 */
/* Compiler : Turbo C++ 4.5 */
/* Memory Model : Large */
/*-----*/
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <dos.h>
#include <conio.h>
#include <stdio.h>
int pil=0,
    x=0,
    w=0,
    y=0,
    z=0,
    i1=0,
    j1=0,
    k1=0,
    i2=0,
    j2=0,
    k2=0,
    i5=0,
    j5=0,
    k5=0,
    i6=0,
    j6=0,
    k6=0,
    Besarx=0;
long w1,wt;
int ngroup=0,
    nsweep=0;
double nspin=0;
double probabilitas=0;
double Etotal=0,Ekuadrat=0,
    rata_group_energi=0,
    rata_group_kuadrat=0;
int jumlahnol=0,jumlahsatu=0;
double groupe=0,groupe2=0;
double energi_total=0,
    E1=0,
    E1_rata_rata=0,
    E2=0,
    E2_rata_rata=0,
    delta_energi=0,
    E6=0,
    E8_rata_rata=0;
double rata_rata=0,
    rata_rata_kuadrat=0,
    kuadrat=0,
    kuadrat_rata_rata=0,
    rata_sweep_energi=0,
    rata_sweep_kuadrat=0,
    E_hitung=0,
    delta=0,

```

```

        absolut=0,
        pertemperatur=0,
        sum_sweep_kuadrat=0,
        sum_sweep_energi=0,
        sume=0,
        sume2=0;

int nvar=0;
double delt=0,Tawal=0,d=0;
double huge spesifik_heat[2400],E3[4000],T[25];
double kalor_jenis=0;
int jumlah_random=0,
    nilai_random=0;
int nilai_random2=0;
int sweepke=0,grupke=0;

long huge S[100L][100L][100L];/*jumlah posisi atom oksigen maksimum 64x64x3*/
double cv=0,
    cvc=0;
double v1=0,
    v2=0,
    v3=0,
    v4=0;
int hitung2=0,hitung1=0,n=0;
double kb=0,E=0;
char pil1='Y';
void tampilan(void),input_parameter(void),energi_pembanding(void),kisi(void);
void kisi_awal(void),energi_awal(void),kisi_berikut(void),energi_berikut(void);
int hitungE_M(void);
void hasil(void),temperatur(void),kalor(void),tabel(void);
extern int _directvideo=0;
int main(void)
{
    register int isweep,igroup,test,ivar;

    tampilan();
    input_parameter();
    temperatur();
    w1=time('\0');
    randomize();
    /*Perhitungan dengan temperatur yang berbeda-beda*/
    for (ivar=1;ivar<=nvar;++ivar)
    {
        T[1]=Tawal;
        T[ivar+1]=T[ivar]+delt;
        kalor_jenis=0;
        sume=0;
        sume2=0;
        /*Menentukan posisi awal atom dan menghitung energinya*/
        kisi();
        energi_awal();
        gotoxy(20,12);
        printf("Nx=%d Ny=%d Nz=%d \r\n",x,y,z);
        gotoxy(20,20);printf("wait...");
        sum_sweep_kuadrat=0;sum_sweep_energi=0,grupke=0;
        hitung2=0;hitung1=0;E_hitung=0;

        n=0;

```

```

for (igroup=1;igroup<=ngroup;++igroup)
{
groupe=0;groupe2=0;Etotal=0;
Ekuadrat=0;sweepke=0;
for (isweep=1;isweep<=nsweep;++isweep)
{
hitung2=hitung2+1;
hitung1=hitung2-1;
/*Menentukan posisi kedua,ketiga dst atom dan menghitung energinya*/
kisi_berikut();
energi_berikut();
E3[hitung2]=E6;
/*Mencari delta energi,E2-E1,E3-E2,E4-E3 dst*/
if (hitung2==1)
delta=E3[hitung2]-E1;
else
delta=E3[hitung2]-E3[hitung1];
kb=0.00008125;
if (delta <= 0)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
pertemperatur=delta/(2*T[ivar]*kb);
probabilitas=0.5*(1-tanh(pertemperatur));
nilai_random=random(32760);
d=nilai_random/(double)32760;
/*Membandingkan bilangan random dengan probabilitas
untuk menentukan apakah kondisi tersebut bisa diterima
sebagai kondisi berikutnya*/
if (d<probabilitas)
{
E_hitung=E3[hitung2];
n=0;
}
else
{
if (hitung2==1)
{
E3[hitung1+1]=E1;
E_hitung=0;
}
else
{
n=n+1;
E3[hitung1+1]=E3[hitung1-(n-1)];
E_hitung=0;
}
}
}
}
Etotal=Etotal+E_hitung;
Ekuadrat=Ekuadrat+(E_hitung*E_hitung);
}

rata_sweep_energi=Etotal/nsweep;
rata_sweep_kuadrat=Ekuadrat/nsweep;
sum_sweep_energi=sum_sweep_energi+rata_sweep_energi;

```

```

        sum_sweep_kuadrat=sum_sweep_kuadrat+rata_sweep_kuadrat;
    }
    rata_group_energi=sum_sweep_energi/ngroup;
    rata_rata_kuadrat=sum_sweep_kuadrat/ngroup;
    kuadrat_rata_rata=rata_group_energi*rata_group_energi;
/* c=<E^2>-<E>^2/k*T */
specific_heat[ivar]=(rata_rata_kuadrat-kuadrat_rata_rata)/(kb*T[ivar]*T[ivar]);
    }
    while (pil1=='Y' || pil1=='y')
    {

        tabel();
        printf("press any key ");getch();
        getch();

        printf("Ingin melihat tabel lagi<Y/T>?");
        pil1=getch();
    }
    return 0;
}
void tampilan()
{
    register int i,j;
    clrscr();
    gotoxy(30,2);printf("COMPUTATIONAL PHYSICS\r\n");

    gotoxy(18,5);printf("Departement of Physics,University of Indonesia\r\n");
    gotoxy(10,6);printf("\xC9");
    for (i=1;i<=59;++i)
        printf("\xCD");
        printf("\xBB\r\n");
        for (j=7;j<=20;++j)
        {
            gotoxy(10,j);printf("\xBA");
            gotoxy(70,j);printf("\xBA\r\n");
        }
        gotoxy(10,21);
        printf("\xCB");
        for (i=1;i<=59;++i)
            printf("\xCD");
            printf("\xBC\r\n");
            gotoxy(33,8);printf("Copyright 2006\r\n");
            gotoxy(30,10);printf("ASYNNNI MODEL\r\n");
            gotoxy(22,11);printf("Using The Metropolis Algorithm\r\n");
            gotoxy(36,14);printf("*****\r\n");
            gotoxy(22,17);printf("Type ctrl-break to stop at a promt.\r\n");
            gotoxy(28,19);
            printf("Press any key to comtinue...");
            getch();
    }
void input_parameter()
{
    clrscr();
    gotoxy(20,10);
    printf("Arah_x,Nx (<=95) = ");
    scanf("%d",&x);
    while(x<=0 || x>95)
    {

```

```

        gotoxy(20,10);
        printf("Arah_x, Nx (<=95) = ");
        scanf("%d",&x);
    }
    gotoxy(20,12);
    printf("Arah_y , Ny (<=95) = ");
    scanf("%d",&y);
    while (y<=0 || y>95)
    {
        gotoxy(20,12);
        printf("Arah_y, Ny (<=95) = ");
        scanf("%d",&y);
    }
    gotoxy(20,14);
    printf("Arah_z , Nz (<=95) = ");
    scanf("%d",&z);
    while (z<=0 || z>95)
    {
        gotoxy(29,14);
        printf("Arah_z , Nz (<=95) = ");
        scanf("%d",&z);
    }
    gotoxy(20,16);
    printf("v1= ");
    scanf("%lf",&v1);
    gotoxy(20,18);
    printf("v2= ");
    scanf("%lf",&v2);
    gotoxy(20,20);
    printf("v3= ");
    scanf("%lf",&v3);
    gotoxy(20,22);
    printf("v4= ");
    scanf("%lf",&v4);
    clrscr();
    gotoxy(20,12);
    printf("Jumlah group, ngroup (>0) = ");
    scanf("%d",&ngroup);
    while(ngroup<=0)
    {
        gotoxy(20,12);
        printf("Jumlah group, ngroup (>0) = ");
        scanf("%d",&ngroup);
    }
    gotoxy(20,14);
    printf("Jumlah sweep , nsweep(>0) = ");
    scanf("%d",&nsweep);
    while(nsweep<=0)
    {
        gotoxy(20,14);
        printf("Jumlah sweep , nsweep(>0) = ");
        scanf("%d",&nsweep);
    }
    printf("Besarx = ");scanf("%d",&Besarx);
    }
    void temperatur()
    {
        clrscr();

```



```

gotoxy(27,9);printf("Temp awal =");scanf("%lf",&Tawal);
gotoxy(27,11);printf("delta temp=");scanf("%lf",&delt);
gotoxy(27,13);printf("Jumlah variasi=");scanf("%d",&nvar);
}
void kisi()
{
for (i2=0;i2<=x;++i2)
{
for (j2=0;j2<=y;++j2)
{
for( k2=0;k2<=z;++k2)
{
nilai_random = random(32760);
switch(Besarx)
{
case 0:
if (nilai_random<32766)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i2][j2][k2]=1;
else
S[i2][j2][k2]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/

```



```
/* Interaksi atom oksigen dengan tetangganya*/
```

```
F=S[i1][j1][k1]*(S[i1p][j1][k1]+S[i1m][j1][k1]+S[i1][j1p][k1]+S[i1][j1m][k1])  
+S[i1][j1][k1p]*(S[i1p][j1][k1p]+S[i1m][j1][k1p]+S[i1][j1p][k1p]+S[i1][j1m][k1p])  
+S[i1][j1][k1m]*(S[i1p][j1][k1m]+S[i1m][j1][k1m]+S[i1][j1p][k1m]+S[i1][j1m][k1m])  
+S[i1][j1][k11p]*(S[i1p][j1][k11p]+S[i1m][j1][k11p]+S[i1][j1p][k11p]+S[i1][j1m][k11p])  
+S[i1][j1][k11m]*(S[i1p][j1][k11m]+S[i1m][j1][k11m]+S[i1][j1p][k11m]+S[i1][j1m][k11m]);
```

```
G=S[i1][j1][k1]*(S[i1m][j1m][k1]+S[i1p][j1p][k1])  
+S[i1][j1][k1p]*(S[i1m][j1m][k1p]+S[i1p][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1m][j1m][k1m]+S[i1p][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1m][j1m][k11p]+S[i1p][j1p][k11p])  
+S[i1][j1][k11m]*(S[i1m][j1m][k11m]+S[i1p][j1p][k11m]);
```

```
P=S[i1][j1][k1]*(S[i1p][j1m][k1]+S[i1m][j1p][k1])  
+S[i1][j1][k1p]*(S[i1p][j1m][k1p]+S[i1m][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1p][j1m][k1m]+S[i1m][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1p][j1m][k11p]+S[i1m][j1p][k11p])  
+S[i1][j1][k11m]*(S[i1p][j1m][k11m]+S[i1m][j1p][k11m]);
```

```
GP=S[i1][j1][k1]*(S[i1m][j1m][k1]+S[i1p][j1p][k1])  
+S[i1][j1][k1p]*(S[i1m][j1m][k1p]+S[i1p][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1m][j1m][k1m]+S[i1p][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1m][j1m][k11p]+S[i1p][j1p][k11p])  
+S[i1][j1][k11m]*(S[i1m][j1m][k11m]+S[i1p][j1p][k11m])  
+S[i1][j1][k1]*(S[i1p][j1m][k1]+S[i1m][j1p][k1])  
+S[i1][j1][k1p]*(S[i1p][j1m][k1p]+S[i1m][j1p][k1p])  
+S[i1][j1][k1m]*(S[i1p][j1m][k1m]+S[i1m][j1p][k1m])  
+S[i1][j1][k11p]*(S[i1p][j1m][k11p]+S[i1m][j1p][k11p])  
+S[i1][j1][k11m]*(S[i1p][j1m][k11m]+S[i1m][j1p][k11m]);
```

```
sumF=sumF+F;  
sumG=sumG+G;  
sumP=sumP+P;  
sumGP=sumGP+GP;  
E1=v1*sumF-v2*sumG+v3*sumP-v4*sumGP+3.3725;  
}
```

```
}  
}  
}  
void kisi_berikut()  
{  
for (i5=0;i5<=x;++i5)  
{  
for (j5=0;j5<=y;++j5)  
{  
for( k5=0;k5<=z;++k5)  
{  
nilai_random = random(32760);  
switch(Besarx)  
{  
case 0:  
if (nilai_random<32766)  
S[i5][j5][k5]=1;  
else  
S[i5][j5][k5]=0;
```

```

break;
/* 90% posisi atom oksigen terisi(x=0.1)*/
case 1:
if(nilai_random<29489)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;

/* 80% posisi atom oksigen terisi (x=0.2) */
case 2:
if(nilai_random<26213)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 70% posisi atom oksigen terisi (x=0.3)*/
case 3:
if(nilai_random<22936)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 60% posisi atom oksigen terisi (x=0.4)*/
case 4:
if(nilai_random<22113)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 50% posisi atom oksigen terisi (x=0.5)*/
case 5:
if(nilai_random<21294)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 40% posisi atom oksigen terisi (x=0.6)*/
case 6:
if(nilai_random<13106)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 30% posisi atom oksigen terisi (x=0.7)*/
case 7:
if(nilai_random<9829)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
/* 20% posisi atom oksigen terisi (x=0.8)*/
case 8:
if(nilai_random<6553)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;

```

```

/* 10% posisi atom oksigen terisi (x=0.9)*/
case 9:
if(nilai_random<3277)
S[i5][j5][k5]=1;
else
S[i5][j5][k5]=0;
break;
}
}
}
}
}
void energi_berikut()
{
double F6=0,G6=0,P6=0,GP6=0;
double sumF6=0,sumG6=0,sumP6=0,sumGP6=0;
int i6m=0,i6p=0,
j6m=0,j6p=0,
k6m=0,k6p=0,
k66m=0,k66p=0;
int loop;
for(i6=0;i6<=x;++i6)
{
if(i6<x) i6p=i6+1;else i6p=1;
if(i6>1) i6m=i6-1;else i6m=x;
for(j6=0;j6<=y;++j6)
{
if(j6<y) j6p=j6+1;else j6p=1;
if(j6>1) j6m=j6-1;else j6m=y;
for(k6=0;k6<=z;++k6)
{
if(k6<z) k6p=k6+1;else k6p=1;
if(k6>1) k6m=k6-1;else k6m=z;
for(k6=0;k6<=z;++k6)
{
if(k6<z) k66p=k6+1;else k66p=1;
if(k6>1) k66m=k6-1;else k66m=z;
}
}
}
}
}
/* Interaksi atom oksigen dengan tetangganya*/
F6=S[i6][j6][k6]*(S[i6p][j6][k6]+S[i6m][j6][k6]+S[i6][j6p][k6]+S[i6][j6m][k6])
+S[i6][j6][k6p]*(S[i6p][j6][k6p]+S[i6m][j6][k6p]+S[i6][j6p][k6p]+S[i6][j6m][k6p])
+S[i6][j6][k6m]*(S[i6p][j6][k6m]+S[i6m][j6][k6m]+S[i6][j6p][k6m]+S[i6][j6m][k6m])
+S[i6][j6][k66p]*(S[i6p][j6][k66p]+S[i6m][j6][k66p]+S[i6][j6p][k66p]+S[i6][j6m][k66p])
+S[i6][j6][k66m]*(S[i6p][j6][k66m]+S[i6m][j6][k66m]+S[i6][j6p][k66m]+S[i6][j6m][k66m]);
G6=S[i6][j6][k6]*(S[i6m][j6m][k6]+S[i6p][j6p][k6])
+S[i6][j6][k6p]*(S[i6m][j6m][k6p]+S[i6p][j6p][k6p])
+S[i6][j6][k6m]*(S[i6m][j6m][k6m]+S[i6p][j6p][k6m])
+S[i6][j6][k66p]*(S[i6m][j6m][k66p]+S[i6p][j6p][k66p])
+S[i6][j6][k66m]*(S[i6m][j6m][k66m]+S[i6p][j6p][k66m]);
P6=S[i6][j6][k6]*(S[i6p][j6m][k6]+S[i6m][j6p][k6])
+S[i6][j6][k6p]*(S[i6p][j6m][k6p]+S[i6m][j6p][k6p])
+S[i6][j6][k6m]*(S[i6p][j6m][k6m]+S[i6m][j6p][k6m])
+S[i6][j6][k66p]*(S[i6p][j6m][k66p]+S[i6m][j6p][k66p])

```

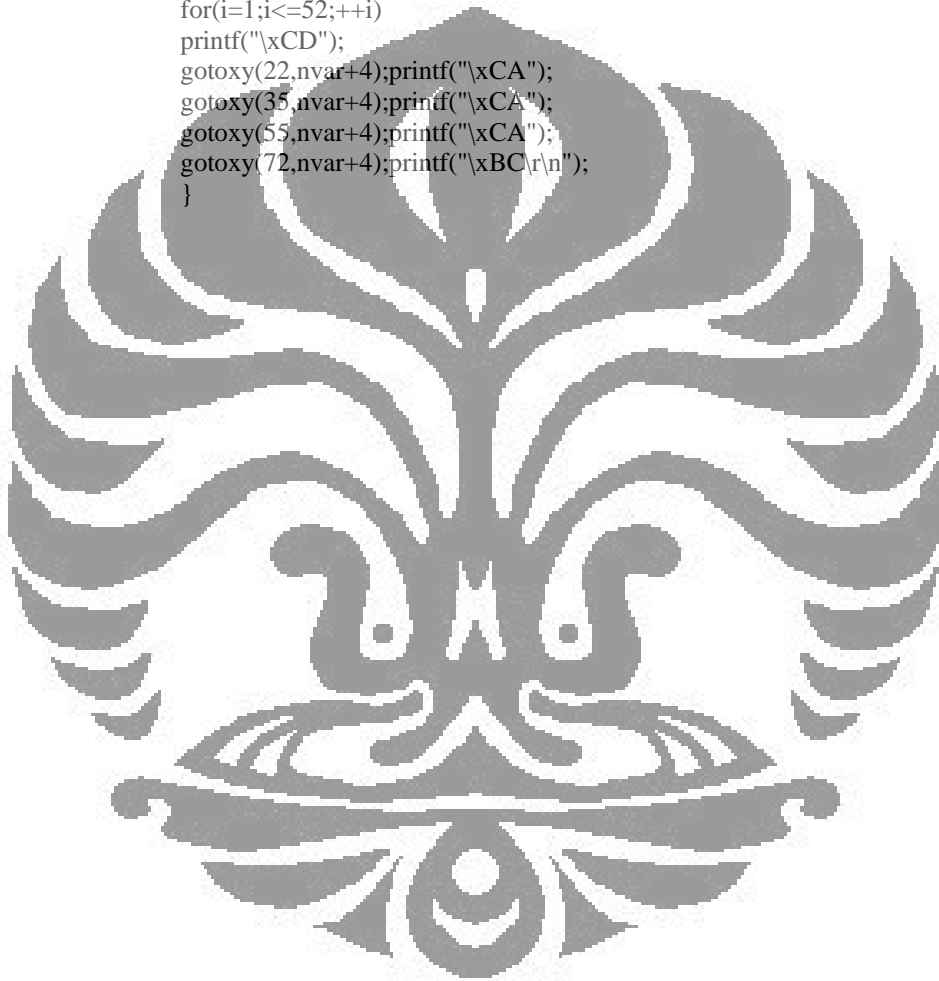


```

for(i=4;i<=(nvar+3);++i)
{
gotoxy(19,i);printf("\xBA");printf("%d",i-3);
gotoxy(22,i);printf("\xBA");
if(i>4)

gotoxy(23,(nvar+7-i));printf("%+1.41f",T[nvar+4-i]);
gotoxy(35,(nvar+7-i));printf("\xBA");
gotoxy(40,(nvar+7-i));printf("%+1.6f",specific_heat[i-4]);
gotoxy(55,(nvar+7-i));printf("\xBA");
gotoxy(72,i);printf("\xBA");
}
gotoxy(19,nvar+4);printf("\xC8");
for(i=1;i<=52;++i)
printf("\xCD");
gotoxy(22,nvar+4);printf("\xCA");
gotoxy(35,nvar+4);printf("\xCA");
gotoxy(55,nvar+4);printf("\xCA");
gotoxy(72,nvar+4);printf("\xBC\r\n");
}

```



```

/*-----*/
/* PROGRAM MODEL ASYNNNI TI 1201 */
/* Saepul Qodar S2 Fisika UI */
/* OS : Microsoft Windows XP SP2 */
/* Compiler : Turbo C++ 4.5 */
/* Memory Model : Large */
/*-----*/
/* Programnya sama seperti pada TI 2201 hanya berbeda pada nilai masukannya */

/*-----*/
/* PROGRAM MODEL ASYNNNI TI 1212 */
/* Saepul Qodar S2 Fisika UI */
/* OS : Microsoft Windows XP SP2 */
/* Compiler : Turbo C++ 4.5 */
/* Memory Model : Large */
/*-----*/
/* Programnya sama seperti pada TI 2212 hanya berbeda pada nilai masukkan dan energi interaksinya */

/*-----*/
/* PROGRAM MODEL ASYNNNI TI 1223 */
/*-----*/
/* Programnya sama seperti pada TI 2223 hanya berbeda pada nilai masukkan dan energi interaksinya */

/*-----*/
/* PROGRAM MODEL ASYNNNI TI 1234 */
/*-----*/
/* Programnya sama seperti pada TIO bilayers 4 bidang CuO hanya berbeda pada nilai masukkan dan energi interaksinya */

```