

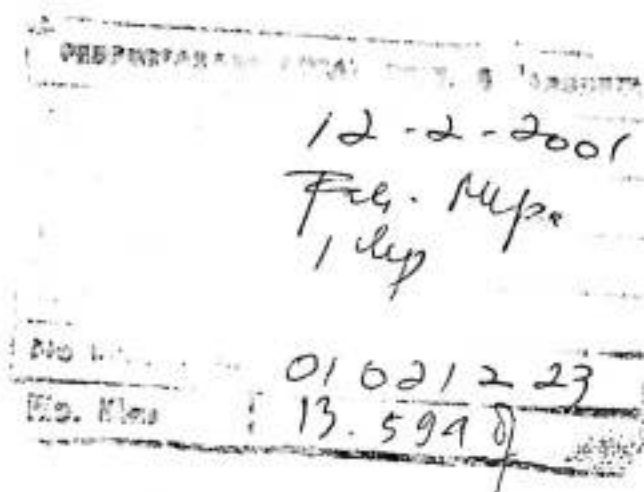
**RANCANG BANGUN PERANGKAT LUNAK  
UNTUK MENENTUKAN POLA INTENSITAS DIFRAKSI SERBUK  
DENGAN FUNGSI PROFIL PSEUDO-VOIGHT**



**OLEH**

**SYAIFUL ALAM**

**92 03 873**



**PROGRAM STUDI FISIKA  
JURUSAN FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2000**

**RANCANG BANGUN PERANGKAT LUNAK  
UNTUK MENENTUKAN POLA INTENSITAS DIFRAKSI SERBUK  
DENGAN FUNGSI PROFIL PSEUDO-VOIGHT**

**OLEH**

**SYAIFUL ALAM  
92 03 178**

**SKRIPSI**

*Untuk melengkapi tugas-tugas dan  
memenuhi syarat memperoleh  
Gelara Sarjana Fisika*

**PROGRAM STUDI FISIKA  
JURUSAN FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2000**

**RANCANG BANGUN PERANGKAT LUNAK  
UNTUK MENENTUKAN POLA INTENSITAS DIFRAKSI SERBUK  
DENGAN FUNGSI PROFIL PSEUDO-VOIGHT**

**OLEH**

**SYAIFUL ALAM  
92 03 178**

**Disetujui oleh  
Pembimbing Utama,**



**(Dr. Agus Purwanto)**

**NIP. 330 003 591**

**Pembimbing Pertama,**



**(Drs. Safaruddin A. Prasad, M.Si.)**

**NIP. 131 864 127**

**Pembimbing Kedua,**



**(Drs. Bambang Sugeng, M.Si.)**

**NIP. 330 003 154**

**Pada tanggal, Agustus 2000**

# SKRIPSI

OLEH

**SYAIFUL ALAM**

92 03 178



**PROGRAM STUDI FISIKA**

**JURUSAN FISIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2000**

## **Mutiara Hikmah :**

“Apabila Anda membaca Al-Qur’an, maknanya akan jelas dihadapan Anda.

Tetapi bila Anda membacanya sekali lagi,

akan Anda temukan pula makna lain yang berbeda dengan makna-makna sebelumnya.

Demikian seterusnya, sampai-sampai Anda (dapat) menemukan kalimat atau kata yang mempunyai arti bermacam-macam, semuanya benar atau mungkin benar.

(Ayat-ayat Al-Qur’an) bagaikan Intan

setiap sudutnya memancarkan cahaya yang berbeda-beda

dengan apa yang terpancar dari sudut-sudut lain

Dan tidak mustahil, jika Anda mempersilakan orang lain memandangnya,

maka Ia akan melihat lebih banyak ketimbang apa yang Anda lihat”.

**Abdullah Darraz**

(dalam Al-Na-ba’ Al-Azhim)

# FISIKA



# MAKASSAR

# UCAPAN TERIMA KASIH

## Bismillahirrahmaanirrahim

### Assalamu 'Alaikum Warahmatullah Wabarakatuh

Syukur Alhamdulillah Rabbilalamin, ketika tulisan ini terwujud sebagaimana adanya, maka rangkaian perjalanan penulis dalam menyelesaikan program studi S-1 pada Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin, sudah mencapai titik akhir.

Sebagai bagian dari suatu perjalanan yang masih sangat panjang, penulis menyadari masih banyak kekurangan dan kelemahan didalamnya. Namun demikian, penulis tetap optimis untuk menggunakan kesempatan-kesempatan selanjutnya dalam berbuat yang lebih baik. Adapun judul tulisan ini adalah "*Rancang Bangun Perangkat Lunak Untuk Menentukan Intensitas Pola Difraksi Silikon Dengan Fungsi Profil Pseudo-Voight*" merupakan hasil penelitian yang penulis lakukan melalui suatu proses pembuatan software (program) dengan menggunakan compiler bahasa pemrograman C++, yang dalam hal ini **BORLAND C++ versi 5,0**.

Dalam rangka perampungan tulisan ini, penulis telah mendapat banyak sumbangsih baik berupa spirit maupun materi yang amat sangat berharga, baik secara langsung maupun tidak langsung.

- 1) Kepada **Ibunda Quratul Aeni** dan **Ayahanda Ahmad Elsyah**, sembah sujud penulis atas limpahan kasih sayang serta siraman iman yang diberikan sejak penulis masih dalam kandungan hingga penulis menyelesaikan studi.
- 2) Para Pembimbing penulis, untuk itu penulis ingin menghaturkan rasa terima kasih kepada Bapak ;

- **DR. Agus Purwanto** selaku pembimbing utama, yang telah memberikan bimbingan dan menyediakan fasilitas-fasilitas serta mengarahkan penulis dalam membuat dan menyelesaikan tulisan ini.
- **Drs. Safaruddin A. Prasad, M. Si.**, selaku pembimbing pertama, yang telah memberikan saran-saran dan motivasi penyelesaian tulisan ini.
- **Drs. Bambang Sugeng, M. Si.**, selaku pembimbing kedua yang telah memperkaya penulis dalam Metodologi Penelitian.

- 3) Pada kesempatan ini pula, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada **Bapak Ketua Jurusan Fisika** Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.
- 4) Selanjutnya **Bapak Drs. Altin Massinai, MT** dan **Drs. Sakka, M. Si.**, yang banyak memotivasi serta mengingatkan penulis dalam tugas akhir ini.
- 5) **Bapak Drs. Syamsu Arief, M. Si.** yang juga banyak membantu dalam penulisan metodologi pembuatan program.
- 6) Ibu **DR. Sri Suryani, D.E.A.**, yang telah membantu penulis dalam menemukan metodologi penulisan skripsi yang tepat.
- 7) **Bapak Drs. Tasrief Surungan, M. Sc.**, yang telah mengoreksi beberapa kesalahan penulisan Abstract.
- 8) **Seluruh Staf Pengajar dan Staf Jurusan** yang telah memberi arti dan kesan khusus kepada penulis selama menuntut ilmu pada Jurusan Fisika.
- 9) **Bapak Kepala PPSM serta seluruh staf** yang telah memberi kesan pada suasana kerja yang ramah dan harmonis selama penulis melakukan penelitian di BATAN.
- 10) Selain itu, penulis menyampaikan salam hormat dan terima kasih kepada teman-teman se-angkatan 92, antara lain **Jumaris, S. Si., Wasir, S. Si., Munawar Ch., S. Si., Rosliana Eso, S. Si.**, dan teman seperjuangan **Baina Afkril, S. Si.** serta **Abd. Rahim Syam, S. Si.**, dan masih banyak lainnya; **Bang Wahyu, S. Si.**, dengan satu katanya konsentrasi, **Bang Ali, S. Si.**, dengan katanya saya juga pasti bisa, **Adik Lina, S. Si.**, juga atas print-out lembaran bebas lab. dan pustaka.
- 11) Kepada Saudara-saudaraku umumnya di Ikatan Mahasiswa Muhammadiyah (IMM), dan terkhusus Pengurus **Dewan Pimpinan Daerah (DPD) IMM Sulawesi Selatan periode 1999-2001**, penulis menyampaikan rasa terima kasih atas pengertian dan dukungannya dalam kebersamaan selama ini; **Bang Ibe'** atas



bantuannya sehingga penulis dapat mengolah data penelitian, Bang Asis atas Mousenya.

Sesuatu hal yang mustahil, jika semua yang memberi kontribusi kepada penulis dalam penyelesaian skripsi ini dapat dituliskan namanya di lembaran ini, oleh karena itu penulis menyampaikan ucapan : **Jazakumullahu Khairan Katsiiran.**

Disadari sepenuhnya bahwa kandungan tulisan ini masih banyak terdapat kekeliruan dan belum memberikan sumbangan yang berarti, namun penulis tetap berharap semoga apa yang dikerjakan selama ini sedikitnya dapat memberikan kontribusi terhadap kemajuan IPTEK yang dapat dimanfaatkan untuk kesejahteraan serta dapat bernilai ibadah buat penulis.

Fastabiqul Khaerat, semoga tulisan ini bermanfaat adanya.

Makassar, September 2000

Penulis,

**SYAIFUL ALAM**

# DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN .....	iv
UCAPAN TERIMA KASIH .....	v
DAFTAR ISI .....	viii
SARI BACAAN.....	x
ABSTRACT .....	xi
<b>BAB 1 PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Ruang Lingkup Penelitian .....	2
1.3 Tujuan Penelitian .....	2
1.4 Waktu dan Tempat Penelitian .....	2
<b>BAB 2 TINJAUAN PUSTAKA</b>	
2.1 Kristalografi dan Difraksi .....	3
2.1.1 Struktur Kristal .....	3
2.1.2 Bidang dan Arah Kristalografi .....	4
2.1.3 Difraksi Bragg .....	5
2.1.4 Hubungan antara Parameter Kisi dan Jarak antar Bidang Kristal .....	7
2.1.5 Hamburan oleh Unit Sel .....	7
2.2 Fungsi Profil Pseudo-Voight .....	11
<b>BAB 3 METODE PENELITIAN</b> .....	14

## **BAB 4 HASIL DAN PEMBAHASAN**

4.1 Analisis Keperluan.....	15
4.2 Diagram Alir (Flow Chart) Pemrograman.....	16
4.3 Desain Program .....	18
4.3.1 Deklarasi .....	18
4.3.2 Definisi Fungsi .....	22
4.3.3 Klasifikasi .....	23
4.3.4 Implementasi Program Penentuan Pola Intensitas.....	24
4.3.5 Hasil Pola Intensitas Difraksi .....	24
4.3.6 Analisis Hasil Perancangan Pola Intensitas .....	25

## **BAB 5 SIMPULAN DAN SARAN**

5.1 SIMPULAN .....	26
5.2 SARAN .....	27

<b>DAFTAR PUSTAKA .....</b>	<b>28</b>
-----------------------------	-----------

<b>LAMPIRAN-LAMPIRAN .....</b>	<b>29-42</b>
--------------------------------	--------------

## SARI BACAAN

Telah dibuat sebuah perangkat lunak dengan menggunakan program **BORLAND C++** untuk analisis hasil eksperimen difraksi neutron serbuk. Dengan menggunakan paket aplikasi Microsoft Excel, data input yang terdiri dari sudut difraksi Bragg dan intensitas sinar akan menghasilkan output berupa pola intensitas eksperimen. Data eksperimen yang terdiri dari posisi atom ( $x,y,z$ ), bidang penghambur ( $hkl$ ), parameter kisi ( $a$ ) dan panjang gelombang ( $\lambda$ ) sinar dijadikan input bagi perangkat lunak yang dibuat, sehingga menghasilkan pola intensitas perhitungan. Pola ini dicocokkan dengan pola intensitas eksperimen.

## ABSTRACT

A software of analysing the powder neutron diffraction has been made in **BORLAND C++** Program. By using Excel program, the data input which consists of Bragg angle diffraction and intensity result in pattern of experimental intensity. The experimental data such as atomic position (x,y,z), scattering plane (hkl), lattice parameter (a) and wavelength ( $\lambda$ ) are the input of the software is obtained. This has by which the pattern of calculated intensity a good agreement with the pattern of experimental intensity.

# FISIKA



# MAKASSAR

## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang

Operasi difraktometer neutron serbuk dengan sudut detektor terhadap sampel akan mengakibatkan puncak difraksi tergeser dan mengalami perluasan sehingga puncak tidak simetri.

Ada beberapa macam fungsi yang umum digunakan dalam metode perbaikan profil. Fungsi-fungsi tersebut antara lain : fungsi Time of Flight (TOF), fungsi Constant Wavelength (CW) dan fungsi Profil Pseudo-Voight. Pada penelitian ini fungsi yang digunakan adalah fungsi Profil Pseudo-Voight, fungsi ini memperbaiki profil sangat baik; fungsi Profil Pseudo-Voight merupakan pendekatan dari fungsi Gaussian dan Lorentzian.

Perangkat lunak untuk analisa struktur kristal berdasarkan data difraksi sinar-x dan neutron telah banyak tersedia. Namun demikian, perangkat lunak untuk analisa struktur kristal yang inkomensurat<sup>?</sup> masih belum banyak tersedia baik secara komersil maupun gratis. Oleh karena itu akan dibuat perangkat lunak dengan menggunakan bahasa pemrograman C++ yang memanfaatkan fasilitas objek (object-oriented programming, disingkat OOP) sehingga memudahkan pengembangan selanjutnya.

## **1.2 Ruang Lingkup Penelitian**

Penelitian yang akan dilakukan meliputi :

1. Data observasi yang digunakan hanya pada silikon (si) standar.
2. Intensitas terhitung yang akan dilakukan pada data observasi melibatkan fungsi Profil Pseudo-Voight, faktor skala dan intensitas background.

## **1.3. Tujuan Penelitian**

1. Membuat sebuah perangkat lunak untuk menghitung intensitas difraksi polikristal silikon standar dengan Borland C++.
2. Menentukan pola intensitas difraksi polikristal silikon standar.

## **1.4. Waktu dan Tempat Penelitian**

Waktu yang dipergunakan untuk penelitian sejak tanggal 18 Pebruari 1997 sampai dengan 28 Agustus 1998, bertempat di Pusat Penelitian Sains Materi (PPSM), Kawasan Pusat Penelitian Ilmu Pengetahuan dan Teknologi (Puspiptek), Badan Tenaga Atom Nasional (BATAN), Serpong, Tangerang.



# FISIKA



# MAKASSAR

## BAB 2

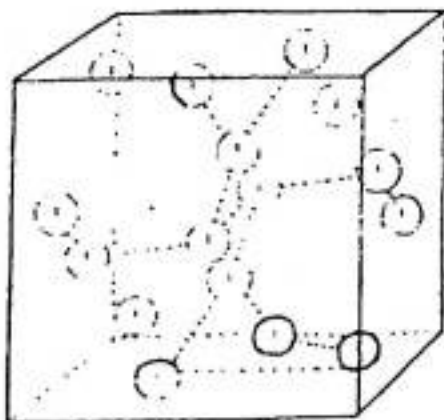
### TINJAUAN PUSTAKA

#### 2.1 Kristalografi dan Difraksi

##### 2.1.1 Struktur Kristal

Kristal adalah susunan atom yang teratur. Dalam material padat kristalin, atom-atom disusun secara berulang dengan jumlah yang tak berhingga oleh unit sel dalam ruang tiga dimensi. Struktur kristal dapat digambarkan dalam bagian-bagian kisi. Kelompok atom itu disebut dengan basis, pada saat basis mengalami pengulangan akan membentuk kristal. Terdapat tujuh sistem kristal dan empat belas kisi bravais yang dimiliki oleh material kristal seperti diperlihatkan pada Tabel 2.1.

Struktur kristal yang dimiliki oleh silikon (si) adalah struktur kubus pemusatan ruang (kpr). Bentuk kristal tersebut ditunjukkan pada Gambar 2.1.



Gambar 2.1 struktur Kristal pemusatan ruang (kpr).

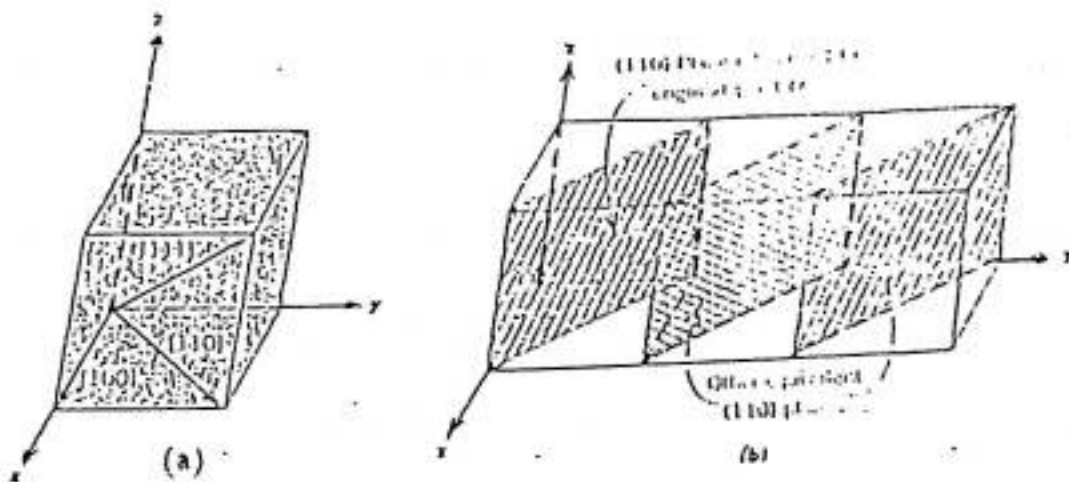
Tabel 2.1 Sistem kristal dan kisi bravais

Sistem Kristal	Panjang dan Sudut antar Sumbu	Kisi Bravais	Simbol Kristal
Kubus	$a=b=c$ $\alpha=\beta=\gamma=90^\circ$	sederhana pemusatan ruang pemusatan sisi	P I F
Tetragonal	$a=b\neq c$ $\alpha=\beta=\gamma=90^\circ$	sederhana pemusatan ruang	P I
Ortorombik	$a\neq b\neq c$ $\alpha=\beta=\gamma=90^\circ$	sederhana pemusatan ruang pemusatan dasar pemusatan sisi	P I C F
Rombohedral	$a=b=c$ $\alpha=\beta=\gamma\neq 90^\circ$	sederhana	R
Heksagonal	$a=b\neq c$ $\alpha=\beta=90^\circ, \gamma=120^\circ$	sederhana	P
Monoklinik	$a\neq b\neq c$ $\alpha=\gamma=90^\circ\neq\beta$	sederhana pemusatan dasar	P C
Triklirik	$a\neq b\neq c$ $\alpha\neq\beta\neq\gamma\neq 90^\circ$	sederhana	P

### 2.1.2 Bidang dan Arah Kristalografi

Dalam pembahasan material kristalin, perlu diketahui letak bidang-bidang dan arah kristalografi yang ada pada struktur kristal tersebut. Bidang kristalografi ditandai dengan aturan tertentu yang biasa disebut dengan indeks Miller (hkl) dan arah kristalografi ditandai dengan kurung siku [hkl].

Arah kristalografi adalah garis antara dua titik yang berupa sebuah vektor ditandai dengan panjang proyeksi garis tersebut dengan sumbu kristalografi (x-y-z). sedangkan bidang kristalografi ditandai dengan panjang garis dari titik pusat koordinat sumbu kristalografi ke titik hasil perpotongan bidang tersebut pada masing-masing sumbu. Bidang paralel memiliki indeks Miller yang sama, seperti ditunjukkan pada Gambar 2.2.

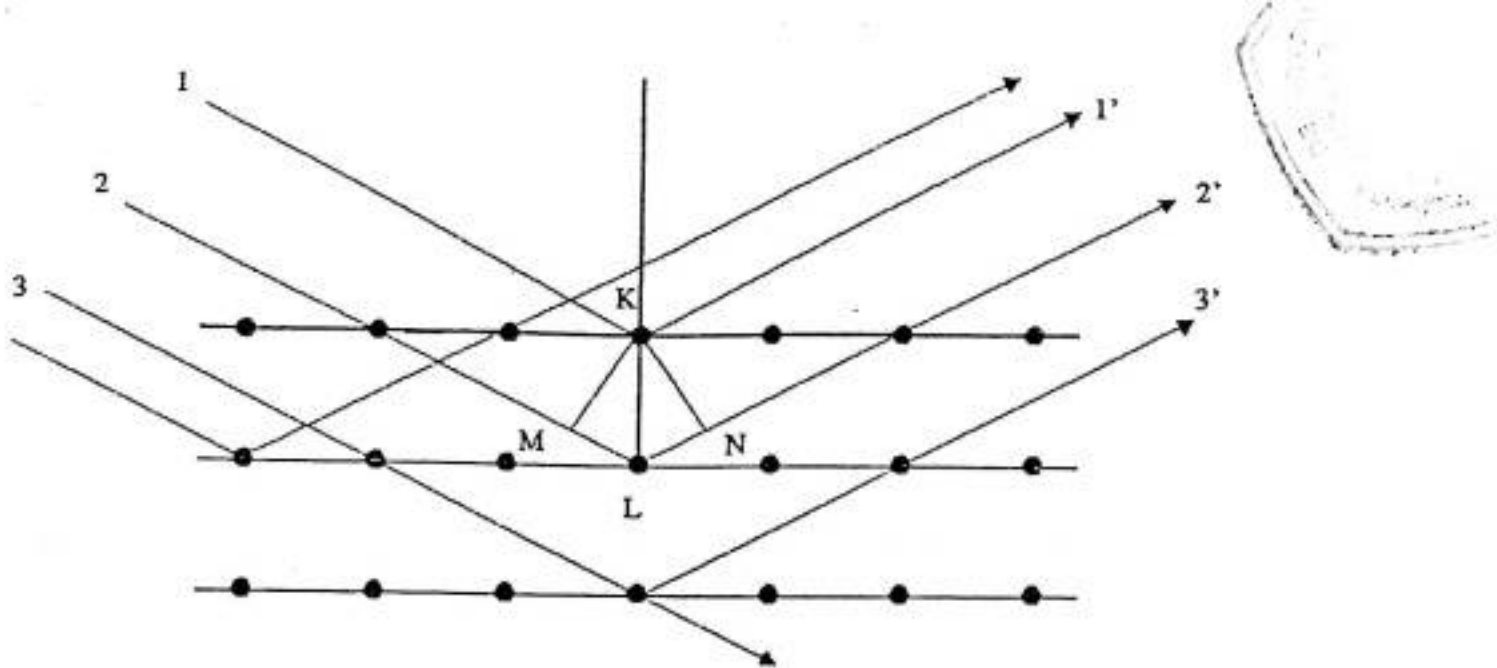


Gambar 2.2 Bidang dan Arah Kristalografi

### 2.1.3 Difraksi Bragg

Sinar-x akan terdifraksi bila mengenai atom-atom dalam material. Pada material non kristalin, difraksi terpecah ke segala arah, sementara material kristalin akan memancarkan sinar-x pada arah tertentu sesuai dengan bidang-bidang kristal.

Peristiwa yang terjadi saat sinar-x mengenai atom-atom pada bidang kristal ditunjukkan pada Gambar 2.3.



Gambar 2.3 Difraksi sinar-x oleh atom-atom pada bidang kristal

Andaikan sinar-x dengan panjang gelombang ( $\lambda$ ) menuju kristal membentuk sudut (diukur antara pancaran datang dengan bidang kristal). Semua atom-atom pada bidang pertama akan mendifraksikan sinar datang pada arah yang sejajar dengan sinar 1' dengan fasa yang sama. Peristiwa yang sama juga terjadi untuk atom-atom pada bidang lain.

Sinar 1 dan 2 akan terdifraksi oleh atom-atom K dan L dengan beda lintasan :

$$ML + LN = 2d' \sin \theta \quad (2.1)$$

sinar terdifraksi (1' dan 2') akan sefasa bila beda lintasan sebanding dengan n jumlah panjang gelombang, atau :

$$n\lambda = 2d' \sin \theta \quad (2.2)$$

persamaan 2.4 dikenal sebagai hukum Bragg, yang dapat ditulis sebagai berikut :

$$\lambda = 2d_{hkl} \sin \theta \quad (2.3)$$

dengan  $d_{hkl} = \frac{d'}{n}$  merupakan jarak antar bidang sebagai refleksi orde pertama dari bidang-bidang.

#### 2.1.4 Hubungan antara Parameter Kisi dengan Jarak antar Bidang Kristal

Suatu kristal memiliki banyak bidang hkl dan dipisahkan oleh jarak tertentu yang dilambangkan dengan  $d_{hkl}$ . Jarak ( $d_{hkl}$ ) antar bidang yang berdekatan dan paralel merupakan fungsi dari indeks Miller dan parameter kisi yang bergantung pada struktur kristalnya.

Hubungan antara jarak antar bidang dan parameter kisi untuk sistem kristal kubus adalah :

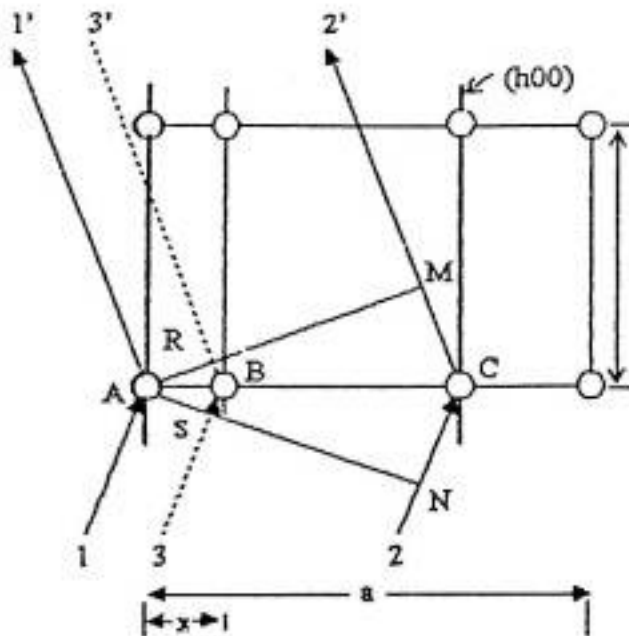
$$\text{kubus : } \frac{1}{(d_{hkl})^2} = \frac{h^2 + k^2 + l^2}{a^2} \quad (2.4)$$

Hubungan antara jarak antar bidang kristal dan parameter kisi untuk kristal lainnya ditampilkan pada lampiran II.

#### 2.1.5 Hamburan oleh Unit Sel

Sebagai asumsi bahwa hukum Bragg dipatuhi dalam peristiwa hamburan oleh unit sel, maka intensitas terdifraksi dapat ditentukan dari kristal. Kristal cukup dipandang sebagai unit sel. Pada peristiwa hamburan

akan nampak beda fasa dari gelombang terhambur. Hal ini akan lebih jelas jika ditinjau perbedaan fasa antara gelombang terhambur oleh atom pusat dengan atom lain. Secara geometris dapat dilihat pada Gambar 2.4.



Gambar 2.4 Peristiwa hamburan oleh unit sel

Pandang A sebagai atom pusat dan diasumsikan terjadi difraksi pada bidang (h00). Ini berarti bahwa hukum Bragg dipatuhi pada refleksi bidang (h00).

Beda lintasan antara 2' dan 1' adalah :

$$\delta_{2,1'} = MCN = 2d_{h00} \sin \theta = \lambda \quad (2.5)$$

dari definisi indeks Miller, diperoleh :

$$d_{h00} = \frac{a}{h} \quad (2.6)$$

Refleksi sinar-x pada atom B dengan jarak x dari atom A dihamburkan pada arah yang sama. Jika diasumsikan bahwa refkesi yang terjadi mematuhi

hukum Bragg seperti refleksi (h00), maka beda lintasan antara sinar 3' dan 1' lebih kecil dari satu panjang gelombang dan dapat dinyatakan :

$$\delta_{31'} = RBS = \frac{AB}{AC}(\lambda) = \frac{x}{\left(\frac{a}{h}\right)}\lambda \quad (2.7)$$

Beda fasa dinyatakan dalam ukuran sudut , beda lintasan dinyatakan dengan panjang gelombang. Hubungan antara beda fasa ( $\phi$ ) dengan beda lintasan ( $\delta$ ) adalah satu panjang gelombang ( $\lambda$ ) sama dengan  $360^\circ$  atau  $2\pi$  radian, secara matematis ditulis :

$$\phi = \frac{\delta}{\lambda}(2\pi) \quad (2.8)$$

Keuntungan dalam penggunaan ukuran sudut yaitu menegaskan bahwa beda fasa tidak bergantung pada panjang gelombang. Beda fasa antara gelombang terhambur oleh atom B dan atom A adalah :

$$\phi_{31'} = \frac{\delta_{31'}}{\lambda}2\pi = \frac{2\pi hx}{a} \quad (2.9)$$

jika posisi atom B dinyatakan dengan fraksi koordinat u, maka beda fasa menjadi, maka beda fasa menjadi  $\phi = 2\pi hu$ .

Atom B terletak pada koordinat ruang tiga dimensi seperti Gambar 2.4, dimana atom B mempunyai koordinat (x,y,z) atau dinyatakan dalam fraksi

koordinat sebagai  $\left(\frac{x}{a}, \frac{y}{b}, \frac{z}{c}\right)$  ,masing-masing sama dengan (u,v,w).

Dengan demikian beda fasa antara gelombang terhambur oleh atom B dan atom A pusat untuk refleksi hkl dinyatakan sebagai :



$$\phi = 2\pi(hu + kv + lw) \quad (2.10)$$

persamaan 2.10 adalah persamaan umum yang dapat digunakan pada unit sel.

Tinjauan tambahan adalah penghambur gelombang oleh atom-atom dalam unit sel. Amplitudo gelombang oleh atom penghambur dinyatakan dengan  $f_j$ . Dari persamaan 2.10 dihubungkan dengan faktor hamburan, maka gelombang terhambur dinyatakan dalam bentuk eksponensial kompleks sebagai :

$$Ae^{i\phi} = f e^{2\pi(hu + kv + lw)} \quad (2.11)$$

Resultan gelombang terhambur oleh semua atom dalam unit sel disebut dengan faktor struktur, diperoleh dengan penjumlahan semua gelombang penghambur dari seluruh atom. Jika unit sel berisi 1,2,3,...N atom, dengan fraksi koordinat  $u_1v_1w_1, u_2v_2w_2, u_3v_3w_3, \dots, u_nv_nw_n$ , dan atom penghambur  $f_1, f_2, f_3, \dots, f_n$ , maka faktor struktur yang diberikan dari refleksi bidang hkl adalah :

$$F = f_1 e^{2\pi(hu_1 + kv_1 + lw_1)} + f_2 e^{2\pi(hu_2 + kv_2 + lw_2)} + \dots + f_n e^{2\pi(hu_n + kv_n + lw_n)} \quad (2.12)$$

faktor struktur untuk N atom dalam unit sel adalah :

$$F_{hkl} = \sum_1^N f_n e^{2\pi(hu_n + kv_n + lw_n)} \quad (2.13)$$

persamaan 2.13 dapat ditulis dalam suku trigonometri menjadi :

$$F = \sum_1^N f_n \left[ \cos 2\pi(hu_n + kv_n + lw_n) + i \sin 2\pi(hu_n + kv_n + lw_n) \right] \quad (2.14)$$

secara umum persamaan 2.14 dapat ditulis dalam bilangan kompleks menjadi :

$$F = a + ib \quad (2.15)$$

dengan :

$$a = \sum_1^N f_n \cos 2\pi(hu_n + kv_n + lw_n) \quad (2.16)$$

$$b = \sum_1^N f_n i \sin 2\pi(hu_n + kv_n + lw_n) \quad (2.17)$$

Intensitas terdifraksi oleh semua atom dalam sel satuan dapat diperoleh dengan mengalikan persamaan 2.15 dengan kompleks konjugatnya seperti di bawah ini :

$$|F|^2 = (a + ib)(a - ib) = a^2 + b^2 \quad (2.18)$$

## 2.2 Fungsi Profil Pseudo-Voight

Fungsi profil pseudo-voight digunakan untuk memodel bentuk puncak, didekati dengan manipulasi fungsi Gausssian dan Lorentzian. Komponen Gausssian dan Lorentzian mewakili pengaruh, masing-masing terhadap resolusi alat dan perluasan ukuran partikel.

Perumusan fungsi profil pseudo-voight dapat diberikan dari general structure analysis system (GSAS) sebagai berikut :

$$H(\Delta T) = \frac{1}{6} \left[ \frac{1}{F(\Delta T)} + 4F(\Delta T + \frac{1}{4} AsCot2\theta_B) + F(\Delta T + AsCot2\theta_B) \right] \quad (2.19)$$

dengan

$$F(\Delta T) = \eta L(\Delta T, \Gamma) + (1 - \eta)G(\Delta T) \quad (2.20)$$

hubungan full weight half-maximum (FWHM) dengan komponen Gausssian dan Lorentzian dapat ditampilkan dengan ekspansi deret di bawah ini :

$$\eta = 1,36603q - 0,47719q^2 + 0,11116q^3 \quad (2.21)$$

dengan :

$$q = \frac{\Gamma_L}{\Gamma} \quad (2.22)$$

koefisien pada persamaan 2.22 digunakan karena adanya faktor normalisasi pada persamaan 2.21.

$\eta$  dan  $\Gamma$  digunakan sebagai variabel dalam proses pencocokan. Jika  $\Gamma$  sukar menghubungkan dengan parameter-parameter fisika,  $\Gamma_G$  dan  $\Gamma_L$  digunakan secara langsung. Pendekatan yang digunakan untuk menentukan  $\Gamma$  adalah ekspansi deret yang lain, diturunkan dari seperangkat komputer generated yaitu :

$$\Gamma = \left( \Gamma_G^5 + A\Gamma_G^4\Gamma_L + B\Gamma_G^3\Gamma_L^2 + C\Gamma_G^2\Gamma_L^3 + D\Gamma_G\Gamma_L^4 + \Gamma_L^5 \right)^{0,2} \quad (2.23)$$

dimana, nilai konstanta persamaan 2.24 adalah :

$$A = 2,69269 \quad C = 4,47163$$

$$B = 2,42843 \quad D = 0,07842$$

dan persamaan sigma gaussian dan lorentzian masing-masing adalah :

$$\Gamma_G = U \tan^2 \theta + V \tan^2 \theta + W + P \quad (2.24)$$

$$\Gamma_L = X \tan \theta + \frac{Y}{\cos \theta} \quad (2.25)$$

perbedaan secara horisontal dan berhingga tentu akan menyebabkan pelebaran puncak, maka asumsi perbedaan puncak seperti ini secara tepat digambarkan dengan fungsi Gaussian :

$$G(\Delta T) = \frac{1}{\sqrt{\sigma^2 2\pi}} \exp\left(-\Delta T^2 / 2\sigma^2\right) \quad (2.26)$$

# FISIKA



# MAKASSAR

## BAB 3

### METODE PENELITIAN

Dalam menentukan pola intensitas difraksi serbuk silikon didasarkan pada tahapan-tahapan proses sebagai berikut :

1. **Studi literatur**, dalam hal ini mencari dan mempelajari fungsi-fungsi profil yang berkaitan dengan penentuan pola intensitas difraksi dari berbagai sumber pustaka (referensi).
2. Membuat rancangan pola intensitas dari pemodelan matematik yang telah ditetapkan.
3. Menguji rancangan pola intensitas dalam bentuk program (perangkat lunak) dengan menggunakan compiler Bahasa C++ yang dalam hal ini **BORLAND C++** versi 5.0 oleh **Borland International**.
4. Membandingkan hasil yang diperoleh dengan hasil eksperimen.

# FISIKA



# MAKASSAR

## BAB 4

### HASIL DAN PEMBAHASAN

Proses percobaan dari perangkat lunak ini dilakukan pada komputer dengan prosesor 80486, akan tetapi untuk pemrosesan dengan kecepatan tinggi dapat dipergunakan prosesor yang lebih tinggi. Media tampilan yang digunakan adalah monitor jenis VGA yang dapat menampilkan 16 tingkat keabuan dengan resolusi 640 X 480 titik.

#### 4.1 Analisis Keperluan

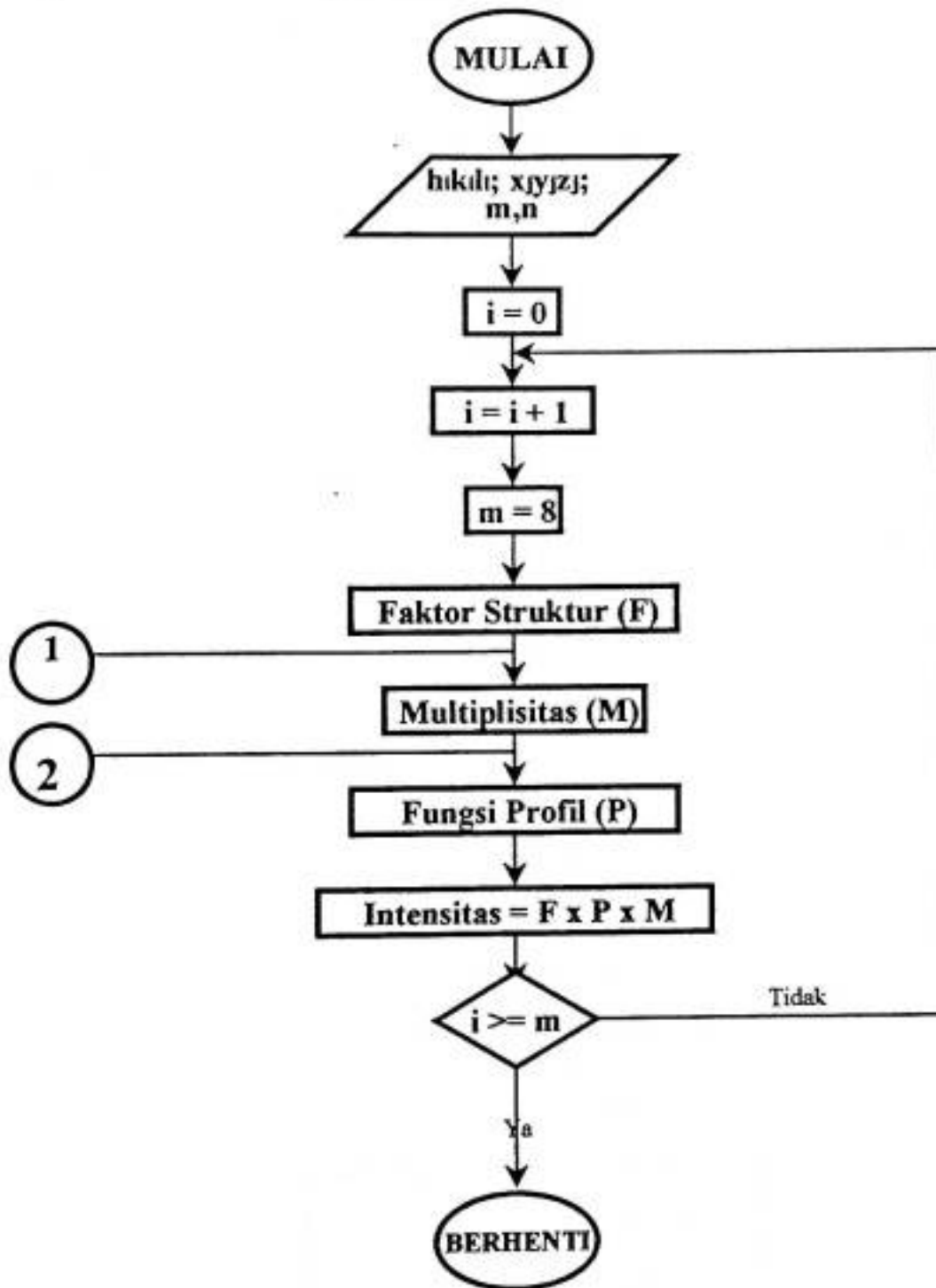
Dari Hasil Eksperimen terhadap serbuk Silikon dengan menggunakan High Resolution Powder Diffractometer (HRPD) menggunakan sumber neutron akan diperoleh antara lain : data posisi atom, parameter kisi, indeks Miller dan faktor multiplisitas.

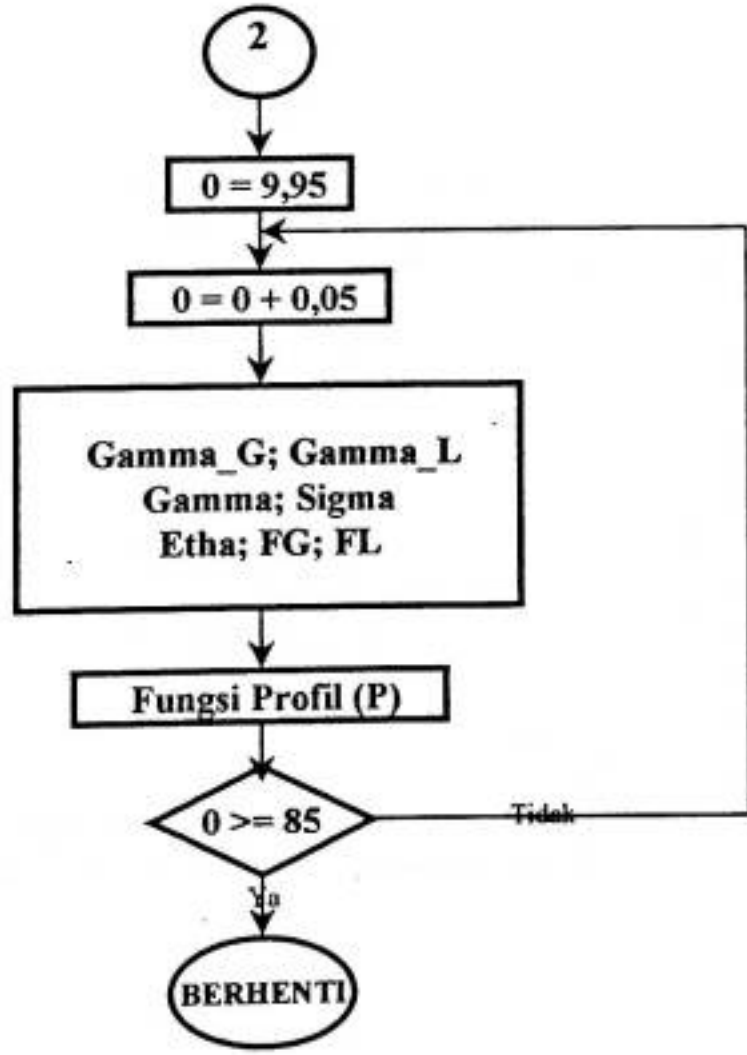
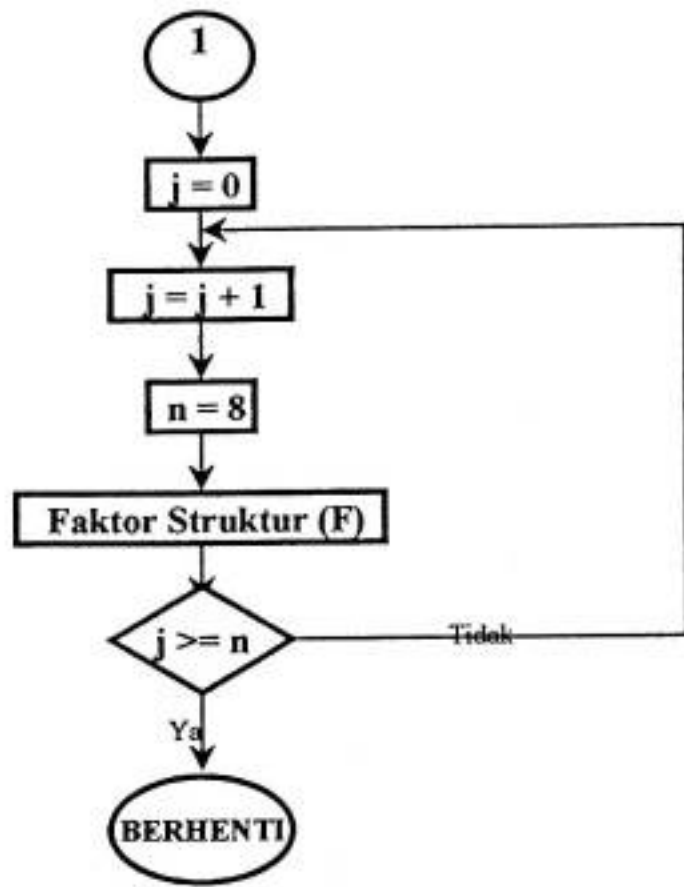
#### 4.2 Diagram Alir (Flow Chart) Pemrograman

Diagram alir (flow chart) merupakan bagan yang menjelaskan suatu deretan blok dan anak panah yang menyatakan urutan proses tertentu.



## 4.2 Diagram Alir Pemrograman





## 4.3 Desain Program

Notasi-notasi :

- $\lambda$  = Sumber neutron
- $x, y, z$  = Posisi atom
- $hkl$  = Bidang penghambur
- $a$  = Parameter kisi
- $U, V, W, P, X, Y$  = Konstanta profil
- $A, B, C, D$  = Konstanta fungsi Gamma/T
- $K$  = Faktor skala
- $M$  = Faktor multiplisitas

### 4.3.1 Deklarasi

Deklarasi merupakan instruksi kepada kompiler agar menyediakan byte memori yang akan diterima dan memeriksa ketepatan pemanggilan fungsi. Variabel dan fungsi harus dideklarasikan sebelum digunakan.

#### Deklarasi variabel

Variabel C ddideklarasikan dengan cara menspesifikasikan nama tipe sebelum nama variabel. Secara umum bentuk deklarasi variabel adalah :

**tipe variabel;**

Keterangan :

**tipe** : adalah tipe data yang diperlukan.

**variabel** : satu atau lebih variabel yang dituiis dengan tanda pisah koma  
(,).

Variabel pada program ini dideklarasikan dengan menggunakan penunjuk.

### Deklarasi fungsi

seperti cara mendeklarasikan Variabel, deklarasi fungsi juga menspesifikasikan nama tipe sebelum nama fungsi. Secara umum bentuk deklarasi fungsi adalah :

```
tipe nama_fungsi(tipe parameter1, tipe parameter2, ...);
```

Keterangan :

- **tipe**, tipe fungsi tergantung dari tipe data hasil balik yang akan dikembalikan oleh fungsi. Jika hasil balik dari fungsi misalnya berupa nilai numerik, maka tipe fungsi dapat ditulis **float**, **double** atau **long double**. Tapi jika fungsi tidak memberikan hasil balik, maka ditulis **void**.
- **nama\_fungsi** merupakan nama fungsi.
- **tipe parameter** merupakan tipe parameter. Untuk menyatakan fungsi tanpa parameter dispesifikasikan **Void**, bila tipe parameter tidak dispesifikasikan bawaannya adalah **void**.

Data  $h, k, l; x, y, z; U, V, W, P, X, Y$  dideklarasikan berupa fungsi. Data-data tersebut deklarasi fungsinya dirancang sebagai berikut :

1. **double** interferensi(**double**  $h$ , **double**  $k$ , **double**  $l$ );
2. **double** gamma\_g(**double**  $\theta$ , **double**  $U$ , **double**  $V$ , **double**  $W$ , **double**  $P$ );
3. **double** gamma\_l(**double**  $\theta$ , **double**  $X$ , **double**  $Y$ );
4. **double** eta(**double**  $\theta$ , **double**  $U$ , **double**  $V$ , **double**  $W$ , **double**  $P$ , **double**  $X$ , **double**  $Y$ );
5. **double** lorentzian(**double**  $\theta$ , **double**  $\arg$ , **double**  $U$ , **double**  $V$ , **double**  $W$ , **double**  $P$ , **double**  $X$ , **double**  $Y$ );
6. **double** gaussian (**double**  $\theta$ , **double**  $\arg$ , **double**  $U$ , **double**  $V$ , **double**  $W$ , **double**  $P$ , **double**  $X$ , **double**  $Y$ );
7. **double** gamma(**double**  $\theta$ , **double**  $\arg$ , **double**  $U$ , **double**  $V$ , **double**  $W$ , **double**  $P$ , **double**  $X$ , **double**  $Y$ );
8. **double** thetaBragg(**double**  $h$ , **double**  $k$ , **double**  $l$ );
9. **double** d\_spasing(**double**  $\theta$ );
10. **double** thermal(**double**  $\theta$ );

Data yang mempunyai harga sebuah konstanta dideklarasikan dengan tipe data integer (ditulis "**int**") seperti :  $\lambda, a, U, V, W, P, X, Y, A, B, C, D$  sedangkan data yang mengembalikan nilai dideklarasikan dengan tipe data **double** (ditulis "**double**") seperti :  $h, k, l$ .

variabel dan fungsi di atas dirancang dengan memanfaatkan fasilitas kelas (ditulis "**class**") yang disediakan oleh C++. Secara umum bentuk class adalah sebagai berikut :

```
class nama_kelas
{
  private:
    tipe nama_anggota_private1;
    tipe nama_anggota_private2;
    ...
  public:
    tipe nama_anggota_public1;
    tipe nama_anggota_public2;
    ...
}(nama_objek);
```

Keterangan;

- **nama\_kelas** adalah nama kelas yang diciptakan. Persyaratan nama kelas seperti nama struktur.
- **tipe** menyatakan tipe data anggota kelas.
- **nama\_anggota\_private1** sampai dengan **nama\_anggota\_public2** menyatakan nama anggota kelas. Anggota kelas dapat berupa data maupun fungsi. Anggota kelas berupa fungsi disebut dengan fungsi kelas dan anggota anggota kelas berupa data disebut data kelas.

Deklarasi Fungsi-fungsi anggota dapat dinyatakan termasuk kelompok bagian public, private atau protected. Pengelompokan fungsi-fungsi anggota

tersebut harus diberi spesifikasi perubah akses (access modifier) anggota kelas dengan tanda titik dua (:). Anggota private suatu kelas hanya dapat diakses oleh fungsi anggota kelas tersebut; anggota protected suatu kelas selain dapat diakses oleh fungsi anggota, juga dapat diakses oleh fungsi anggota kelas turunan; sedangkan anggota public suatu kelas selain dapat diakses oleh anggota kelas turunan juga dapat diakses oleh fungsi luar.

Fungsi-fungsi anggota tersebut dikelompokkan dalam bagian public agar dapat diakses oleh fungsi luar.

Didalam kelas ini juga diciptakan struktur (ditulis "struct") dengan nama struktur yaitu matrix, digunakan untuk anggota kelas yang butuh proses iterasi.

#### **4.3.2 Definisi fungsi**

Definisi fungsi adalah menspesifikasi tugas fungsi tertentu. Mendefinisikan fungsi secara umum terdiri dari dua komponen utama yaitu definisi fungsi dan tubuh fungsi. Definisi fungsi sama seperti mendeklarasikan fungsi hanya pada definisi fungsi tidak diberi tanda titik koma di akhir definisi dan tubuh fungsi berupa statement-statement yang akan melakukan tugas tertentu. statement-statement tersebut diapit dengan tanda kurung kurawal buka dan tutup, secara umum bentuk definisi fungsi adalah sebagai berikut :

```
tipe nama_fungsi(argumen_1, argumen_2, ...)
```

```
{
```

```
..... Statement.....
```

```
..... Statement.....
```

```
}
```

Mendefinisikan fungsi dapat dilakukan langsung dalam definisi kelas atau di luar kelas. Fungsi-fungsi anggota dalam program ini didefinisikan diluar kelas. Agar kompiler tahu bahwa fungsi-fungsi anggota tersebut merupakan anggota kelas hkl, maka sebelum nama fungsi-fungsi anggota tersebut dispesifikasikan kelas hkl dengan menggunakan operator selektor kelas ("::").

### 4.3.3 Klasifikasi

Struktur dan Anggota-anggota kelas dideklarasikan pada sebuah file judul yang dikenal dengan nama file header atau sebuah file yang berektensi (".h"). Sedangkan definisi fungsi-fungsi anggota tersebut diimplementasi pada sebuah file yang dikenal dengan nama file cpp atau sebuah file yang berektensi (".cpp").





#### 4.3.4 Implementasi Program Penentuan Pola Intensitas

Hasil eksekusi program yang telah disusun, ditampilkan dalam bentuk tabel, yaitu tabel antara dua-theta Bragg dengan Intensitas dapat dilihat pada *tabel 4.1*.

#### 4.3.5 Hasil Pola Intensitas Difraksi

Hasil perhitungan yang diperoleh disajikan secara sederhana dalam bentuk tabel di bawah ini :

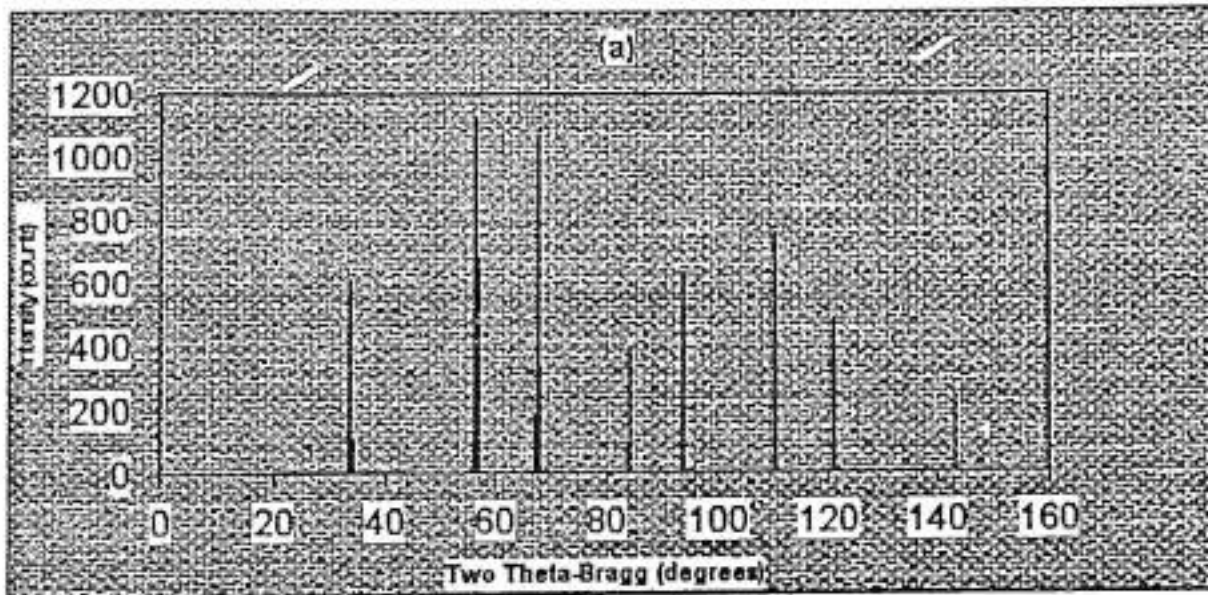
Line	hkl	$\theta_{\text{Bragg}}$ (degrees)	I (counts)
1.	111	33,8	610,7822
2.	220	56,7	1128,415
3.	311	67,6	1067,466
4.	400	84,3	390,8664
5.	331	93,9	630,1616
6.	422	110,5	758,2908
7.	511	121,4	300,0321
8.	440	143,3	226,0889

*Tabel 4.1 tabel hasil perhitungan*

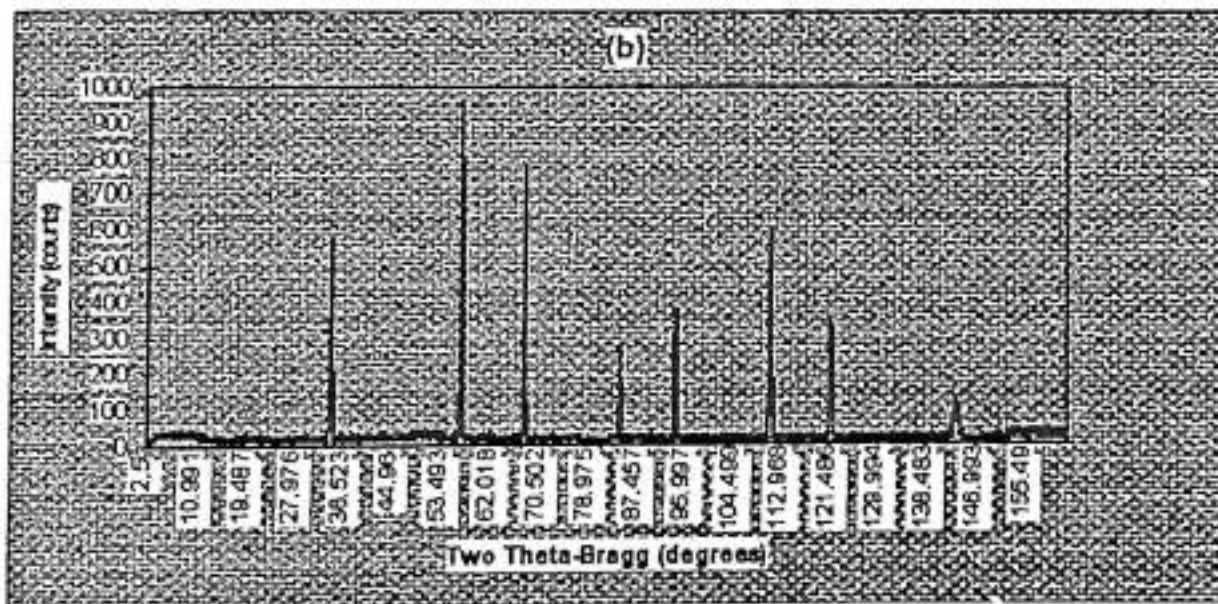
Adapun dalam bentuk pola intensitas difraksi dapat dilihat pada Gambar 4.1.

#### 4.3.6 Analisis Hasil Perancangan Pola Intensitas

Pola intensitas difraksi hasil perhitungan dan eksperimen disajikan di bawah ini :



Gambar 4.1 Pola intensitas difraksi hasil perhitungan



Gambar 4.2 Pola intensitas difraksi hasil eksperimen

Sehingga dapat dikatakan bahwa pola difraksi hasil perhitungan sama dengan pola difraksi hasil eksperimen.

# FISIKA



# MAKASSAR

## BAB 5

### SIMPULAN DAN SARAN

#### 5.1 SIMPULAN :

Perangkat lunak yang telah dirancang dapat menentukan pola difraksi intensitas polikristal dengan menggunakan material silikon standar.

Berdasarkan analisa dan pembahasan yang telah diuraikan, maka simpulan yang dapat diambil adalah :

- 1) Pola intensitas difraksi serbuk silikon secara visual dapat ditampilkan dengan menggunakan komputer. Tampilan pola yang dimaksudkan, terwujud melalui pembuatan software (program) yang dirancang dengan menggunakan compiler bahasa pemrograman C++ yang dalam hal ini BORLAND C++. Program yang dihasilkan tersebut, terwujud melalui suatu pemodelan matematik yang secara lengkap terlihat seperti pada lampiran 1.
- 2) Dari hasil yang diperoleh, maka akan tampak persamaan pola intensitas perhitungan dengan pola intensitas eksperimen. Hal ini terlihat dari persamaan sudut difraksi yang dihasilkan dari perhitungan dengan sudut difraksi hasil eksperimen, demikian pula dengan Intensitas.

## **5.2 SARAN :**

Dari hasil yang diperoleh, tampak bahwa masih terlihat kekurangan. Untuk mendapatkan hasil yang lebih optimal, penulis mengharapkan adanya penelitian yang lebih lanjut mengenai penyeleksian parameter-parameter profil pseudo-voight dengan menggunakan metode simulated annealing agar diperoleh intensitas difraksi yang dimaksudkan.

# FISIKA



# MAKASSAR

## DAFTAR PUSTAKA

1. Cullity, B. D. ; *Elements of X-Ray Diffraction*, Second Edition Addition  
Wisley Publishing Company Inc., USA, 1997.
2. Howard, C. J. ; *Journal of Applied Crystallography : The Approximation  
of Asymmetric Neutron Powder Diffraction Peaks by Sums of  
Gaussians*, 15, (615-620), Australia, 1982.
3. Thompson, P., Hastings, J. B. ; *Journal of Applied Crystallography :  
Rietveld Refinement of Debye-Scherrer Synchotron X-ray Data  
from Al<sub>2</sub>O<sub>3</sub>*, 20, (79-83), USA, 1987.
4. GSAS (General Structure Analysis System) Technical Manual, USA, 1984.
5. Herianto T. ; *Tuntunan Praktis Pemrograman C++*, PT Elex Media  
Competindo, Jakarta, 1995.
6. Setiadi. ; *Tuntunan Praktis Pemrograman Borland C++*, PT Elex Media  
Competindo, Jakarta, 1998.
7. Potts S., Walnum C. ; *Pemrograman Berorientasi Objek dengan Borland  
C++ (Special Edition Using Borland C++ 4.5)*, Terjemahan oleh  
Putrajaya, I., Widjoseno, D., ANDI, Yogyakarta, 1997.
8. Van Vlask, L. ; *Ilmu dan Teknologi Bahan (Ilmu Logam dan Bukan Logam)*,  
Terjemahan oleh Sriati, Dj., Penerbit Erlangga, 1991.

# FISIKA



# MAKASSAR



## LAMPIRAN 1

### LISTING PROGRAM PENENTUAN POLA INTENSITAS DIFRAKSI SILIKON STANDARD (File Header)

```
#ifndef _MATRIX_H_
#define _MATRIX_H_
class matrix {
    struct matrep {
        double **m; // pointer to the matrix
        int r,c; // number of rows and columns
        int n; // reference count
    } *p;

private:
    void error(char * msg1, char * msg2 = "");
public:
    matrix(int mrows = 1, int columns = 1, double initval = 0);
    matrix(int mrows, int columns, double* initvalues);
    matrix(char * flag, int dimension); // create an ident matrix
    matrix(char * matfile); // read from a "standard" matrix file
    matrix(matrix& x); // copy-initializer
    ~matrix();
    int rows() const { return p->r; }; // rows in matrix
    int cols() const { return p->c; }; // cols in matrix
    matrix operator=(const matrix& rval); // matrix assignment
    // Write a "standard" matrix file:
    matrix operator+(const matrix& rval); // matrix addition
    matrix operator+(const double rval); // scalar addition
    matrix operator-(const matrix& rval); // matrix subtraction
    matrix operator-(const double rval); // scalar subtraction
    matrix operator-(); // unary minus
    matrix operator*(const matrix& rval); // matrix multiplication
    matrix operator*(const double rval); // scalar multiplication
    double & val(int row, int col); // element selection;
    // can be used to read or write an element.
    void print(char * msg = ""); // print matrix with a message
    matrix profil(int theta_banyaknya, double theta_min, double theta_del,
        double h, double k, double l);
    void intensitas();
```

```

double interferensi(double h,double k,double l);
double eta(double theta,double U,double V,double W,double P,double X,double Y);
double lorentzian(double theta,double arg,double U,double V,double W,double
    P,double X,double Y);
double gaussian(double theta,double arg,double U,double V,double W,double P);
double gamma_g(double theta,double U,double V,double W,double P);
double gamma_l(double theta,double X,double Y);
double gamma(double theta,double U,double V,double W,double P,double X,double
    Y);
double ThetaBragg(double h, double k, double l);
double d_spacing(double theta);
double Thermal(double theta);
double F_L(double theta);

private: // functions used by inverse() and determinant()
void copy_column(matrix& m, int from_col, int to_col);
void deepcopy(matrix& from, matrix& to); // make an image
double & mval(int row, int col) {
    return (p->m[row][col]);
}
};
#endif // _MATRIX_H_

```

## LAMPIRAN 2

### LISTING PROGRAM PENENTUAN POLA INTENSITAS DIFRAKSI SILIKON STANDARD (File CPP)

```
#define COMPACT

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include "matrix.h"

void matrix::error(char * msg1, char * msg2) {
    fprintf(stderr, "matrix error: %s %s\n", msg1, msg2);
    exit(1);
}

matrix::matrix(int mrows, int columns, double initval)
{
    // create the structure:
    p = new matrep;
    p->r = mrows;
    p->c = columns;

    // allocate memory for the actual matrix:
    p->m = new double *[mrows];
    for (int x = 0; x < mrows; x++)
        p->m[x] = new double[columns];
    p->n = 1; // so far, there's one reference to this data
    for (int i=0; i < mrows; i++) {
        for (int j = 0; j < columns; j++)
            mval(i,j) = initval;
    }
}

matrix::matrix(int mrows, int columns, double* initvalues)
{
    //printf("mrows = %d, columns = %d\n", mrows, columns);
    // create the structure:
```

```

p = new matrep;
p->r = mrows;
p->c = columns;
// allocate memory for the actual matrix:
p->m = new double *[mrows];
for (int x = 0; x < mrows; x++)
    p->m[x] = new double[columns];
p->n = 1; // so far, there's one reference to this data
int c = 0;
for (int i=0; i < mrows; i++) {
    for (int j = 0; j < columns; j++)
        mval(i,j) = initvalues[c++];
}
}

// create an identity matrix:
matrix::matrix(char * flag, int dimension)
{
    if (flag[0] != 'I')
        error("to create an identity matrix: "
            "matrix(\"I\",dimension)");
    p = new matrep;
    p->r = dimension;
    p->c = dimension;
    p->m = new double *[dimension];
    for (int x = 0; x < dimension; x++)
        p->m[x] = new double[dimension];
    p->n = 1;
    for (int i=0; i < dimension; i++) {
        for (int j = 0; j < dimension; j++)
            mval(i,j) = (i == j ? 1 : 0);
    }
}

// error message when trying to read a "standard"
// matrix file:
static char nonstandard[] =
    "is a `non-standard' file. A `standard' matrix file must\n"
    "start with the dimensions of the matrix, i.e.: \n"
    "\t rows=12 columns=14\n or abbreviated: \n\t r=12 c=14\n"
    "Notice rows appear before columns, and chars are lower case\n"
    "comments follow `#' signs to end of line, data follows :::\n";

matrix::matrix(matrix& x) {
    x.p->n++; // we're adding another reference.
}

```

```

    p = x.p; // point to the new matrep.
}

```

```

matrix::~~matrix() {
    if (--p->n == 0) { // if reference count goes to 0
        for (int x = 0; x < rows(); x++)
            delete p->m[x];
        delete p->m; // delete data
        delete p;
    }
}

```

```

matrix matrix::operator=(const matrix& rval) {
    // clean up current value:
    if (--p->n == 0) { // If nobody else is referencing us...
        for (int x = 0; x < rows(); x++)
            delete p->m[x];
        delete p->m; // ...nobody else can clean us up...
        delete p;
    }
    // connect to new value:
    rval.p->n++; // tell the rval it has another reference
    p = rval.p; // point at the rval matrep
    return *this;
}

```

```

matrix matrix::operator+(const matrix& arg) {
    if(( rows() != arg.rows() ) || ( cols() != arg.cols() ))
        error("must have equal dimensions for addition!");
    matrix sum(rows(),cols());
    for (int i=0; i < rows(); i++) {
        for (int j = 0; j < cols(); j++)
            sum.mval(i,j) = mval(i,j) + arg.p->m[i][j];
    }
    return sum; // see note for operator*()
}

```

```

matrix matrix::operator+(const double arg) {
    matrix sum(rows(),cols());
    for (int i=0; i < rows(); i++) {
        for (int j = 0; j < cols(); j++)
            sum.mval(i,j) = mval(i,j) + arg;
    }
    return sum; // see note for operator*()
}

```

```
}
```

```
matrix matrix::operator-(const matrix& arg) {  
    if(( rows() != arg.rows()) || ( cols() != arg.cols()))  
        error("must have equal dimensions for subtraction!");  
    matrix sum(rows(),cols());  
    for (int i=0; i< rows(); i++) {  
        for (int j = 0; j < cols(); j++)  
            sum.mval(i,j) = mval(i,j) - arg.p->m[i][j];  
    }  
    return sum; // see note for operator*()  
}
```

```
matrix matrix::operator-(const double arg) {  
    matrix sum(rows(),cols());  
    for (int i=0; i< rows(); i++) {  
        for (int j = 0; j < cols(); j++)  
            sum.mval(i,j) = mval(i,j) - arg;  
    }  
    return sum; // see note for operator*()  
}
```

```
matrix matrix::operator-() {  
    matrix unaryminus(rows(),cols());  
    for (int i=0; i< rows(); i++) {  
        for (int j = 0; j < cols(); j++)  
            unaryminus.mval(i,j) = -mval(i,j);  
    }  
    return unaryminus;  
}
```

```
matrix matrix::operator*(const matrix& arg) {  
    if( cols() != arg.rows())  
        error("# rows of second mat must equal "  
            "# cols of first for multiply!");  
    matrix result(rows(),arg.cols());  
    for(int row = 0; row < rows(); row++) {  
        for(int col = 0; col < arg.cols(); col++){  
            double sum = 0;  
            for(int i = 0; i < cols(); i++)  
                sum += mval(row,i) * arg.p->m[i][col];  
            result.mval(row,col) = sum;  
        }  
    }  
    return result; // Returning a local variable?
```

```

// copy-initializer happens before the destructor,
// so reference count is 2 when destructor is called,
// thus destructor doesn't free the memory.
}

matrix matrix::operator*(const double arg) {
    matrix result(rows(),cols());
    for (int i=0; i< rows(); i++) {
        for (int j = 0; j < cols(); j++)
            result.mval(i,j) = mval(i,j) * arg;
    }
    return result;
}

double & matrix::val(int row, int col) {
    if (row >= 0 && row < rows() && col >= 0 && col < cols())
        return (mval(row,col));
    else
        error("index out of range");
}

void matrix::print(char *msg) {
    if (*msg) printf("%s:\n",msg);
    for (int row=0; row< rows(); row++){
        for (int col = 0; col < cols(); col++)
            printf("%6.6f ", mval(row,col));
        printf("\n");
    }
}

/*****
The private support functions for determinant & inverse.
*****/

// copy the from_col of mm to the to_col of "this"
void matrix::copy_column(matrix& mm, int from_col, int to_col) {
    if(rows() != mm.rows())
        error("number of rows must be equal for copy_column()");
    for(int row=0; row < rows(); row++)
        mval(row,to_col) = mm.mval(row,from_col);
}

main()
{
    matrix A;

```

```

A.intensitas();
}

void matrix::intensitas()
{
int jj;
matrix A("hkl-singkat.dat");
matrix B;
matrix T;
matrix U;
matrix DD;
matrix D(A.rows(),1,0.0);
matrix skala("skala.dat");
double
pi,temp,temp0,temp1,temp2,C,theta_deg,theta_max_deg,theta_min_deg,theta_del_deg,
theta,theta_max,theta_min,theta_del,theta_banyaknya,faktorlorentz,polarisasilorentz,
thermal;
pi=acos(-1.);
theta_min_deg=10.;
theta_max_deg=85.;
theta_del_deg=0.05;
theta_min=theta_min_deg*pi/180.;
theta_max=theta_max_deg*pi/180.;
theta_del=theta_del_deg*pi/180.;
theta_banyaknya=1+(theta_max-theta_min)/theta_del;
jj= (int)theta_banyaknya;
printf("harga jj %d",jj);
matrix E(jj,3,0.0);
int i=0;
for (int i=0;i<A.rows();i++)
{
B=A.profil(jj,theta_min,theta_del,
A.mval(i,0),A.mval(i,1),A.mval(i,2));
C=B.interferensi(A.mval(i,0),A.mval(i,1),A.mval(i,2));
D=B*C*skala.mval(0,0)*A.mval(i,3);
theta=theta_min;
theta_deg=theta_min_deg;
for (int j=0;j<jj;j++)
{
//double th=theta+j*theta_del;
E.mval(j,0)=2.*(theta_deg+j*theta_del_deg);
E.mval(j,1)=A.d_spacing(theta+j*theta_del);
}
}
}

```



```

thermal=T.Thermal(theta+j*theta_del);
E.mval(j,2)=E.mval(j,2)+temp;

}
}
E.write_standard("hasil.dat","2-theta d-spacing intensitas");

```

```

}

```

```

double matrix::interferensi(double h,double k,double l)
{
matrix A("atom.dat");
matrix B;
double hasil,pi,arg,real,imag;
pi=acos(-1.);
real=0.;
imag=0.;
for(int i=0;i<A.rows();i++)
{
arg=2.*pi*(h*A.mval(i,0)+k*A.mval(i,1)+l*A.mval(i,2));
real=real+cos(arg);
imag=imag+sin(arg);
}
hasil=pow(real,2)+pow(imag,2);
printf("\n hasil interferensi %f",hasil);
return hasil;
}

```

```

matrix matrix::profil(int theta_banyaknya,double theta_min,double theta_del,
double h,double k,double l)
{
double thetabragg,theta,arg[3],f[3],U,V,W,P,X,Y,As,gauss,lorentz,eta,hasil;
matrix A("profil.dat");
matrix B;
matrix C(theta_banyaknya,1,1.0);
U=A.mval(0,0);
V=A.mval(1,0);
W=A.mval(2,0);
P=A.mval(3,0);
X=A.mval(4,0);
Y=A.mval(5,0);
As=A.mval(6,0);
thetabragg=B.ThetaBragg(h,k,l);
double pi=acos(-1);

```



```
printf("\n \n posisi 2-theta Bragg %f",2.*thetabragg*180/pi);
theta=theta_min;
int ii=0;
for(ii=0;ii<theta_banyaknya;ii++){
    eta=A.eta(theta,U,V,W,P,X,Y);
    arg[0]=2.*(theta-thetabragg);
    arg[1]=arg[0]+As/(4.*tan(2.*thetabragg));
    arg[2]=arg[0]+As/tan(2.*thetabragg);
    for(int i=0;i<3;i++){
        lorentz=A.lorentzian(theta,arg[i],U,V,W,P,X,Y);
        gauss=A.gaussian(theta,arg[i],U,V,W,P);
        ff[i]=eta*lorentz+(1.-eta)*gauss;
    }
    theta = theta + theta_del;
    hasil=(ff[0]+4.*ff[1]*As/(4.*tan(2.*thetabragg))+ff[2]*As/tan(2.*thetabragg))/6.;
    C.mval(ii,0)=hasil;
}
return C;
}
```

```
double matrix::gamma_g(double theta,double U,double V,double W, double P)
{
    double suku0, suku1, suku2, suku3, hasil;
    suku0=U*pow(tan(theta),2);
    suku1=V*tan(theta);
    suku2=W;
    suku3=P/pow(cos(theta),2);
    hasil=sqrt(suku0+suku1+suku2+suku3);
    return hasil;
}
```

```
double matrix::gamma_l(double theta,double X,double Y)
{
    double suku0, suku1, hasil;
    suku0=X*sin(theta)/cos(theta);
    suku1=Y/cos(theta);
    hasil=suku0+suku1;
    return hasil;
}
```

```
double matrix::eta(double theta,double U,double V,double W,double P,double
X,double Y)
{
    double q,faktor1,faktor2,hasil;
```

```

matrix A;
faktor1=A.gamma_l(theta,X,Y);
faktor2=A.gamma(theta,U,V,W,P,X,Y);
q=faktor1/faktor2;
hasil=1.36603*q-0.47719*pow(q,2)+0.11116*pow(q,3);
return hasil;
}

```

```

double matrix::lorentzian(double theta,double arg,double U,double V,double
W,double P,double X,double Y)
{
double pi,a,faktor1,faktor2,hasil;
matrix A;
pi=acos(-1.);
a=A.gamma(theta,U,V,W,P,X,Y);
faktor1=a/(2.*pi);
faktor2=pow(a/2.,2)+pow(arg,2);
hasil=faktor1/faktor2;
return hasil;
}

```

```

double matrix::gaussian(double theta,double arg,double U,double V,double
W,double P)
{
double arg1,pi,sigma,faktor1,faktor2,hasil;
matrix A;
pi=acos(-1.);
sigma=A.gamma_g(theta,U,V,W,P)/sqrt(8.*log(2.));
faktor1=sqrt(2.*pi)*sigma;
arg1=-pow(arg,2)/(2*pow(sigma,2));
faktor2=exp(arg1);
hasil=faktor2/faktor1;
return hasil;
}

```

```

double matrix::gamma(double theta,double U,double V,double W,double P,double
X,double Y)
{
matrix Prof;
double x,y,temp,hasil;
x=Prof.gamma_g(theta,U,V,W,P);
y=Prof.gamma_l(theta,X,Y);
const double A=2.69269;
const double B=2.42843;
const double C=4.47163;

```

```

const double D=0.07842;
temp=pow(x,5.)+A*pow(x,4.)*y+B*pow(x,3.)*pow(y,2.)+C*pow(x,2.)*pow(y,3.)+
  +D*x*pow(y,4.)+pow(y,5.);
hasil=pow(temp,0.2);
return hasil;
}

```

```

double matrix::ThetaBragg(double h, double k, double l)
{
double a,d,denom,lambda,temp,thetabragg;
lambda=1.822174;
a=5.4307000;
temp=pow(h,2)+pow(k,2)+pow(l,2);
denom=sqrt(temp);
d=a/denom;
thetabragg=asin(lambda/(2.*d));
return thetabragg;
}

```

```

double matrix::d_spacing(double theta)
{
double d,denom,lambda;
lambda=1.822174;
denom=2.*sin(theta);
d=lambda/denom;
return d;
}

```

```

double matrix::Thermal(double theta)
{
matrix scale("therm.dat");
double lambda,pi,temp1,temp2,thermal;
pi=acos(-1.);
lambda=1.822174;
temp1=-8*pow(pi,2)*scale.mval(0,0)*pow(sin(theta),2);
temp2=pow(lambda,2);
thermal=exp(temp1/temp2);
return thermal;
}

```



**DÉPARTEMEN PENDIDIKAN DAN KEBUDAYAAN  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN**

KAMPUS UNHAS TAMALANREA JALAN PERINTIS KEMERDEKAAN  
TELP. 510200, 510192, 512016 (PES. 2451, 2616, 2455, 2456, 2457, 2458) UJUNG PANDANG 90245

No. : 142 /J04.12.17/PL.2/97  
1 : Izin Penelitian

13 Februari 1997

kepada : Kepala Pusat Sains dan Materi,  
Badan Tenaga Atom Nasional (PPOM DATAN),  
Kawasan PUSPITEK Serpong Tangerang

D a r a t :

Dengan hormat,

Berdasarkan surat Ketua Jurusan Fisika FMIPA UNHAS No. 020/PT04.FMIPA-03/14/97 tanggal 14 Februari 1997, maka dengan ini kami mohon kiranya kepada mahasiswa tersebut di bawah ini dapat diberi izin melakukan penelitian pada Kantor Pusat Sains dan Materi Badan Tenaga Atom Nasional (PPOM DATAN) Kawasan Puspittek Serpong Tangerang Jawa Barat sebagai salah satu tugas akhirnya, mahasiswa tersebut yaitu :

Nama	: Syaiful Aidi
No. Pokok	: 92 03 170
Jurusan	: Fisika
Prog. Studi	: Fisika

Demikian permohonan kami, atas kerjasamanya terlebih dahulu disampaikan terima kasih.



Asisten Dekan I,

Abdur Jalaluddin  
No. 675 574

diarahkan kepada :  
Kepala Jurusan Fisika  
FMIPA UNHAS



# BADAN TENAGA ATOM NASIONAL

## PUSAT PENELITIAN SAINS MATERI

KAWASAN PUSPIPTK SERPONG - TANGERANG

TELP. 7560562 - 7560537 - 7560538; LANGSUNG : 7560922 - 7560926  
FAX. 7560926, TELEX 46354 PPTAS IA, KAWAT : BATAN JAKARTA INDONESIA

Serpong, 28 Agustus 1998

Nomor : 499/HM 03/1998  
Lampiran : -  
Perihal : Tugas Akhir

Kepada Yth. :  
Ketua Jurusan Fisika FMIPA  
Universitas Hasanuddin  
**UJUNG PANDANG**

Dengan hormat,

Dengan ini diberitahukan bahwa mahasiswa Saudara atas nama :

N a m a : Syaiful Alam  
N I M : 92 03 178  
Jurusan : Fisika, FMIPA

telah menyelesaikan penelitian tugas akhirnya di Pusat Penelitian Sains Materi BATAN Serpong dan juga urusan administrasi telah diselesaikan dengan judul :

### **Perhitungan Intensitas Difraksi Serbuk Silikon dengan Fungsi Profil Pseudo-Voight**

Demikian pemberitahuan ini kami sampaikan, atas perhatian Saudara kami mengucapkan terima kasih.

Pos. KEPALA PUSAT PENELITIAN  
SAINS MATERI,



R. R d w a n  
NIP. 330003472

Tembusan Yth. :  
1. Ka. PPSM  
2. Ka ISN