

**PENGARUH LEBAR KOLIMATOR TERHADAP KUALITAS
CITRA REKONSTRUKSI DENGAN MENGGUNAKAN METODE
ITERASI ALJABAR**

SKRIPSI

1009 07/05

50

9

EVA SUSANTI



**JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS AIRLANGGA
SURABAYA
2005**



**PENGARUH LEBAR KOLIMATOR TERHADAP KUALITAS
CITRA REKONSTRUKSI DENGAN MENGGUNAKAN METODE
ITERASI ALJABAR**

SKRIPSI

**Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Sains
Bidang Fisika pada Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Airlangga Surabaya**

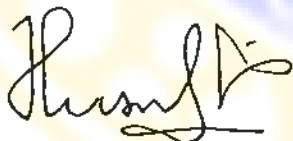
Oleh :

EVA SUSANTI
089912055

Tanggal Lulus : 12 Agustus 2005

Disetujui oleh

Pembimbing I,



Khusnul Ain, S.T., M.Si
NIP. 132 166 487

Pembimbing II,



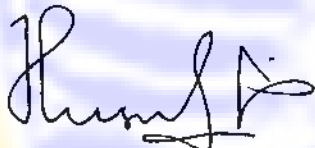
Nuril Ukhrowiyah, S.Si, M.Si
NIP. 132 230 839

LEMBAR PENGESAHAN SKRIPSI

Judul : Pengaruh Lebar Kolimator Terhadap Kualitas Citra Rekonstruksi Dengan Menggunakan Metode Iterasi Aljabar
Penyusun : Eva Susanti
NIM : 089912055
Tanggal Ujian : 12 Agustus 2005


Disetujui Oleh :

Pembimbing I,



Khusnul Ain, S.T., M.Si
NIP. 132 166 487

Pembimbing II,



Nuril Ukhrowiyah, S.Si, M.Si
NIP. 132 230 839

Mengetahui,

Ketua Jurusan Fisika FMIPA

UNAIR



Drs. Fujiyanto, M.S.
NIP. 131 756 001

PEDOMAN PENGGUNAAN SKRIPSI

Skripsi ini tidak dipublikasikan namun tersedia di perpustakaan dalam lingkungan Universitas Airlangga.

Diperkenankan untuk dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seijin dan harus menyebutkan sumbernya sesuai kebiasaan ilmiah.

Dokumen ini merupakan hak milik Universitas Airlangga.

Eva Susanti. 2005. Pengaruh Lebar Kolimator Terhadap Kualitas Citra Rekonstruksi Dengan Menggunakan Metode Iterasi Aljabar., Skripsi ini dibuat dibawah bimbingan Khusnul Ain, S.T, M.Si. dan Nuril Ukhrowiyah, S.Si., M.Si, Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Airlangga.

ABSTRAK

Pembuatan program simulator rekonstruksi citra menggunakan metode iterasi aljabar (ART) dengan lebar kolimator yang berbeda telah dilakukan. Lebar kolimator dibuat dengan variasi $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$. Selain itu, juga telah dibuat program simulator proses *scanning* dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$. Penelitian ini bertujuan untuk mengetahui seberapa besar pengaruh lebar kolimator terhadap kualitas citra yang dihasilkan pada rekonstruksi citra menggunakan metode ART.

Simulator proses *scanning* menghasilkan sinogram. Sinogram tersebut direkonstruksi dengan simulator rekonstruksi citra menggunakan program rekonstruksi dengan metode ART yang dibuat dalam penelitian ini. Sebagai pembandingan dalam penelitian ini sinogram juga direkonstruksi menggunakan metode proyeksi balik terfilter (FBP). Kemudian citra-citra hasil rekonstruksi dianalisis kualitasnya. Kualitas citra hasil rekonstruksi ditentukan dengan membedakan citra-citra hasil rekonstruksi terhadap objek sintetik. Perbedaan dilakukan secara visual dan numerik. Perbedaan secara visual dilakukan dengan membedakan citra-citra hasil rekonstruksi dan profil garis horisontalnya. Perbedaan secara numerik dilakukan dengan menghitung *rmsd*.

Hasil perbedaan secara visual dan numerik menunjukkan kualitas citra hasil rekonstruksi dengan dengan metode ART lebih mendekati objek referensi dibandingkan dengan metode FBP. Hal ini didukung profil garis horisontal citra-citra hasil rekonstruksi berhimpit dengan profil garis horisontal objek referensi. Hasil perbedaan citra-citra hasil rekonstruksi terhadap objek referensi menunjukkan bahwa lebar kolimator yang lebih lebar (dalam penelitian ini diwakili $w = 3\Delta x$ dan $w = 5\Delta x$) dari yang umum digunakan ($w = \Delta x$) relatif tidak berpengaruh pada rekonstruksi citra dengan metode ART. Hasil perbedaan secara numerik menunjukkan bahwa semakin lebar kolimator yang digunakan semakin besar nilai *rmsd* untuk metode FBP.

Kata Kunci : Lebar Kolimator, Rekonstruksi ART, Tomografi Komputer.

Eva Susanti. 2005. *Influence of Collimator Wide To Reconstruction Image Quality By Using Algebra Iteration Method.* This skripsi is made under tuition of Khusnul Ain, S.T, M.Si. dan Nuril Ukhrowiyah, S.Si, M.Si, Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Airlangga.

ABSTRAK

Making the simulator program image reconstruction using algebra iteration method (ART) with different widely collimator is has been conducted. Collimator wide made with variation $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$. Also has been made a simulator program process of scanning with collimator wide $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$.

Scanning simulator process yield the sinogram. The sinogram reconstruction use reconstruction program with ART method made this research. As a comparator in this research sinogram also reconstructed using filtered back projection (FBP) method. Then images of reconstructing result analysed its quality. Image quality result of reconstruction determined by differentiating images of reconstruction result to synthetic object. Differentiation conduted visually and numerik. Differentiation visually is conduted diferentiating images of reconstructing result and its horizontal line profile. Differentiation numerik is conducted by calculating rmsd.

Result of differentiation visually and numerik show the image quality of reconstructing result with the ART method more coming near of reference object compared to FBP method. This supported by a horizontal line profile images of reconstructing result that to squeez with the horizontal line profile of reference object. Result of images differentiation of reconstructing result to reference object indicate that more of wide collimator (in this research is deputized by $w = 3\Delta x$ and $w = 5\Delta x$) from used occasionally ($w = \Delta x$) relatively do not have influential to the image reconstruction with ART method. Result of differentiation by numeric indicate that wide progressively collimator used ever grater assess its rmsd for FBP method.

Keyword : Collimator Wide, ART Reconstruction, Computed Tomography.

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kepada Allah SWT karena dengan rahmat-Nyalah sehingga penulis telah menyelesaikan skripsi dengan judul **“ Pengaruh Lebar Kolimator Terhadap Kualitas Citra Rekonstruksi Dengan Menggunakan Metode Iterasi Aljabar”**.

Skripsi ini merupakan salah satu syarat dalam meraih gelar Sarjana Sains bidang Fisika pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Airlangga Surabaya. Pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Bapak Khusnul Ain, S.T, M.Si, selaku pembimbing I yang telah menyediakan waktu, tenaga, dan pikiran untuk membimbing dan memberikan saran,
2. Ibu Nuril Ukhrowiyah, S.Si, M.Si., selaku pembimbing II atas segala perhatian dan kesabarannya dalam memberikan bimbingan selama menyelesaikan skripsi ini.
3. Bapak Drs. Pujiyanto, M.S ., selaku Ketua Jurusan Fisika FMIPA UNAIR.
4. Bapak Drs. Djoni Izak R., M.Si., selaku dosen wali atas segala dorongan dan nasehatnya.
5. Bapak Drs. Adri Supardi, MS dan Ibu Riries Rulaningtyas, ST selaku dosen penguji yang telah memberikan ilmu serta saran dalam skripsi ini.
6. Keluargaku yang selalu memberikan dorongan moril dan materiil selama ini.
7. Teman-temanku angkatan 99, terutama Diana dan Endro terima kasih sarannya.
8. Ida Puah (00), Indah (00), Citra (98), Purnomo (00) dan Mas Arya atas bantuannya.

9. Semua pihak yang telah membantu dalam pelaksanaan skripsi ini.

Penulis menyadari sepenuhnya masih banyak terdapat kekurangan didalam penyusunan skripsi ini, oleh karena itu saran dan kritik yang membangun dari berbagai pihak sangat penulis harapkan agar skripsi ini lebih sempurna.

Demikian semoga skripsi ini dapat memberikan ilmu dan manfaat demi kemajuan bersama. Amin.

Surabaya, Juli 2005

Penulis

DAFTAR ISI

	Halaman
Abstrak	i
Kata Pengantar	iii
Daftar Isi	v
Daftar Lampiran	vii
Daftar Gambar	viii
Daftar Tabel	x
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Penelitian.....	5
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA	
2.1 Koefisien Serapan Linear	6
2.2 Prinsip Sistem Tomografi Komputer Transmisi Generasi Pertama	8
2.3 Metode Sampling	9
2.4 Metode Rekonstruksi	11
2.4.1 Metode Rekonstruksi ART	11
2.4.2 Metode Rekonstruksi FBP	13

BAB III METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian 15

3.2 Alat dan Bahan Penelitian 15

3.3 Prosedur Penelitian 16

 3.3.1 Proses *Scanning*..... 17

 3.3.2 Proses Rekonstruksi..... 18

3.4 Analisis Data 20

BAB IV HASIL DAN PEMBAHASAN

4.1 Tampilan Simulator Proses *Scanning*

 TK Generasi Pertama 21

4.2 Sinogram Hasil Proses *Scanning*..... 22

4.3 Tampilan Program Rekonstruksi Citra

 Metode Rekonstruksi Aljabar..... 23

4.4 Citra Hasil Rekonstruksi..... 24

4.5 Hasil Analisis Kualitas Citra..... 27

4.6 Pembahasan..... 30

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan..... 33

5.2 Saran..... 34

DAFTAR PUSTAKA 35

LAMPIRAN..... 37

DAFTAR LAMPIRAN

Lampiran		Halaman
A	Program Simulator Proses <i>Scanning</i>	37
B	Program Simulator Proses Rekonstruksi	43
C	Laporan Nilai <i>Discrepancy</i>	54
D	Tabel nilai Rmsd	64

DAFTAR GAMBAR

Gambar		Halaman
Gambar 2.1	Proses pelemahan intensitas sinar gamma oleh suatu bahan dengan ketebalan x	7
Gambar 2.2	Proses pemayaran pada sistem tomografi komputer generasi Pertama	9
Gambar 2.3	Sampling data (a) $w = \Delta x$ (b) $w = 3\Delta x$ (c) $w = 5\Delta x$	10
Gambar 3.1	Objek sintetik : (a) Segitiga.dat (b) Modul.dat	15
Gambar 3.2	Blok Diagram proses simulasi	16
Gambar 3.3	<i>Flow chart</i> Simulator Proses <i>Scanning</i> TK Generasi Pertama	18
Gambar 3.4	<i>Flow chart</i> Program Rekonstruksi Citra Menggunakan Metode ART	19
Gambar 4.1	Tampilan simulator proses <i>scanning</i>	21
Gambar 4.2	Menu File	21
Gambar 4.3	Menu Proses	22
Gambar 4.4	Sinogram dari objek Segitiga (a) Sinogram $w = \Delta x$ (b) Sinogram $w = 3\Delta x$ (c) Sinogram $w = 5\Delta x$	22
Gambar 4.5	Sinogram dari objek Modul (a) Sinogram $w = \Delta x$ (b) Sinogram $w = 3\Delta x$ (c) Sinogram $w = 5\Delta x$	23
Gambar 4.6	Simulator Program Rekonstruksi Citra dengan Metode ART	23
Gambar 4.7	Menu File	24
Gambar 4.8	Menu Pengaturan	24
Gambar 4.9	Menu Iterasi	24

Gambar 4.10	Citra Segitiga hasil rekonstruksi ART (a) $w = \Delta x$ (b) $w = 3\Delta x$ (c) $w = 5\Delta x$	25
Gambar 4.11	Citra Modul hasil rekonstruksi ART (a) $w = \Delta x$ (b) $w = 3\Delta x$ (c) $w = 5\Delta x$	25
Gambar 4.12	Hubungan antara nilai discrepancy dan proses iterasi pada objek Segitiga.dat	25
Gambar 4.13	Hubungan antara nilai discrepancy dan proses iterasi pada objek Modul.dat	26
Gambar 4.14	Citra segitiga hasil rekonstruksi metode FBP (a.1 & a.2) $w = \Delta x$, (b.1& b.2) $w = 3\Delta x$ dan (c.1& c.2) $w = 5\Delta x$	26
Gambar 4.15	Citra Modul hasil rekonstruksi metode FBP (a.1 & a.2) $w = \Delta x$, (b.1& b.2) $w = 3\Delta x$ dan (c.1& c.2) $w = 5\Delta x$	27
Gambar 4.16	Profil garis horisontal dari objek segitiga	28
Gaambar 4.17	Profil garis horisontal dari objek Modul	28
Gambar 4.18	Grafik hubungan antara lebar kolimator dengan nilai rmsd	29

DAFTAR TABEL

Tabel		Halaman
Tabel 4.1	Tabel Lama Proses Rekonstruksi	30



BAB I

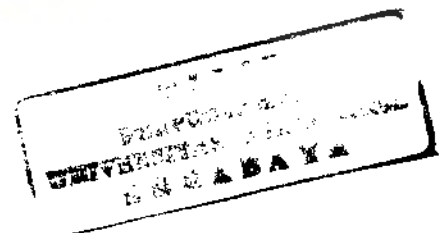
PENDAHULUAN

1.1 Latar Belakang

Teknik tomografi komputer atau TK adalah suatu teknik untuk mendapatkan gambaran internal obyek tanpa merusak atau membelah obyek tersebut. Perkembangan TK berawal dari penemuan sinar-X oleh Roentgen pada tahun 1895 yang selanjutnya menghasilkan teknik radiografi. Sejak penemuan itu radiografi berkembang pesat. Kemudian J.Radon pada tahun 1917 memberikan formulasi matematik untuk merekonstruksi sebuah fungsi dua dimensi yang dikenal dengan transformasi Radon. Hal itu yang menjadi ide dasar teknik tomografi.

Penerapan dari konsep rekonstruksi ini digunakan beberapa ahli seperti Bracewell pada tahun 1956 untuk merekonstruksi citra emisi gelombang mikro dari permukaan matahari dan Cormack pada tahun 1963 yang merekonstruksi citra menggunakan transmisi sinar gamma melalui sebuah silinder Aluminium. Hingga akhirnya Housfield pada tahun 1972 sukses mengimpletasikan idenya membentuk CT Scan (Computed Tomography Scanner), yaitu sebuah sistem tomografi di bidang medis yang berbasis komputer.

Berdasarkan letak sumber radiasi, sistem TK dibedakan menjadi dua yaitu sistem TK transmisi dan emisi. Sistem TK transmisi adalah sistem TK yang sumber radiasinya terletak di luar obyek. Sedangkan sistem TK emisi adalah sistem TK yang sumber radiasi terletak di dalam obyek yang akan diteliti. Sistem TK transmisi lebih



cepat perkembangannya dibanding dengan TK emisi, serta telah banyak diaplikasikan di berbagai bidang baik medis maupun industri (Widodo, 2000).

Berdasarkan berkas sumber radiasi, jumlah detektor serta prinsip pengambilan datanya, dikenal sistem TK generasi pertama, kedua, ketiga, keempat dan kelima. Sistem TK generasi pertama adalah sistem TK paling sederhana karena hanya memanfaatkan sebuah sumber radiasi dengan berkas pensil dan sebuah detektor dengan sumber dan detektor saling tegak lurus terhadap gerak obyek. Sedangkan pada generasi selanjutnya merupakan hasil dari pengembangan TK generasi pertama dengan menambahkan jumlah detektor dan mengubah prinsip pengambilan datanya yaitu dari geometri berkas paralel menjadi geometri berkas kipas. Sistem TK generasi kelima merupakan perkembangan TK yang terbaru. Sistem TK generasi kelima mempunyai prinsip sumber-detektor terus bergerak rotasi (model TK spiral) sedangkan obyeknya juga bergerak.

Prinsip dasar kerja sistem tomografi komputer menyangkut distribusi serapan radiasi suatu bahan. Dengan adanya distribusi serapan radiasi oleh bahan, proses dalam sistem TK dapat menghasilkan citra internal objek. Secara garis besar teknik tomografi meliputi dua tahap proses yaitu akuisisi data dan proses rekonstruksi. Sistem akuisisi data meliputi sumber radiasi, sistem pencacah, obyek dan sistem pendeteksi. Komputer dalam tomografi sebenarnya hanya digunakan sebagai alat bantu, tetapi peranannya amat besar yaitu sebagai pengendali pengambilan data, pengolah data dan penyaji citra.

Pada penelitian ini digunakan sistem TK translasi rotasi yang tersusun dari satu sumber radiasi dan satu detektor. Sumber dan detektor bergerak translasi dan rotasi secara sinkron. Banyaknya gerakan translasi dan rotasi ditentukan oleh lebar

langkah translasi dan rotasi. Lebar langkah ini ditentukan oleh metode sampling yang digunakan. Metode sampling mensyaratkan bahwa lebar langkah translasi Δx adalah lebih kecil atau sama dengan setengah lebar kolimator atau lebar sumber radiasi w .

Atau secara matematika dapat ditulis $\Delta x \leq \frac{w}{2}$ (Wells dkk 1997).

Dari rumusan tersebut dapat disimpulkan bahwa lebar kolimator yang digunakan tidak selalu tunggal. Lebar kolimator yang digunakan dapat sembarang. Telah dibuat simulator TK generasi pertama dengan lebar kolimator $w = \Delta x$ (Sutopo, 2003). Lebar kolimator ini mempengaruhi intensitas yang mengenai objek. Intensitas yang dihasilkan kecil akan menyebabkan tingginya error distribusi. Pada penelitian ini dilakukan dengan mengambil jumlah data sesuai dengan teori sampling tetapi lebar kolimator yang digunakan tiga kali ($w = 3\Delta x$) dan lima kali ($w = 5\Delta x$) lebar langkah translasi.

Sedangkan proses rekonstruksi yang terus dikembangkan saat ini adalah metode transformasi dan metode iterasi (Mueller, 1998). Salah satu contoh metode transformasi adalah metode rekonstruksi proyeksi balik terfilter (Filtered Back Projection = FBP). Dan contoh metode iterasi adalah metode rekonstruksi iterasi aljabar (Algebraic Reconstruction Technique).

Metode rekonstruksi FBP yang paling sering digunakan dalam aplikasi pada tomografi komputer transmisi. Metode tersebut memiliki keunggulan yaitu memperoleh hasil yang akurat dan proses cepat. Namun metode ini memiliki aturan cara sampling dan jumlah data proyeksi yang sangat ketat, sehingga metode ini tidak dapat digunakan pada kondisi dimana jumlah data proyeksi kurang dari yang disyaratkan (Saucer dan Bouman, 1993 ; Censor, 1983)

Namun sebaliknya, metode rekonstruksi ART sangat fleksibel. Dalam penelitian ini proses rekonstruksinya menggunakan metode ART. Karena metode ini dapat digunakan untuk merekonstruksi citra dengan lebar kolimator berbeda dari yang umum digunakan. Selain itu menurut penelitian yang telah dilakukan Suparta dan Kusminarto pada tahun 1989 menunjukkan bahwa metode rekonstruksi ART potensial menghasilkan citra yang akurat.

1.2 Rumusan Masalah

Dari latar belakang yang telah dikemukakan maka dapat diambil rumusan masalah sebagai berikut :

1. Bagaimana membuat program simulator pengambilan data dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$?
2. Bagaimana membuat program rekonstruksi citra dengan metode ART pada data sampling dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$?
3. Bagaimana kualitas citra hasil rekonstruksi metode ART terhadap kualitas citra hasil rekonstruksi metode FBP ?
4. Adakah perbedaan proses rekonstruksi dengan metode ART pada data sampling dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$ terhadap lama proses rekonstruksi dengan metode FBP pada data sampling dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$?

1.3 Batasan penelitian

Dalam penelitian ini lebar kolimator hanya dibatasi pada ukuran $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$. Metode rekonstruksi yang digunakan hanya metode ART dan metode FBP.

1.4 Tujuan penelitian

1. Membuat program simulator pengambilan data dengan menggunakan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$.
2. Membuat program rekonstruksi citra dengan metode ART dari data sampling dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$.
3. Membedakan kualitas citra hasil rekonstruksi dengan metode ART terhadap kualitas citra hasil rekonstruksi dengan metode FBP.
4. Membedakan lama proses rekonstruksi dengan metode ART dari data sampling dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$ terhadap lama proses rekonstruksi dengan metode FBP dari data hasil sampling dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$.

1.5 Manfaat Penelitian

Hasil penelitian ini dapat digunakan sebagai :

1. Sarana studi, penelitian dan pengembangan sistem tomografi lebih lanjut, khususnya di jurusan Fisika FMIPA UNAIR.
2. Dihasilkan sistem akuisisi data dan metode rekonstruksi yang tepat pada TK generasi pertama dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$.

BAB II

TINJAUAN PUSTAKA

Teknik tomografi komputer transmisi telah mengalami kemajuan yang pesat. Teknik ini telah diaplikasikan ke berbagai bidang diantaranya dalam bidang kedokteran sebagai perangkat diagnosa dan di bidang industri (Widodo, 2000). Hal itu karena teknik tomografi komputer memberikan informasi yang detail tentang tampang lintang suatu obyek. Dalam bab ini akan dijelaskan hal-hal yang terkait dengan tomografi komputer yaitu koefisien serapan linear, prinsip sistem tomografi komputer transmisi generasi pertama, metode ART dan metode FBP.

2.1 Koefisien Serapan Linear

Di dalam sistem tomografi komputer transmisi, digunakan sinar radiasi yang dapat menembus bahan. Radiasi tersebut dapat berupa sinar X, sinar gamma, neutron atau proton. Sehingga praktis pengetahuan mengenai interaksi radiasi dengan materi sangat penting untuk dipahami karena terkait erat dengan data yang dihasilkannya. Radiasi tersebut akan berinteraksi dengan materi yang akan menyebabkan turunnya intensitas radiasi yang diteruskan, menjadi I , yang nantinya akan dideteksi oleh detektor. Menurunnya intensitas memberikan informasi akan adanya pelemahan radiasi oleh materi dinyatakan oleh koefisien serapan linear μ , atau koefisien serapan massa $\mu_M = \mu / \rho$, dengan ρ adalah kerapatan massa.

Tingkat pelemahan radiasi oleh materi bergantung pada proses penyerapan dan penghamburan foton sinar gamma oleh atom bahan yaitu melalui peristiwa

interaksi fotolistrik, interaksi Compton, dan produksi berpasangan. Bila ketiga peristiwa tersebut diasumsikan saling bebas, maka koefisien serapan linear μ merupakan penjumlahan dari koefisien ketiganya proses tersebut, yaitu :

$$\mu_{total} = \mu_{ph} + \mu_{Compton} + \mu_{pp} \quad (2.1)$$

dengan μ_{ph} adalah koefisien atenuasi linear fotolistrik, μ_c adalah koefisien atenuasi Compton, μ_{pp} adalah koefisien atenuasi linear produksi berpasangan.

Bila sinar gamma berintensitas I melewati suatu materi penyerap dengan ketebalan dx , maka intensitas I akan mengalami pelemahan radiasi sebesar dI , yang dapat dinyatakan dengan persamaan berikut :

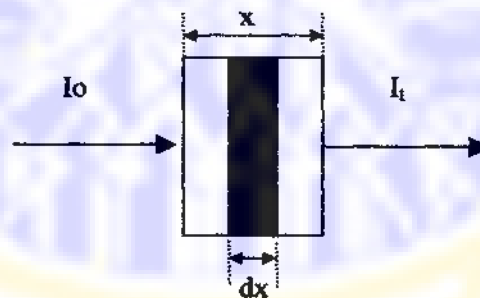
$$\frac{dI}{I} = -\mu dx \quad (2.2)$$

dengan μ adalah koefisien serapanan linear dan tanda negatif (-) menunjukkan bahwa intensitas menurun dengan meningkatnya ketebalan bahan (Agarwall,1989).

Bila tebal suatu materi total adalah x , intensitas sebelum menembus materi adalah I_0 dan bila dianggap bahwa materinya homogen maka setelah melewati materi adalah :

$$I = I_0 e^{-\mu x} \quad (2.3)$$

Hal ini dapat diperlihatkan dengan gambar 2.1 di bawah ini :



Gambar 2.1 Proses pelemahan intensitas sinar gamma oleh bahan dengan ketebalan x

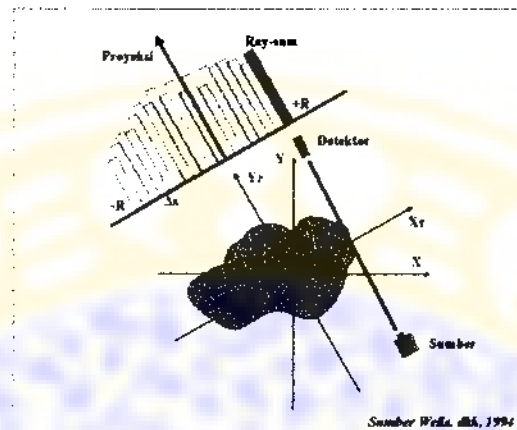
2.2 Prinsip Sistem Tomografi Komputer Transmisi Generasi Pertama

Tomografi komputer generasi pertama merupakan perangkat TK-Scanner yang paling sederhana (Wells, dkk, 1997). TK generasi pertama disebut juga TK translasi dan rotasi karena hanya memanfaatkan satu sumber dan satu detektor yang digerakkan secara translasi dan rotasi, untuk mendapatkan informasi data yang lengkap. Namun sistem ini tidak cocok jika digunakan dalam bidang kedokteran karena prosesnya membutuhkan waktu yang lama, walaupun citra yang dihasilkan akurat.

Setiap satu posisi sumber-detektor akan didapatkan satu data intensitas yang tercacah oleh detektor yang didefinisikan sebagai satu ray-sum. Besaran ray-sum ini dapat dituliskan sebagai

$$p_{\phi}(x_r) = \ln(I_0 / I_1) = \int_{L_{\phi}(x_r)} \mu(x, y) ds \quad (2.4)$$

dengan $\mu(x, y)$ adalah nilai koefisien serapan linear bahan pada koordinat (x, y) sepanjang garis lintasan berkas radiasi $L_{\phi}(x_r)$. Kemudian posisi sumber dan detektor digerakkan paralel (langkah translasi) sepanjang sumbu x sampai akhir daerah pengambilan data (pemayaran, scanning), maka di dapat sekumpulan ray-sum yang disebut satu proyeksi. Setelah didapatkan dengan lengkap sekumpulan ray-sum untuk satu proyeksi, maka posisi obyek di putar sejauh $\Delta\phi$ tegak lurus terhadap posisi sumber dan detektor (langkah rotasi). Obyek diputar dengan pengambilan data hanya pada range $0 \leq \phi \leq \pi$, karena range $\pi \leq \phi \leq 2\pi$ memiliki kesesuaian dengan $0 \leq \phi \leq \pi$. Prinsip pengambilan data pada sistem tomografi komputer generasi pertama ditunjukkan pada gambar 2.2.



Gambar 2.2 Proses pemayaran pada sistem tomografi komputer generasi pertama

2.3 Metode Sampling

Pemayaran tunggal pada suatu objek menghasilkan sebuah ray-sum. Berdasarkan persamaan (2.4), ray-sum berkas ke- j dapat dinotasikan dengan p_j , yang merupakan integral garis dari fungsi koefisien pelemahan linier (μ_i) sepanjang berkas ke- j . (Ain dan Ukhrowiyah 2004). Sehingga dapat ditulis,

$$p_j \approx \sum_{i=1}^N w_{ij} \mu_i \quad \text{dengan } j = 1, 2, 3, \dots, M \quad (2.5)$$

dengan $j = (\phi, x_r)$ dan $i = (x, y)$, sedangkan w_{ij} adalah nilai bobot luasan piksel ke- i yang terpapar berkas ray-sum ke- j .

Sekumpulan raysum dalam 1 proyeksi didapatkan dengan melakukan pemayaran pada obyek secara translasi dengan interval Δx tegak lurus posisi sumber detektor. Sebuah proyeksi didapatkan dari sejumlah N raysum dengan interval Δx , sepanjang arah x , dan sejumlah M proyeksi didapatkan dari gerakan rotasi dengan interval rotasi $\Delta\phi$. Masing-masing dapat dirumuskan sebagai :

$$N = \frac{2R}{\Delta x} \quad (2.6)$$

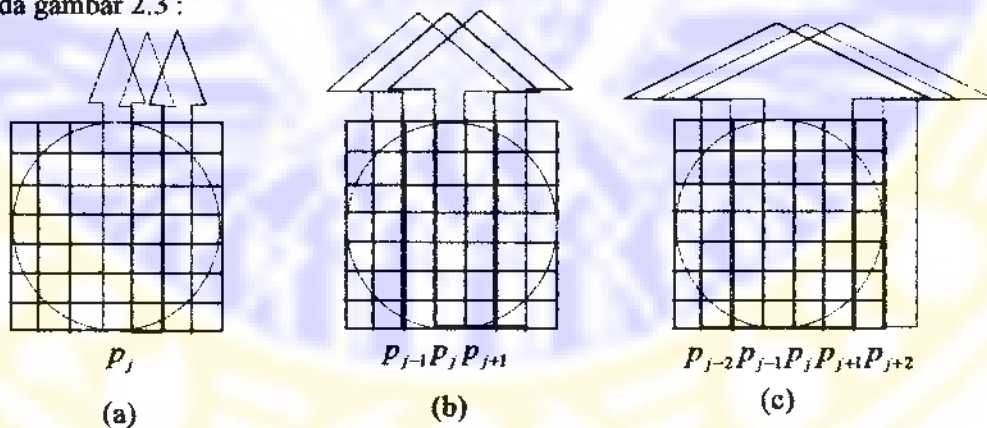
$$M = \frac{\pi}{\Delta \phi} \quad (2.7)$$

Untuk memperoleh pola sinogram yang paling mendekati obyek yang ditransformasikan maka pada persamaan 2.6 dan 2.7, $\Delta \phi$ dan Δx tidak dapat dipilih sembarang (Suparta, dkk, 1999) karena resolusi citra yang dihasilkan salah satunya tergantung pada langkah translasi dan rotasi. Batasan langkah translasi adalah :

$$\Delta x \leq \frac{w}{2} \quad (2.8)$$

dengan w adalah lebar sinar (didefinisikan oleh ukuran lebar kolimator pada sumber dan detektor).

Pada proses pengambilan data idealnya kontinyu dan makin banyak data proyeksi yang didapatkan, maka citra yang dihasilkan semakin baik. Dalam penelitian ini digunakan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$. Model proses pengambilan data dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$ ditunjukkan pada gambar 2.3 :



Gambar 2.3 Sampling Data (a) $w = \Delta x$ (b) $w = 3\Delta x$ (c) $w = 5\Delta x$

Ray-sum dari sampling data menggunakan lebar kolimator $w = 3\Delta x$, dapat dihitung dengan menjumlah p_{j-1} , p_{j+1} , dan p_j sehingga persamaan (2.5) menjadi :

$$p_j \approx \sum_{i=1}^N (w_{ij}\mu_i + w_{i,j-1}\mu_{i-1} + w_{i,j+1}\mu_{i+1}) \quad \text{dengan } j = 1,2,3,\dots,M \quad (2.9)$$

Sedangkan untuk menghitung ray-sum dengan lebar kolimator $w = 5\Delta x$ menggunakan persamaan :

$$p_j \approx \sum_{i=1}^N (w_{ij}\mu_i + w_{i,j-2}\mu_{i-2} + w_{i,j+2}\mu_{i+2} + w_{i,j-1}\mu_{i-1} + w_{i,j+1}\mu_{i+1})$$

dengan $j = 1, 2, 3 \dots M$ (2.10)

2.4 Metode Rekonstruksi

Sekumpulan proyeksi yang didapat selanjutnya akan direkonstruksi untuk menghasilkan sebuah citra. Pada prinsipnya proses rekonstruksi citra merupakan sebuah proses inversi (proyeksi balik) dari ruang (sinogram) ke ruang kartesian (citra). Secara garis besar metode rekonstruksi terbagi tiga bagian yaitu metode proyeksi balik, metode iterasi, dan metode analitik (Widodo, 2000).

2.4.1 Metode Rekonstruksi ART

Salah satu metode rekonstruksi yang paling representatif adalah teknik rekonstruksi aljabar (*Algebraic Reconstruction Technique = ART*). Meskipun membutuhkan waktu lama dan memori komputer yang besar, tetapi metode ini menghasilkan citra yang akurat. Metode ini pertama kali diimplementasikan oleh Gordon, Bender dan Herman pada tahun 1970 untuk menyelesaikan permasalahan rekonstruksi tiga dimensi dalam bidang mikroskopi elektron dan *radiology* (Gordon, 1974).

Proses rekonstruksi iteratif dimulai dengan menggunakan nilai sembarang pada citra rekonstruksi sebagai dugaan awal. Citra dugaan awal tersebut digunakan untuk menghitung nilai-nilai proyeksi semu untuk dibandingkan dengan proyeksi terukur. Jika ada perbedaan, maka suatu faktor koreksi untuk mengoreksi citra dugaan awal tersebut ditentukan berdasarkan selisih proyeksi lainnya terukur dan proyeksi semu tersebut. Langkah tersebut diulangi untuk proyeksi lainnya. Apabila seluruh data proyeksi telah digunakan, maka satu kali iterasi tersebut terus diulang hingga tercapai ketelitian tertentu (Censor, 1983). Metode rekonstruksi ART untuk lebar kolimator $w = \Delta x$ dapat ditunjukkan oleh persamaan (Gordon, 1974) :

$$f_i^{k+1} = f_i^k + \frac{w_{ij}}{\sum w_{ij}} \frac{1}{2} (p_j - q_j) \quad (2.11)$$

Sedangkan metode rekonstruksi ART dengan lebar kolimator $w = 3\Delta x$ dapat ditulis :

$$f_i^{k+1} = f_i^k + \frac{w_{ij} + w_{ij-1} + w_{ij+1}}{\sum (w_{ij-1}^2 + w_{ij}^2 + w_{ij+1}^2)} ((p_{j-1} + p_j + p_{j+1}) - (q_{j-1} + q_j + q_{j+1})) \quad (2.12)$$

Dan untuk lebar kolimator $w = 5\Delta x$ persamaan rekonstruksi ART dapat ditulis:

$$f_i^{k+1} = f_i^k + \frac{w_{ij} + w_{ij-2} + w_{ij+2} + w_{ij-1} + w_{ij+1}}{\sum (w_{ij-2}^2 + w_{ij-1}^2 + w_{ij}^2 + w_{ij+1}^2 + w_{ij+2}^2)} ((p_{j-2} + p_{j-1} + p_j + p_{j+1} + p_{j+2}) - (q_{j-2} + q_{j-1} + q_j + q_{j+1} + q_{j+2})) \quad (2.13)$$

Dimana :

$$f_i^{k+1} = \text{citra setelah iterasi ke- } k+1$$

$$f_i^k = \text{citra sebelumnya (iterasi ke- } k)$$

w_y = bobot yang menunjukkan luasan piksel tertentu yang melewati berkas (nilai antara 0 dan 1)

p_j = raysum terukur yang diperoleh dari simulasi

q_j = raysum semu

i = index piksel $i = 1, \dots, NxN$

j = index ray-sum $j = 1, \dots, NxM$

Untuk menunjukkan kondisi selesainya iterasi maka diperlukan kriteria penghenti. Kriteria penghenti tersebut adalah *discrepancy*. *Discrepancy* menunjukkan besar beda bilangan antara data ray-sum terukur dan ray-sum terhitung (Raparia, 1998). Persamaan tersebut dapat ditulis sebagai berikut,

$$D^k = \left\{ \frac{1}{NxM} \sum_{j=1}^{NxM} \frac{(p_j - q_j^k)^2}{\sum w_y^2} \right\}^{\frac{1}{2}} \quad (2.14)$$

2.4.2 Metode Rekonstruksi FBP

Salah satu metode rekonstruksi adalah metode proyeksi balik penjumlahan proyeksi tersaring secara konvolusi (Suparta, 1999). Proses rekonstruksi FBP terdiri atas dua bagian proses, yaitu proses proyeksi balik (inversi) dari ruang sinogram (Radon) ke ruang citra (kartesian) dan proses konvolusi oleh filter terhadap proyeksi, yang dilangsungkan sebelumnya. Suparta (1999) menyatakan secara analitis proses FBP dapat dituliskan sebagai :

$$\mu(x, y) = \int_0^\pi p(\phi, x, y) d\phi \quad (2.15)$$

dengan

$$p'(x_r, \phi) = \int p(x_r', \phi) h(x_r - x_r') dx_r' = p(x_r', \phi) * h(x_r - x_r') \quad (2.16)$$

dengan $h(x_r)$ adalah filter pengkonvolusi. Bentuk diskrit komputasional persamaan tersebut dapat dituliskan sebagai :

$$\mu[i, j] = \Delta\phi \sum_{m=0}^{m=M-1} p'[i \cos(m\Delta\alpha\phi) + j \sin(m\Delta\phi), m] \quad (2.17)$$

$$p'[n, m] = \Delta x_r \sum_{n'=-N}^{n'+N} p[n', m] h[n - n'] \quad (2.18)$$

Indeks piksel $[i, j]$ menunjukkan posisi piksel pada daerah citra rekonstruksi dengan variasi $-\frac{1}{2}N < i, j < +\frac{1}{2}N$. Indeks ray-sum $[n, m]$ menunjukkan indeks ray-sum ke-

n dari proyeksi ke- m , dengan n bervariasi $-\frac{1}{2}N < n < +\frac{1}{2}N$ dan m bervariasi

$0 < m, M - 1$

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di laboratorium Radiasi, Fisika FMIPA UNAIR, Surabaya. Waktu pelaksanaan penelitian dilakukan selama 12 bulan yaitu mulai Juli 2004 - Juli 2005.

3.2 Alat dan Bahan Penelitian

Penelitian ini dilakukan secara simulasi. Peralatan yang digunakan adalah seperangkat komputer dengan spesifikasi hardware Pentium III dengan RAM 128 MB. Dan software yang digunakan adalah CT Imager, Microsoft Excel, dan Image Comparator. Bahasa pemrograman yang digunakan C++ Builder 4.0.

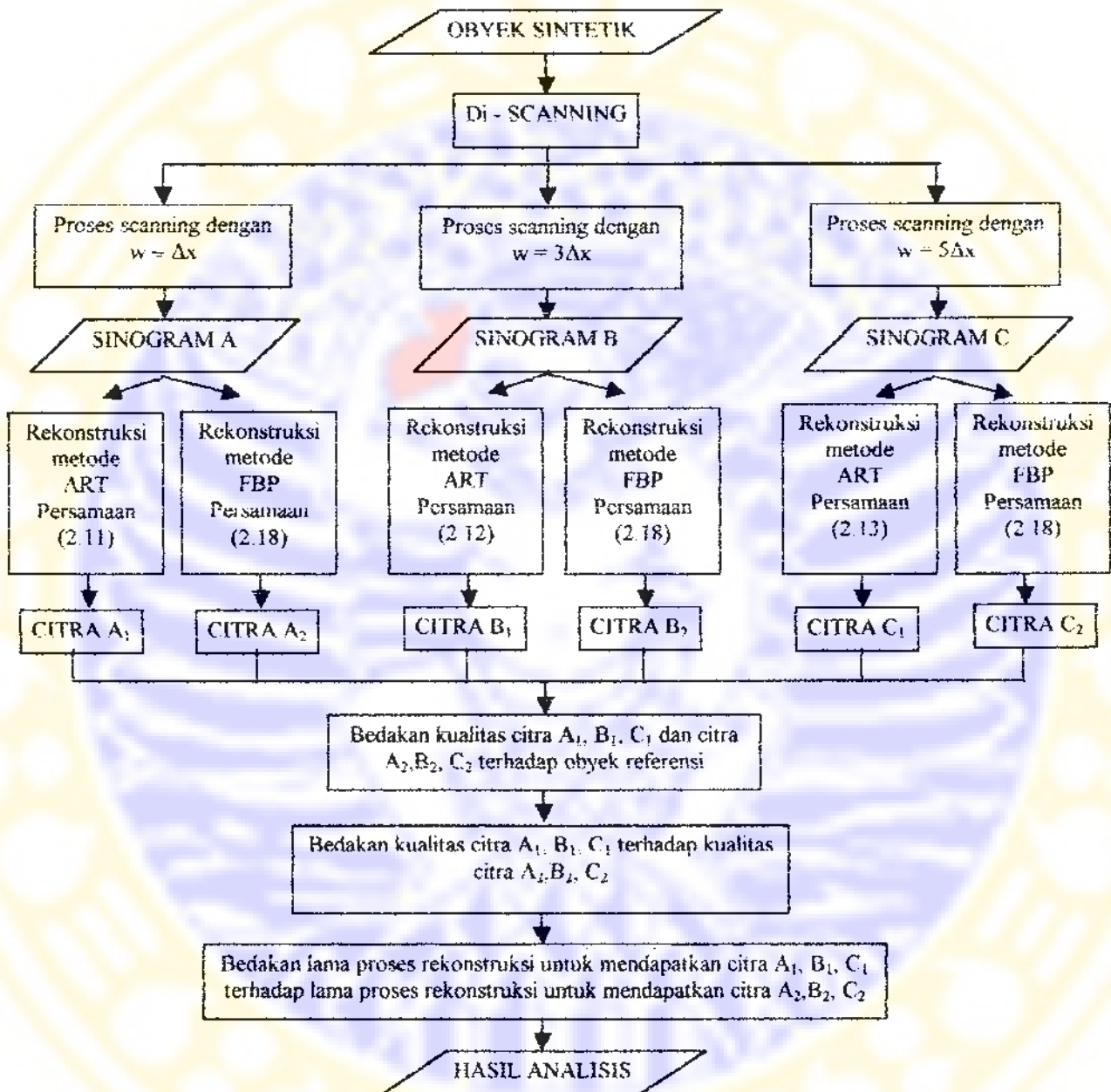
Bahan yang digunakan dalam penelitian ini adalah dua objek sintetik yang telah dibuat peneliti sebelumnya menggunakan Microsoft Excel, yaitu Modul.dat (Suparta,1999) dan Segitiga.dat (Nugroho,2002). Objek tersebut berupa data numerik yang dibuat dengan menggunakan program Microsoft Excel. Objek segitiga mempunyai koefisien pelemahan seragam dan objek modul mempunyai koefisien pelemahan tidak seragam. Kedua objek tersebut berada dalam sebuah lingkaran berdiameter 127 piksel, dengan nama masing-masing file Segitiga.dat dan Modul.dat.



Gambar 3.1 Objek sintetik : (a) Segitiga.dat (b)Modul.dat

3.3 Prosedur Penelitian

Penelitian ini dilakukan secara simulasi. Prosedur simulasi dalam penelitian ini digambarkan pada blok diagram gambar (3.2) sebagai berikut:



Gambar 3.2 Blok Diagram proses simulasi

Prosedur simulasi dimulai dari objek sintetik yang di scan dengan menggunakan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$. Proses scanning

dilakukan secara simulasi yang dibuat pada penelitian. Hasil dari proses scanning berupa sinogram. Kemudian sinogram direkonstruksi menggunakan metode ART yang dibuat dalam penelitian ini dan metode FBP yang tersedia di Laboratorium Radiasi UNAIR untuk menghasilkan citra rekonstruksi.

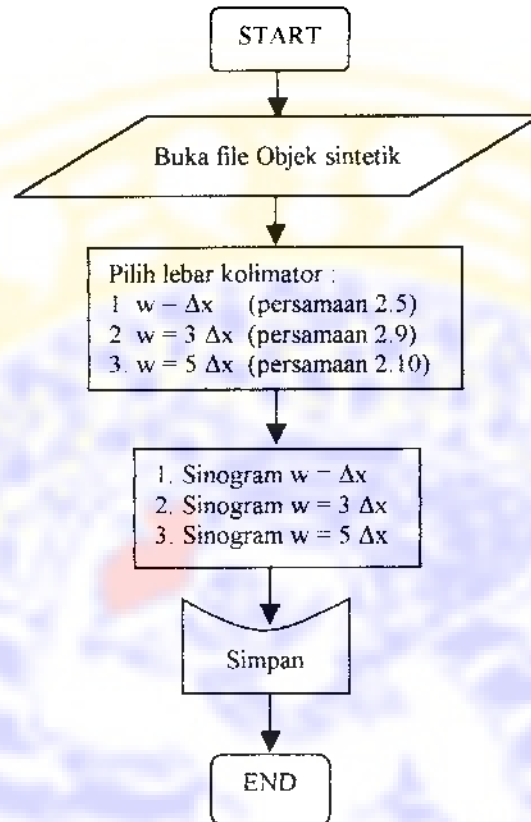
Analisis program rekonstruksi dengan metode ART yang dibuat dalam penelitian ini, menggunakan program CT Image Comparator yang telah dibuat oleh peneliti sebelumnya (Suparta, 1999). Kemudian citra-citra hasil rekonstruksi tersebut dianalisis kualitasnya.

3.3.1 Proses Scanning

Objek sintetik di-scan untuk menghasilkan sinogram. Proses scanning dilakukan secara simulasi menggunakan simulator yang dibuat dengan tiga lebar kolimator yang berbeda yaitu $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$. Adapun algoritma program simulasi proses scanning adalah sebagai berikut,

1. Ambil data objek sintetik (*dat).
2. Pilih lebar kolimator $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$.
3. Hitung ray-sum untuk $w = \Delta x$ dihitung dengan persamaan 2.9, untuk $w = 3\Delta x$ dihitung dengan persamaan 2.10, dan untuk $w = 5\Delta x$ dihitung dengan persamaan 2.11.
4. Simpan hasil sinogram (*att)

Adapun *flow chart* program simulator proses *scanning* dapat dilihat pada gambar 3.3.



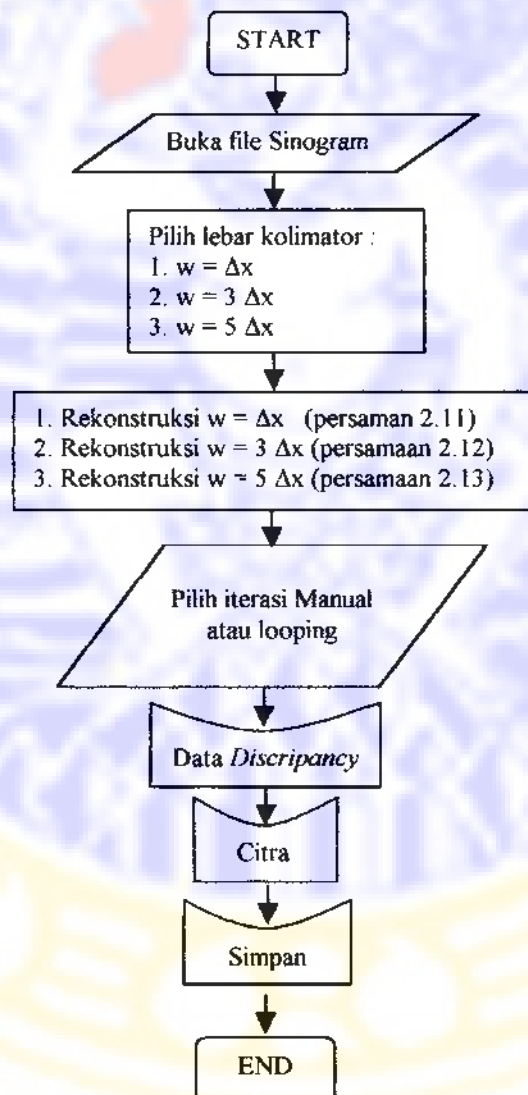
Gambar 3.3 *Flow Chart* Simulator Proses *Scanning* TK Generasi Pertama

3.3.2 Proses Rekonstruksi

Sinogram hasil dari proses *scanning* dengan menggunakan simulator proses *scanning*, direkonstruksi untuk menghasilkan citra. Proses rekonstruksinya menggunakan metode ART yang dibuat dalam penelitian ini dan metode FBP. Untuk rekonstruksi metode FBP, sebelum direkonstruksi sinogramnya dibagi jumlah ray-sumnya N . Rekonstruksi FBP sudah diimplementasikan dalam program CT Imager 1.0 yang sudah tersedia dibuat peneliti sebelumnya (Suparta 1999). Adapun algoritma program rekonstruksi dengan metode ART yang dibuat dalam penelitian ini adalah :

1. Ambil data sinogram (*att)
2. Hitung w_{ij}
5. Pilih lebar kolimator $w = \Delta x$, $w = 3 \Delta x$ dan $w = 5 \Delta x$.
6. Hitung proses ART
7. Pilih iterasi untuk rekonstruksi sinogram
8. Atau pilih menghitung *discrepancy*
9. Simpan hasil citra (*dat) dan (*fpi)

Sedang *flow chart* program rekonstruksinya ditunjukkan pada gambar :



Gambar 3.4 *Flow Chart* Program Rekonstruksi Citra Menggunakan Metode ART

3.4 Analisis Data

Citra hasil rekonstruksi dianalisis kualitasnya. Analisis citra dilakukan dengan cara membedakan citra hasil rekonstruksi terhadap objek referensi. Perbedaan citra-citra hasil rekonstruksi tersebut dilakukan secara visual dan secara numerik. Perbedaan secara visual dilakukan dengan melihat citra hasil rekonstruksi dan representasi profil garis horizontalnya. Perbedaan secara numerik dilakukan dengan menghitung *root mean square difference (rmsd)* yang mengukur kesamaan distribusi μ antar citra yang mengukur perbedaan terbesar nilai-nilai koefisien atenuasi linier antar citra. Perumusan *rmsd* dan secara matematis dapat dinyatakan sebagai berikut :

$$rmsd = \frac{1}{\mu_{max}} \left[\sqrt{\frac{\sum_i \sum_j (\mu_{cit}[i, j] - \mu_{ref}[i, j])^2}{N'}} \right] \times 100\% \quad (3.1)$$

dengan :

$\mu_{cit}[i, j]$ = piksel citra pada koordinat $[i, j]$

$\mu_{ref}[i, j]$ = objek referensi pada koordinat $[i, j]$

i = index piksel $i = N \times N$

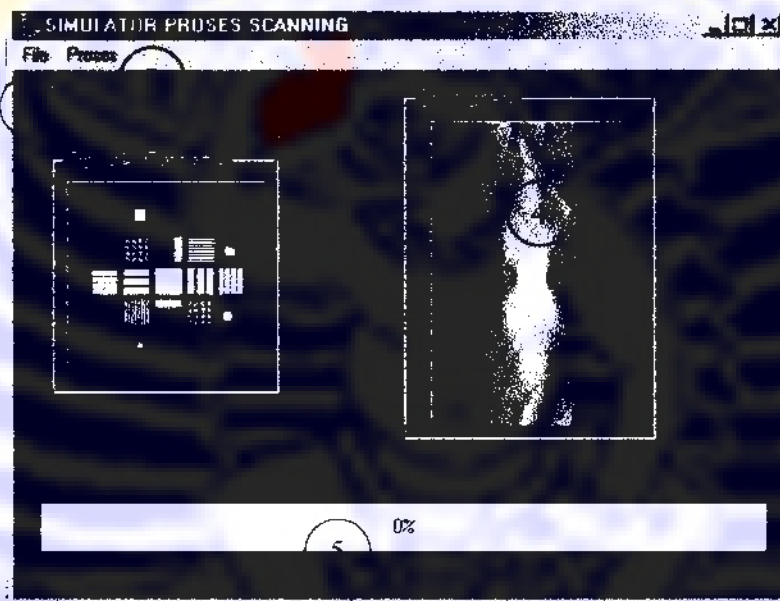
j = index ray-sum $j = N \times M$

Dan koordinat dibatasi sampai N' piksel yang terletak didalam obyek atau lingkaran citra (Herman, 1980; Suparta, 1999). Kedua persamaan tersebut telah diimplementasikan dalam bentuk paket program (Nugroho, 2002), dan dipakai dalam penelitian ini.

BAB IV HASIL DAN PEMBAHASAN

4.1 Tampilan Simulator Proses *Scanning* TK Generasi Pertama

Berdasarkan algoritma dan *flow chart* proses *scanning* pada gambar 3.3, maka dihasilkan simulator proses *scanning* dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$. Tampilan simulator proses *scanning* ditunjukkan pada gambar 4.1 dan *list* programnya ditunjukkan pada lampiran A :

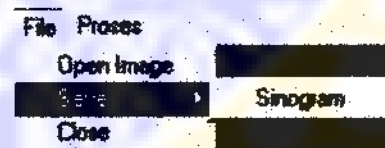


Gambar 4.1 Tampilan simulator proses *scanning*

Keterangan :

1. Menu file dengan pilihan :

- i. Membuka file objek sintetik.
- ii. Menyimpan hasil sinogram.
- iii. Menutup simulator.



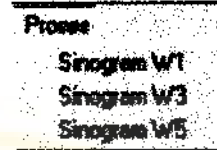
Gambar 4.2 Menu file

2. Menu proses dengan pilihan :

- i. *Scanning* dengan $w = \Delta x$

ii. *Scanning* dengan $w = 3\Delta x$

iii. *Scanning* dengan $w = 5\Delta x$

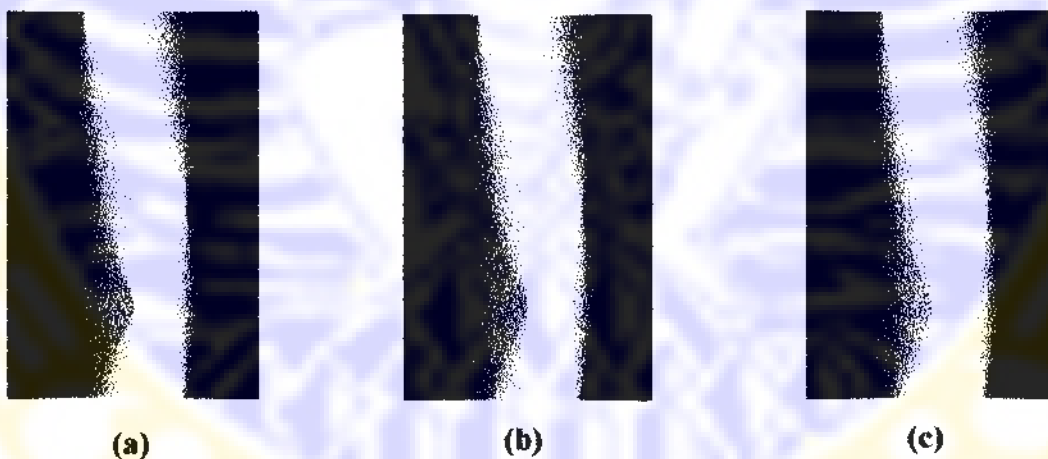


Gambar 4.3 Menu Proses

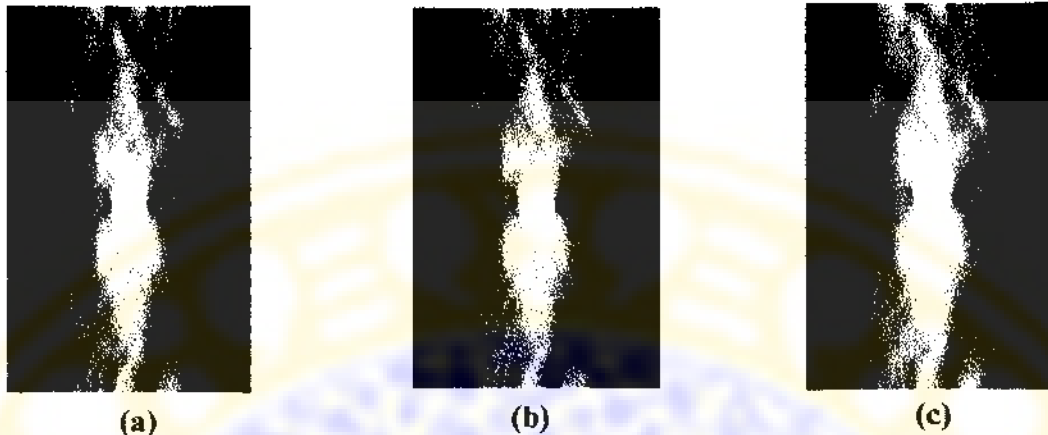
3. Objek sintetis yang di-*scan*.
4. Sinogram hasil dari proses *scanning*.
5. Prosentase kemajuan proses simulasi.

4.2 Sinogram Hasil Proses *Scanning*

Simulator proses *scanning* yang dibuat dalam penelitian ini dapat digunakan untuk men-*scanning* objek sintetis. Proses *scanning* dengan simulator ini dilakukan dengan tiga lebar kolimator berbeda, yaitu dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$. Proses *scanning* dilakukan pada objek Segitiga.dat dan Modul.dat. Sinogram hasil proses *scanning* tersebut ditunjukkan pada gambar 4.4 dan gambar 4.5 :



Gambar 4.4 Sinogram dari objek Segitiga.dat (a) Sinogram $w = \Delta x$
(b) Sinogram $w = 3\Delta x$ (c) Sinogram $w = 5\Delta x$.



(a) Sinogram $w = \Delta x$
 (b) Sinogram $w = 3\Delta x$ (c) Sinogram $w = 5\Delta x$.

4.3 Tampilan Program Rekonstruksi Citra Metode Iterasi Aljabar

Algoritma dan *flow chart* gambar 3.4, diterjemahkan ke dalam bahasa borland C++ Builder sehingga dihasilkan simulator program rekonstruksi citra. Tampilan simulator program rekonstruksi ditunjukkan pada gambar 4.6 dan *list* programnya ditunjukkan pada lampiran B. Simulator ini digunakan untuk merekonstruksi sinogram $w = \Delta x$, $w = 3\Delta x$ dan $w = 5\Delta x$.

Iterasi	Discrepancy	Euclidean Distance
1	2,63743631192323	11,4485653554146
2	1,55851107390289	7,44727941425198
3	1,41316471297052	6,67684935322301
4	1,28177599641004	6,01247360074683
5	1,15037460150297	5,38612570830837
6	1,06509817550728	4,82876342958656
7	0,953609414007858	4,29017447910853
8	0,86095759869877	3,8044436367336
9	0,781591580063098	3,35636242349324
10	0,689934374272807	2,94219568316434
11	0,624016496567716	2,58184994614294

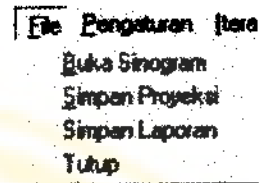
Progress bar: 40%

Gambar 4.6 Simulator Program Rekonstruksi Citra dengan Metode ART

Keterangan :

1. Menu File dengan pilihan :

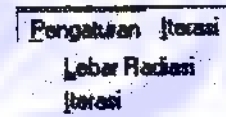
- i. Membuka data sinogram (*.att).
- ii. Menyimpan citra rekonstruksi.
- iii. Menyimpan laporan nilai discrepancy.
- iv. Menutup simulator.



Gambar 4.7 Menu File

2. Menu Pengaturan dengan pilihan :

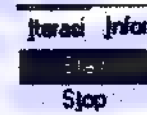
- i. Memilih lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$.
- ii. Memilih proses iterasi :
 - a. Iterasi secara manual.
 - b. Iterasi sampai 50 kali.
 - c. Iterasi dengan input nilai discrepancy.



Gambar 4.8 Menu Pengaturan

3. Menu Iterasi :

- i. Memulai proses iterasi.
- ii. Menghentikan proses iterasi.



Gambar 4.9 Menu Iterasi

4. Sinogram hasil proses *scanning*.

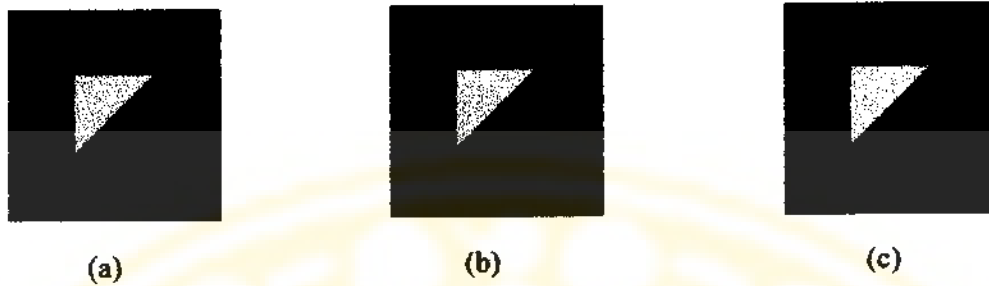
5. Citra hasil rekonstruksi.

6. Laporan nilai *discrepancy*.

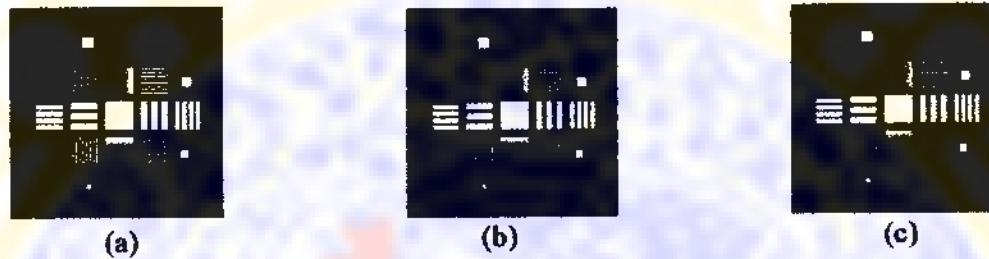
7. Prosentasi kemajuan proses simulasi rekonstruksi.

4.4 Citra Hasil Rekonstruksi

Simulator program rekonstruksi citra metode ART dapat dipergunakan untuk merekonstruksi sinogram pada gambar 4.4 dan 4.5 sehingga dihasilkan citra. Citra-citra hasil rekonstruksi dengan metode ART tersebut ditunjukkan pada gambar 4.10 dan 4.11.

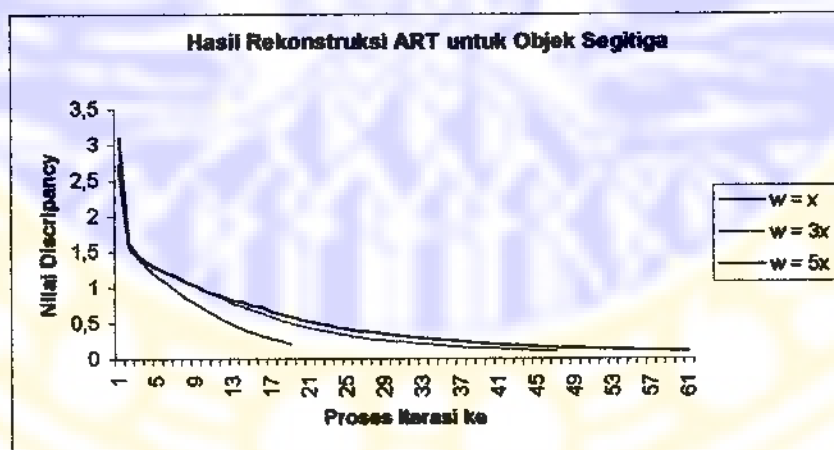


Gambar 4.10 Citra Segitiga hasil rekonstruksi ART (a) $w = \Delta x$
(b) $w = 3\Delta x$ (c) $w = 5\Delta x$

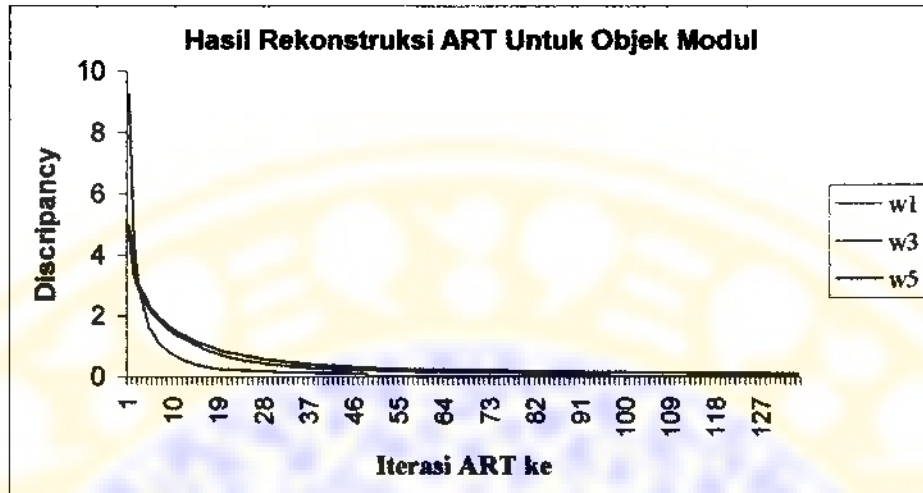


Gambar 4.11 Citra Modul hasil rekonstruksi ART (a) $w = \Delta x$
(b) $w = 3\Delta x$ (c) $w = 5\Delta x$

Pada program rekonstruksi metode ART yang dibuat dalam penelitian ini, kriteria penghenti proses iterasi dilakukan sampai nilai *discrepancy* 10^{-1} . Pengambilan nilai *discrepancy* 10^{-1} , diambil berdasarkan pada nilai *discrepancy* tersebut nilai *rmsd*-nya sudah kecil. Hal itu ditunjukkan pada lampiran C. Hasil-hasil nilai *discrepancy* dan proses iterasi dapat dibuat grafiknya dan diperlihatkan pada gambar 4.12 dan 4.13.

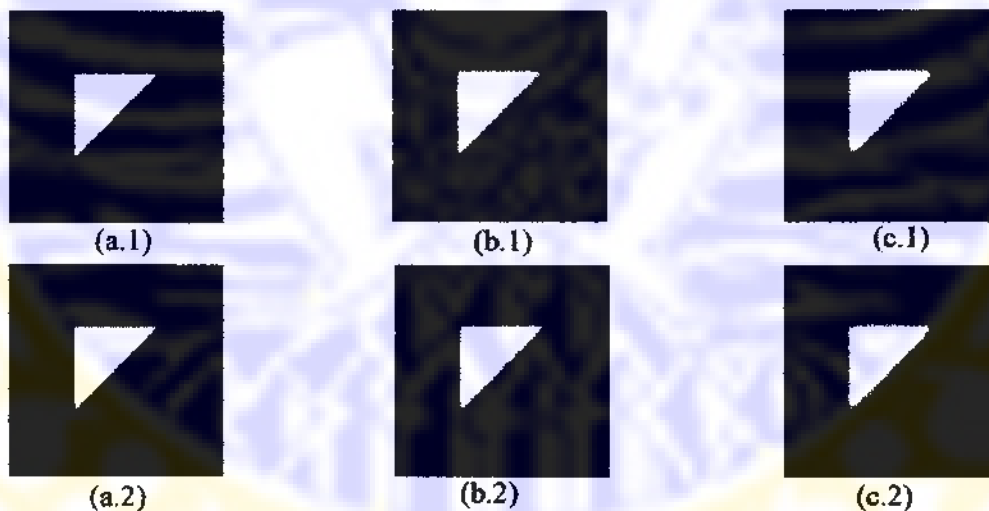


Gambar 4.12 Hubungan antara nilai *discrepancy* dan proses iterasi pada Objek Segitiga.dat



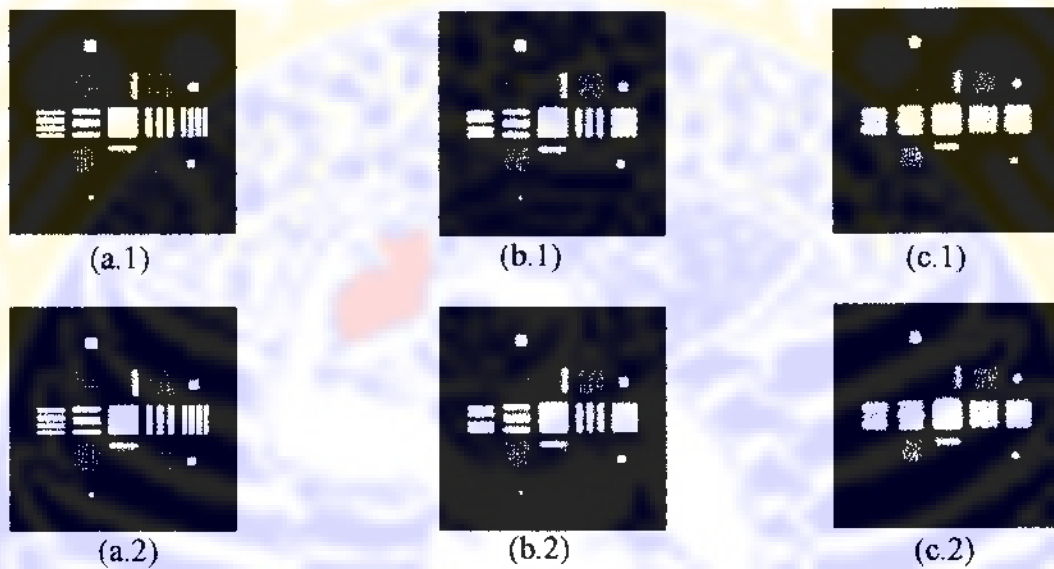
Gambar 4.13 Hubungan antara nilai discrepancy dan proses iterasi pada objek Modul. dat

Untuk mengetahui kualitas citra hasil rekonstruksi dengan metode ART dari program rekonstruksinya yang dibuat pada penelitian ini, maka untuk itu citra-citra hasil rekonstruksi dibandingkan dengan objek referensi dan citra-citra hasil rekonstruksi dengan metode FBP. Untuk itu, sinogram-sinogram gambar 4.4 dan 4.5 direkonstruksi dengan metode FBP menggunakan program CT Imager1.0.



Gambar 4.14 Citra segitiga hasil rekonstruksi metode FBP (a.1 & a.2) $w = \Delta x$, (b.1 & b.2) $w = 3\Delta x$ dan (c.1 & c.2) $w = 5\Delta x$.

Pada metode FBP sinogram hasil proses scanning harus dibagi wN terlebih dahulu, karena jika tidak citra yang dihasilkan secara numerik tidak sesuai objek referensi. Citra-citra hasil rekonstruksi dengan metode FBP tersebut ditunjukkan pada gambar 4.14 dan gambar 4.15.

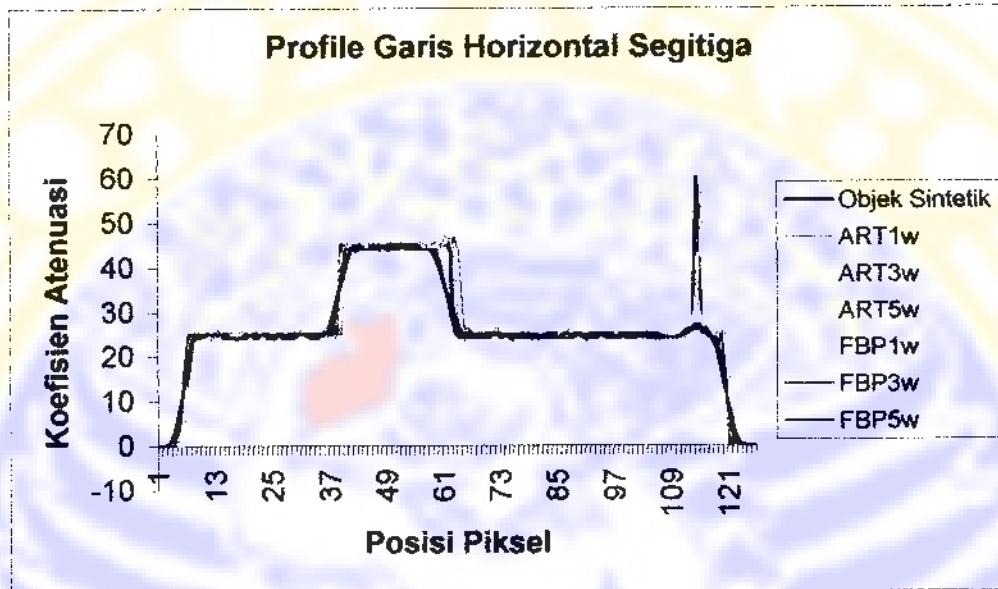


Gambar 4.15 Citra Modul hasil rekonstruksi metode FBP (a.1 & a.2) $w = \Delta x$, (b.1 & b.2) $w = 3\Delta x$ dan (c.1 & c.2) $w = 5\Delta x$.

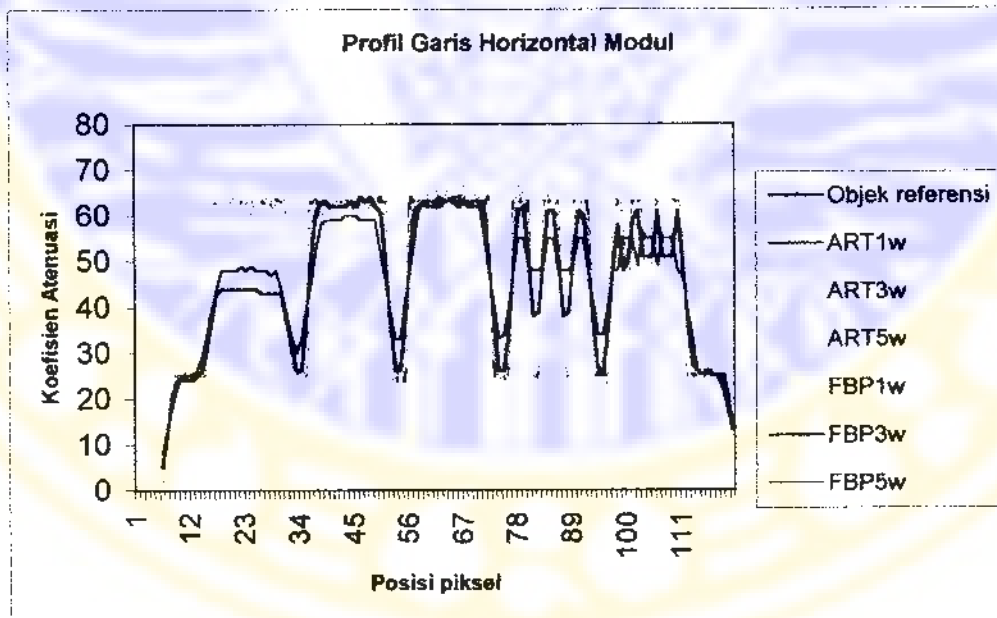
4.5 Hasil analisis kualitas citra

Citra-citra hasil rekonstruksi dengan metode ART dianalisis kualitasnya. Analisis dilakukan dengan cara membedakan kualitas citra hasil rekonstruksi dengan ART terhadap objek referensi dan citra-citra hasil rekonstruksi dengan metode FBP. Pembedaan dilakukan secara visual dan secara numerik. Pembedaan secara visual dengan melihat kualitas citra-citra hasil rekonstruksi pada gambar 4.10, 4.11, 4.14, 4.15 dan profil garis horisontal.

Profil garis horisontal antara citra objek referensi, citra hasil rekonstruksi ART dengan lebar kolimator yang berbeda ditunjukkan pada gambar 4.16 dan gambar 4.17.

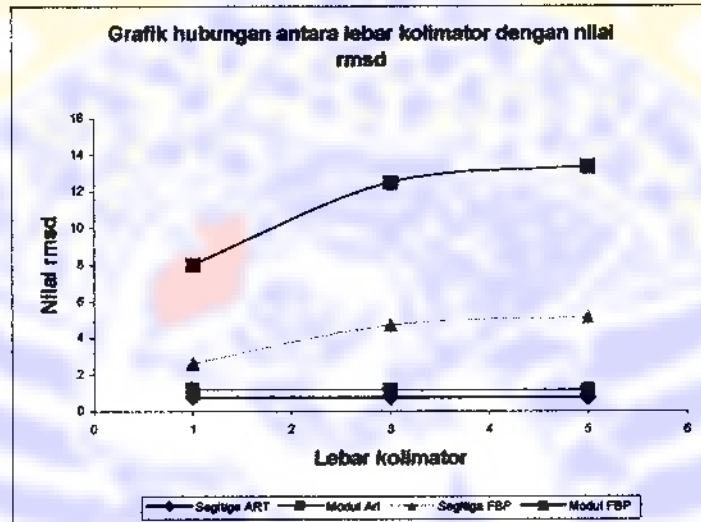


Gambar 4.16 Profil garis horisontal dari objek Segitiga



Gambar 4.17 Profil garis horisontal dari objek Modul

Pembedaan secara numerik dilakukan dengan melihat nilai *rmsd* antara citra-citra hasil rekonstruksi metode ART terhadap objek referensi dan citra-citra hasil rekonstruksi dengan metode FBP. Hasil-hasil nilai *rmsd* tersebut diperlihatkan pada lampiran D. Grafik hubungan antara nilai *rmsd* terhadap lebar kolimator ditunjukkan pada gambar 4.18.



Gambar 4.18 Grafik hubungan antara lebar kolimator dengan nilai *rmsd*

Untuk melihat keunggulan metode rekonstruksi dengan metode ART, selain dengan melihat kualitas citra yang dihasilkan juga melihat lama waktu yang diperlukan untuk proses rekonstruksi. Lama proses rekonstruksi dihitung dengan stop watch. Lama proses rekonstruksi dengan metode ART untuk setiap lebar kolimator adalah berbeda. Sedangkan lama proses rekonstruksi dengan metode FBP adalah sama, baik untuk objek dan lebar kolimator yang berbeda. Lama waktu proses rekonstruksi dengan metode ART dan metode FBP ditunjukkan pada tabel 4.1.

Tabel 4.1 Tabel lama proses rekonstruksi

No.	Objek Sintetik	Lebar kolimator	Lama proses (detik)
1.	Segitiga ART	1	228
		3	1225
		5	2763
2.	Modul ART	1	1349
		3	7296
		5	16353
3.	Segitiga dan Modul FBP	1,3, dan 5	16

4.6 Pembahasan

Program rekonstruksi metode ART dengan menggunakan lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$ tersebut diuji dengan melihat kualitas citra yang dihasilkan. Program rekonstruksi dengan metode ART yang dibuat pada penellitian ini digunakan untuk merekonstruksi sinogram dengan tiga lebar kolimator yang berbeda yaitu $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$.

Kualitas citra hasil rekonstruksi ART dapat dilihat pada gambar 4.10 dan gambar 4.11. Sedangkan untuk citra rekonstruksi metode FBP dapat dilihat kualitasnya pada gambar 4.18 dan gambar 4.19.

Secara visual kualitas citra hasil rekonstruksi dengan metode ART dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$ mendekati objek referensi. Sedangkan kualitas citra hasil rekonstruksi dengan metode FBP dengan lebar kolimator $w = \Delta x$ lebih mendekati objek referensi jika dibandingkan dengan lebar kolimator $w = 3\Delta x$, dan $w = 5\Delta x$. Hal ini ditunjukkan oleh citra dengan

lebar kolimator $w = 3\Delta x$, dan $w = 5\Delta x$ lebih buram dari objek referensi. Hal tersebut juga ditunjukkan pada profil garis horisontal (gambar 4.16 dan 4.17).

Pada profil garis horisontal citra-citra hasil rekonstruksi metode ART, menunjukkan bahwa kualitas citra-citra yang dihasilkan mendekati objek referensi. Sedangkan pada profil garis horisontal citra-citra hasil rekonstruksi metode FBP menunjukkan bahwa kualitas citra hasil rekonstruksi yang mendekati objek referensi hanyalah citra dengan lebar kolimator $w = \Delta x$. Hal ini menunjukkan bahwa lebar kolimator yang lebih lebar dari yang umum digunakan (dalam penelitian ini diwakili oleh $w = 3\Delta x$ dan $w = 5\Delta x$) relatif tidak berpengaruh pada kualitas citra hasil rekonstruksi dengan metode ART. Pada metode FBP lebar kolimator berpengaruh pada kualitas citra yang dihasilkannya.

Grafik hubungan antara nilai rmsd dan lebar kolimator pada gambar 4.18 menunjukkan bahwa nilai rmsd hasil rekonstruksi menggunakan metode ART lebih kecil daripada metode FBP. Hal ini menunjukkan bahwa kualitas citra hasil rekonstruksi menggunakan metode ART lebih baik daripada kualitas citra hasil rekonstruksi menggunakan FBP.

Grafik gambar 4.18 juga menunjukkan bahwa untuk lebar kolimator yang lebih lebar nilai rmsd-nya cenderung meningkat, baik untuk metode rekonstruksi ART maupun metode rekonstruksi FBP. Tetapi tingkat kenaikannya untuk metode rekonstruksi FBP lebih besar daripada metode rekonstruksi ART. Hal ini menunjukkan bahwa lebar kolimator tidak berpengaruh untuk metode ART sedangkan untuk metode FBP berpengaruh.

Tabel 4.1 menunjukkan bahwa proses rekonstruksi metode ART membutuhkan waktu yang lebih lama dibandingkan dengan proses rekonstruksi

dengan metode FBP. Dari tabel dapat diketahui bahwa lama proses rekonstruksi dengan metode FBP lebih cepat dibandingkan proses rekonstruksi dengan metode ART.

Lama proses rekonstruksi menggunakan metode ART dengan lebar kolimator $w = \Delta x$ lebih cepat dibandingkan dengan lebar kolimator $w = 3\Delta x$ dan $w = 5\Delta x$. Walaupun proses rekonstruksi dengan metode ART membutuhkan waktu yang lama tetapi kualitas citra hasil rekonstruksinya lebih baik daripada kualitas citra hasil rekonstruksi dengan metode FBP.

BAB V**KESIMPULAN DAN SARAN****5.1 Kesimpulan**

Berdasarkan analisa data dan pembahasan pada penelitian pengaruh lebar kolimator terhadap kualitas citra rekonstruksi dengan menggunakan metode iterasi aljabar, dapat disimpulkan bahwa :

1. Dengan menggunakan program Borland C++ Builder dan parameter-parameter yang mengaju pada keadaan sesungguhnya dapat dibuat program simulator pengambilan data dengan menggunakan lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$.
2. Dengan menggunakan program Borland C++ Builder dan parameter-parameter yang mengaju pada keadaan yang sesungguhnya dapat dibuat program simulator rekonstruksi citra menggunakan metode ART dengan lebar kolimator $w = \Delta x$, $w = 3\Delta x$, dan $w = 5\Delta x$.
3. Untuk lebar kolimator yang berbeda kualitas citra hasil rekonstruksi dengan direkonstruksi menggunakan metode ART lebih mendekati objek referensi dibandingkan dengan kualitas citra hasil rekonstruksi menggunakan metode FBP. Sedangkan kualitas citra hasil rekonstruksi dengan metode FBP yang mendekati objek referensi adalah kualitas citra dari sinogram dengan proses *scanning* $w = \Delta x$.
4. Proses rekonstruksi metode ART membutuhkan waktu lebih lama dari proses rekonstruksi metode FBP.

5.2 Saran

Pada simulator program rekonstruksi citra yang dibuat dalam penelitian ini belum dapat secara langsung mengenal lebar kolimator yang digunakan pada saat proses *scanning*. Untuk itu perlu dilakukan penelitian lebih lanjut agar simulator program rekonstruksi dapat langsung membedakan bahwa sinogram yang dibuka menggunakan lebar kolimator $w = \Delta x$, $w = 3\Delta x$ atau $w = 5\Delta x$, tanpa harus mengetahui proses *scanning*-nya.

DAFTAR PUSTAKA

- Agarwall, B.K., 1989, *X-Ray Spectroscopy: An Introduction*, 2thed., springer Verlag, Berlin.
- Ain, Khusnul dan Ukhrowiyah, Nuril, 2004, *Algebraic Iteration Accurate Recontraction Recontraction Method In Computer Tomography System By Weight Value Alteration Cloud In Cell Model*, Proceeding of Fisrt Jogja Regional Physics Conference.
- Brigham, E.O., 1988, *The Fast Fourier Transform and its Application*, Prentice-Hall International, Inc.
- Censor, Y., 1983, *Finite Series Expansion Reconstruction Methods*, Proceedings of the IEEE, 70(3).
- Gaskill, J.D., 1978, *Linear Systems, Fourier Transform, and Optics*, John Wiley & Sons, New York.
- Gordon, R., 1974, *A Tutorial on ART (Algebraic Reconstruction Technique)*, IEEE Transaction on Nuclear Science.
- Herman, G.T., 1980, *Image Reconstruction from Projection : The Fundamentals of Computed Tomography*, Academic Press, new York.
- Mueller, K., 1998, *Fast and Accurate three-Dimensional Reconstruction from Cone-Beam projection data Using Algebraic Methods*. Ph.D. Thesis, Ohio State University.
- Nugroho, W. dan Gede B.S., 2002, *Application Software For Computed Tomography Image Comparison*, Presented on Seminar Dies Natalis 47th FMIPA-UGM, Yogyakarta.
- Pramono, D., 1997, *Mudah Menguasai C++ Builder*, Jilid 1 dan 2, PT Gramedia,

Jakarta.

Suparta, G.B. dan Kusminarto, 1986, *Teknik Rekonstruksi secara aljabar untuk tomografi komputer*, Prosiding symposium fisika nasional, Yogyakarta.

Suparta, G.B. dan Kusminarto, 1989, "*Teknik Rekonstruksi Citra Tomografi Secara Aljabar*", Prosiding SFN.XII-HFI, Yogyakarta, 477-483.

Tsoufanidis, 1981, *Measurement and Detection of Radiation*, Hemisphere Publishing Corporation, Perancis.

Wells, P., Smith R. and G.B. Suparta, 1997, "Sampling the Sinogram in Computed Tomography", *Material Evaluation*, 55(7), 772-776.

Widodo, C.S., 2000, *Metode Pengukuran Sensitivitas Perangkat Tomografi Komputer Transmisi*, Universitas Gajah Mada, Yogyakarta.

LAMPIRAN A

Program Simulator Proses Scanning

```

//-----
#include <vc1.h>
#pragma hdrstop
#include "Unitcitra.h"

//-----

#pragma package(smart_init)
#pragma link "CGAUGES"
#pragma resource "*.dfm"
#pragma link "CGAUGES"
TMukaUtama *MukaUtama;
int max_sintetis;
int k;

////////////////////////////////////
//function
////////////////////////////////////
//-----

void __fastcall
TMukaUtama::Ketemumaxmin(float*img[TM
X])
{
max_test=img[1][1];
min_test=img[1][1];
for(int baris=1;baris<=nray;baris++){
for(int kolom=1;kolom<=ncol;kolom++){

if(max_test<img[baris][kolom])max_test=img[
baris][kolom];

if(min_test>img[baris][kolom]min_test=img[
baris][kolom];
total=total+img[baris][kolom];
}
}
}
}
}
//-----

bool __fastcall TMukaUtama::savedata(float
*img[TMX], char *namafile)
{
FILE *pfp;
pfp=fopen(namafile, "\w+");
fprintf(pfp, "%d %3d\n", nray, ncol);
for(int ray=1; ray<=(nray); ray++){
for(int col=1; col<=(ncol); col++){
fprintf(pfp, "%10.2f", img[ray][col]);
fprintf(pfp, "\n");
}
fclose(pfp);
return 1;
}
}
//-----

```

```

bool __fastcall
TMukaUtama::TampilCitra(float*img[TMX],
TImage*Image)
{
float temp;
max_test=img[1][1];
min_test=img[1][1];
for(int baris=1;baris<=nray;baris++){
for(int kolom=1;kolom<=ncol;kolom++){

if(max_test<img[baris][kolom])max_test=img[
baris][kolom];

if(min_test>img[baris][kolom]min_test=img[
baris][kolom];
total=total+img[baris][kolom];
}
}
width=Image->Width;
height=Image->Height;
dx=(float)width/ncol;
dy=(float)height/nray;
CG->MaxValue=width;
for(int i=1;i<=width;i++){
py=i/dy;
for(int j=1;j<=height;j++){
px=j/dx;
temp=(img[px][py]-
min_test)*255/(max_test-min_test);
SetPixel(Image->Canvas-
>Handle,i,j,RGB(temp,temp,temp));
}
Application->ProcessMessages();
CG->AddProgress(1);
}
CG->Progress=0;
Image->Refresh();
return 1;
}
//-----

bool __fastcall
TMukaUtama::TampilCitra2(float*img[TMX],
TImage*Image)
{
float temp;
max_test=img[1][1];
min_test=img[1][1];
nray=int(ncol*pi/2)-1;
for(int baris=1;baris<=nray;baris++){
for(int kolom=1;kolom<=ncol;kolom++){

if(max_test<img[baris][kolom])max_test=img[
baris][kolom];

if(min_test>img[baris][kolom]min_test=img[
baris][kolom];
total=total+img[baris][kolom];
}
}
width=Image->Width;

```

```

height=Image->Height;
dx=(float)width/ncol;
dy=(float)height/nray;
for(int i=1;i<=width;i++){
    py=i/dy;
    for(int j=1;j<=height;j++){
        px=j/dx;
        temp=(img[px][py]-
min_test)*255/(max_test-min_test);
        SetPixel(Image->Canvas-
>Handle,i,j,RGB(temp,temp,temp));
    }
}
return 1;
}
//-----

bool __fastcall
TMukaUtama::readCTdata(float*img[TMX],c
har*namafile)
{
FILE *pfp;
float data,dat;
pfp=fopen(namafile,"r");
fscanf(pfp,"%d %d\n",&nray,&ncol);
for(int
baris_ke=0;baris_ke<=(nray+1);baris_ke++){

img[baris_ke]=(float*)fcalloc(ncol+2,sizeoff
float));
}
for(int p=1;p<=nray;p++){
    for(int q=1;q<=ncol;q++){
        fscanf(pfp,"%f",&data);
        dat=(float)(data);
        img[p][q]=dat;
    }
    fscanf(pfp,"n");
}
fclose(pfp);
return 1;
}
//-----

__fastcall
TMukaUtama::TMukaUtama(TComponent*
Owner)
: TForm(Owner)
{
ImageCitra->Picture=NULL;status=0;
}
//-----

void __fastcall
TMukaUtama::OpenClick(TObject *Sender)
{
if(BukaFile->Execute()){
    FileAsli=BukaFile->Files-
>Strings[0].c_str();

read=readCTdata(citra,FileAsli.c_str());
if(read!=0){
    ImageCitra->Picture=NULL;status=0;
    Ketemumaxmin(citra);
    status=TampilCitra(citra,MukaUtama-
>ImageCitra);
}
}
//-----

void __fastcall
TMukaUtama::TutupClick(TObject *Sender)
{
Close();
}
//-----

void __fastcall
TMukaUtama::SinoW1Click(TObject
*Sender)
{
Image1->Picture=NULL;status1=0;
float a,b,r2dalam,sinteta,costeta;
int nteta,pusat,xai,yai;
float dteta,teta,xa,ya;
float dx,dx1,dy1,b00,b01,b10,b11;
int px,py;
nray=int(ncol*pi/2)-1;
for(int
baris_ke=0;baris_ke<=(nray+1);baris_ke++){

citra3[baris_ke]=(float*)fcalloc(ncol+2,sizeo
f(float));
}
nteta=nray;
dteta=pi/nteta;
pusat=(ncol+1)/2;
CG->MaxValue=nray;
dy=0;
for(int k=1;k<=nteta;k++){
    teta=(float)(k-1)*dteta;
    sinteta=sin(teta);
    costeta=cos(teta);
    for(int i=1;i<=ncol;i++){
        for(int j=1;j<=ncol;j++){
            a=float(i-pusat);
            b=float(pusat-j);
            r2dalam=(a*a+b*b);
            if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                xa=a*costeta+b*sinteta;
                ya=-a*sinteta+b*costeta;
                xai=int(xa);
                yai=int(ya);
                px=1;py=1;
                dx=xa-xai;
                if(dx<0.0)px=-1;
                //dy=ya-yai;
                if(dy<0.0)py=-1;
                dx=fabs(dx);

```

```

dy=fabs(dy);
dx1=fabs(1.0-dx);
dy1=fabs(1.0-dy);

b00=(dx1*dy1)*citra[yai+pusat][xai+pusat];

b01=(dx1*dy)*citra[yai+pusat+py][xai+pusat];

b10=(dx*dy1)*citra[yai+pusat][xai+pusat+px];

b11=(dx*dy)*citra[yai+pusat+py][xai+pusat+px];

citra3[k][i]+=(b00+b01+b10+b11)/ncol;
}
}
}
Application->ProcessMessages();
CG->AddProgress(1);
}
status1=TampilCitra(citra3,Image1);
CG->Progress=0;
}
//-----
void __fastcall
TMukaUtama::sinoW3Click(TObject *Sender)
{
Image1->Picture=NULL;status1=0;
float a,b,r2dalam,sinteta,costeta;
int nteta,pusat,xai,yai;
float dteta,teta,xa,ya;
float dx,dx1,dy1,b00,b01,b10,b11;
int px,py;
int i,j;
nray=int(ncol*pi/2)-1;
for(int
baris_ke=0;baris_ke<=(nray+1);baris_ke++){
citra3[baris_ke]=(float*)falloc(ncol+2,sizeo
f(float));
}
nteta=nray;

dteta=pi/nteta;
pusat=(ncol+1)/2;
CG->MaxValue=nray;
for(k=1;k<=nteta;k++){

teta=(float)(k-1)*dteta;
sinteta=sin(teta);
costeta=cos(teta);
//////////////////////////////////////////////////
for(i=1;i<=ncol;i++){
citra3[k][i]=0;
for(int j=1;j<=ncol;j++){
a=float(i-pusat);
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4)
{
xa=a*costeta+b*sinteta;
ya=-a*sinteta+b*costeta;
xai=int(xa);
yai=int(ya);
px = 1; py = 1;
dx = xa-xai;
if (dx<0.0) px = -1;
// dy = ya-yai;
if (dy<0.0) py = -1;
dx = fabs(dx);
dy = fabs(dy);
dx1 = fabs(1.0-dx);
dy1 = fabs(1.0-dy);
b00 =
(dx1*dy1)*citra[yai+pusat][xai+pusat];
b01 =
(dx1*dy)*citra[yai+pusat+py][xai+pusat];
b10 =
(dx*dy1)*citra[yai+pusat][xai+pusat+px];
b11 =
(dx*dy)*citra[yai+pusat+py][xai+pusat+px];
citra3[k][i]+=(b00+b01+b10+b11);
}
}
for(j=1;j<=ncol;j++){
a=float(i+1-pusat);
b=float(pusat-j);

```

```

r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4)
{
    xa=a*costeta+b*sinteta;
    ya=-a*sinteta+b*costeta;
    xai=int(xa);
    yai=int(ya);
    px = 1; py = 1;
    dx = xa-xai;
    if (dx<0.0) px = -1;
    // dy = ya-yai;
    if (dy<0.0) py = -1;
    dx = fabs(dx);
    dy = fabs(dy);
    dx1 = fabs(1.0-dx);
    dy1 = fabs(1.0-dy);
    b00 =
(dx1*dy1)*citra[yai+pusat][xai+pusat];
    b01 =
(dx1*dy)*citra[yai+pusat+py][xai+pusat];
    b10 =
(dx*dy1)*citra[yai+pusat][xai+pusat+px];
    b11 =
(dx*dy)*citra[yai+pusat+py][xai+pusat+px];
    citra3[k][i]+=(b00+b01+b10+b11);
}
}
//-----
}
Application->ProcessMessages();
CG->AddProgress(1);
}
status1=TampilCitra(citra3,Image1);
CG->Progress=0;
}
//-----

void __fastcall
TMukaUtama::sinoW5Click(TObject *Sender)
{
    Image1->Picture=NULL;status1=0;
    float a,b,r2dalam,sinteta, costeta;
    int nteta, pusat, xai, yai;
    float dteta, teta, xa, ya;
    float dx, dx1, dy1, b00, b01, b10, b11;
    int px, py;
    int i, j;
    nray=int(ncol*pi/2)-1;
    for(int
    baris_ke=0; baris_ke<=(nray+1); baris_ke++){
    citra2[baris_ke]=(float*)falloc(ncol+2, sizeo
f(float));
    citra3[baris_ke]=(float*)falloc(ncol+2, sizeo
f(float));
    }
    nteta=nray;
    dteta=pi/nteta;
    pusat=(ncol+1)/2;
    CG->MaxValue=nray;
    for(k=1; k<=nteta; k++){
        teta=(float)(k-1)*dteta;
        sinteta=sin(teta);
        costeta=cos(teta);
        //-----
        for(i=1; i<=ncol; i++){
            for(j=1; j<=ncol; j++){
                a=float(i-pusat);
                b=float(pusat-j);
                r2dalam=(a*a+b*b);
                if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4)
                {
                    xa=a*costeta+b*sinteta;
                    ya=-a*sinteta+b*costeta;
                    xai=int(xa);
                    yai=int(ya);
                    px = 1; py = 1;
                    dx = xa-xai;
                    if (dx<0.0) px = -1;
                    // dy = ya-yai;
                    // if (dy<0.0) py = -1;
                    dx = fabs(dx);
                    dy = fabs(dy);
                    dx1 = fabs(1.0-dx);
                    dy1 = fabs(1.0-dy);
                    b00 =
(dx1*dy1)*citra[yai+pusat][xai+pusat];
                    b01 =
(dx1*dy)*citra[yai+pusat+py][xai+pusat];
                    b10 =
(dx*dy1)*citra[yai+pusat][xai+pusat+px];
                    b11 =
(dx*dy)*citra[yai+pusat+py][xai+pusat+px];
                    citra3[k][i]+=(b00+b01+b10+b11);
                }
            }
            for(j=1; j<=ncol; j++){
                a=float(i+1-pusat);
                b=float(pusat-j);
                r2dalam=(a*a+b*b);
                if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4)
                {
                    xa=a*costeta+b*sinteta;
                    ya=-a*sinteta+b*costeta;
                    xai=int(xa);
                    yai=int(ya);
                    px = 1; py = 1;
                    dx = xa-xai;
                    if (dx<0.0) px = -1;
                    // dy = ya-yai;
                    // if (dy<0.0) py = -1;
                    dx = fabs(dx);
                    dy = fabs(dy);
                    dx1 = fabs(1.0-dx);

```

```

        dy1 = fabs(1.0-dy);
        b00 =
(dx1*dy1)*citra[yai+pusat][xai+pusat];
        b01 =
(dx1*dy)*citra[yai+pusat+py][xai+pusat];
        b10 =
(dx*dy1)*citra[yai+pusat][xai+pusat+px];
        b11 =
(dx*dy)*citra[yai+pusat+py][xai+pusat+px];
citra3[k][i]+=(b00+b01+b10+b11);
    }
}
//////////

for(j=1;j<=ncol;j++){
a=float(i+2-pusat);
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4)
{
    xa=a*coseteta+b*sinteta;
    ya=-a*sinteta+b*coseteta;
    xai=int(xa);
    yai=int(ya);
    px = 1; py = 1;
    dx = xa-xai;
    if (dx<0.0) px = -1;
    // dy = ya-yai;
    // if (dy<0.0) py = -1;
    dx = fabs(dx);
    dy = fabs(dy);
    dx1 = fabs(1.0-dx);
    dy1 = fabs(1.0-dy);
    b00 =
(dx1*dy1)*citra[yai+pusat][xai+pusat];
    b01 =
(dx1*dy)*citra[yai+pusat+py][xai+pusat];
    b10 =
(dx*dy1)*citra[yai+pusat][xai+pusat+px];
    b11 =
(dx*dy)*citra[yai+pusat+py][xai+pusat+px];
citra3[k][i]+=(b00+b01+b10+b11);
}
}
for(j=1;j<=ncol;j++){
a=float(i-1-pusat);
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4)
{
    xa=a*coseteta+b*sinteta;
    ya=-a*sinteta+b*coseteta;
    xai=int(xa);
    yai=int(ya);
    px = 1; py = 1;
    dx = xa-xai;
    if (dx<0.0) px = -1;
}
}
}
}
Application->ProcessMessages();
CG->AddProgress(1);
}
status1=TampilCitra(citra3,Image1);
CG->Progress=0;
}
//-----
-----

```

```
void __fastcall
TMukaUtama::Save1Click(TObject *Sender)
{
if(status!=1)Application-
>MessageBox("Sorry, There is no data input!",
"Warning",MB_OK|MB_ICONWARNING);
else if (SimpanFile->Execute()){
FileAsli=SimpanFile->Files-
>Strings[0].c_str();
save=savedata(citra3,FileAsli.c_str());
}
}
//-----
-----
```


LAMPIRAN B**Program Simulator Proses Rekonstruksi**

```

//-----
#include <vcl.h>
#pragma hdrstop
#include "Unitcitra.h"
#include <math.h>
#include "unitpengaturan.h"
#include "unitinfo.h"
#include "unititerasi.h"
//-----

#pragma package(smart_init)
#pragma link "CGAUGES"
#pragma resource "*.dfm"
TMukaUtama *MukaUtama;
int max_sintetis;
int stop=0;
////////////////////
//function
////////////////////
//-----

float __fastcall TMukaUtama::Normal(float
data,float min,float max)
{
return (data-min)*max_sintetis/(max-min);
}

void __fastcall
TMukaUtama::Normalisasi(float *
img[TMX],int nbaris,int nkolom)
{
Ketemumaxmin(img,nbaris,nkolom);
for(int baris=1;baris<=nbaris;baris++)
for(int kolom=1;kolom<=nkolom;kolom++)
img[baris][kolom]/=max_test;
}

float __fastcall
TMukaUtama::SigmaLuas1(TBagianPixel
bagian)
{
return bagian.b00+bagian.b01+
bagian.b10+bagian.b11;
}

float __fastcall
TMukaUtama::SigmaLuas2(TBagianPixel
bagian)
{
return
pow(bagian.b00,2)+pow(bagian.b01,2)+
pow(bagian.b10,2)+pow(bagian.b11,2);
}

```

```

//-----
TBagianPixel __fastcall
TMukaUtama::HitungLuasPixel1(float a, float
b, float sinteta,
float costeta, int ncol)
{
TBagianPixel temp;
float xa,ya,dx,dx1,dy1;
xa=a*costeta+b*sinteta;
ya=-a*sinteta+b*costeta;
temp.xai=int(xa);
temp.yai=int(ya);
temp.px=1;temp.py=1;
dx=xa-temp.xai;
if(dx<0.0) temp.px=-1;
// dy=ya-temp.yai;
if(dy<0.0) temp.py=-1;
dx=fabs(dx);
dy=fabs(dy);
dx1=fabs(1.0-dx);
dy1=fabs(1.0-dy);
temp.b00=(dx1*dy1);
temp.b01=(dx1*dy);
temp.b10=(dx*dy1);
temp.b11=(dx*dy);
return temp;
}

TBagianPixel __fastcall
TMukaUtama::HitungLuasPixel2(float a, float
b, float sinteta,
float costeta, int ncol,int pusat,float
*img[TMX])
{
TBagianPixel temp;
float xa,ya,dx,dx1,dy1;
xa=a*costeta+b*sinteta;
ya=-a*sinteta+b*costeta;
temp.xai=int(xa);
temp.yai=int(ya);
temp.px=1;temp.py=1;
dx=xa-temp.xai;
if(dx<0.0) temp.px=-1;
// dy=ya-temp.yai;
if(dy<0.0) temp.py=-1;
dx=fabs(dx);
dy=fabs(dy);
dx1=fabs(1.0-dx);
dy1=fabs(1.0-dy);

temp.b00=(dx1*dy1)*img[temp.yai+pusat][te
mp.xai+pusat];

temp.b01=(dx1*dy)*img[temp.yai+pusat+tem
p.py][temp.xai+pusat];

temp.b10=(dx*dy1)*img[temp.yai+pusat][tem
p.xai+pusat+temp.px];
}

```



```

    }
}
width=Image->Width;
height=Image->Height;
dx=(float)width/ncol;
dy=(float)height/ncol;
CG->MaxValue=width;
lbKeterangan->Caption="Sedang menyiapkan
gambar hasil rekonstruksi...";
for(int i=1;i<=width;i++){
    py=i/dy;
    for(int j=1;j<=height;j++){
        px=j/dx;
        temp=(img[px][py]-
min_test)*255/(max_test-min_test);
        SetPixel(Image->Canvas-
>Handle,i,j,RGB(temp,temp,temp));
    }
    Application->ProcessMessages();
    CG->AddProgress(1);
}
lbKeterangan->Caption="";
CG->Progress=0;
Image->Refresh();
return 1;
}

//-----
bool __fastcall
TMukaUtama::TampilCitra2(float*img[TMX],
TImage*Image)
{
float temp;
max_test=img[1][1];
min_test=img[1][1];
nray=int(ncol*pi/2)-1;
for(int baris=1;baris<=nray;baris++){
    for(int kolom=1;kolom<=ncol;kolom++){

if(max_test<img[baris][kolom])max_test=img[
baris][kolom];

if(min_test>img[baris][kolom])min_test=img[
baris][kolom];
        total=total+img[baris][kolom];
    }
}
width=Image->Width;
height=Image->Height;
dx=(float)width/ncol;
dy=(float)height/nray;
CG->MaxValue=width;
lbKeterangan->Caption="Sedang menyiapkan
gambar sinogram...";
for(int i=1;i<=width;i++){
    py=i/dy;
    for(int j=1;j<=height;j++){
        px=j/dx;
        temp=(img[px][py]-
min_test)*255/(max_test-min_test);

        SetPixel(Image->Canvas-
>Handle,i,j,RGB(temp,temp,temp));
    }
    Application->ProcessMessages();
    CG->AddProgress(1);
}
lbKeterangan->Caption="";
CG->Progress=0;
Image->Refresh();
return 1;
}

//-----
bool __fastcall
TMukaUtama::readCTdata(float*img[TMX],c
har*namafile)
{
FILE *pfp;
float data,dat;
pfp=fopen(namafile,"r");
fscanf(pfp,"%d %d
%d\n",&nray,&ncol,&max_sintetis);
for(int
baris_ke=0;baris_ke<=(nray+1);baris_ke++){

img[baris_ke]=(float*)fcalloc(ncol+2,sizeof(
float));
}
CG->MaxValue=nray;
lbKeterangan->Caption="Sedang membuka
data sinogram...";
for(int p=1;p<=nray;p++){
    for(int q=1;q<=ncol;q++){
        fscanf(pfp,"%f",&data);
        dat=(float)(data);
        img[p][q]=dat;
    }
    fscanf(pfp,"%n");
    Application->ProcessMessages();
    CG->AddProgress(1);
}

fclose(pfp);
lbKeterangan->Caption="";
CG->Progress=0;
return 1;
}

//-----
__fastcall
TMukaUtama::TMukaUtama(TComponent*
Owner)
: TForm(Owner)
{
}

//-----

```

```

void __fastcall
TMukaUtama::OpenClick(TObject *Sender)
{
    LebarRadiasi=1;
    iterasi=1;
    if (BukaFile->Execute()){
        FileAsli=BukaFile->Files-
>Strings[0].c_str();
        read=readCTdata(citra3,FileAsli.c_str());
        if(read!=0){
            Image1->Picture=NULL;status=0;
            // Normalisasi(citra3,nray,ncol);
            status1=TampilCitra2(citra3,MukaUtama-
>Image1);
        }
    }
}
//-----
void __fastcall
TMukaUtama::bobotkuadrat1Click()
{
    float a,b,r2dalam,sinteta,costeta;
    int nteta,pusat,xai,yai;
    float dteta,teta,xa,ya;
    float dx,dx1,dy1,b00,b01,b10,b11;
    nray=int(ncol*pi/2)-1;
    for(int
    baris_ke=0;baris_ke<=nray;baris_ke++){

        citra4[baris_ke]=(float*)fcalloc(ncol+2,sizeo
f(float));
    }
    nteta=nray;
    dteta=pi/nteta;
    pusat=(ncol+1)/2;
    CG->MaxValue=nray;
    lbKeterangan->Caption="Sedang menghitung
    bobot kuadrat...";
    for(int k=1;k<=nteta;k++){
        teta=(float)(k-1)*dteta;
        sinteta=sin(teta);
        costeta=cos(teta);
        for(int i=1;i<=ncol;i++){
            for(int j=1;j<=ncol;j++){
                a=float(i-pusat);
                b=float(pusat-j);
                r2dalam=(a*a+b*b);
                if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                    xa=a*costeta+b*sinteta;
                    ya=-a*sinteta+b*costeta;
                    xai=int(xa);
                    yai=int(ya);
                    px=1;py=1;
                    dx=xa-xai;
                    if(dx<0.0)px=-1;
                    dy=ya-yai;
                    if(dy<0.0)py=-1;

                    dx=fabs(dx);
                    dy=fabs(dy);
                    dx1=fabs(1.0-dx);
                    dy1=fabs(1.0-dy);
                    b00=(dx1*dy1);
                    b01=(dx1*dy);
                    b10=(dx*dy1);
                    b11=(dx*dy);

                    citra4[k][i]+=(b00*b00+b01*b01+b10*b10+b
                    11*b11);
                }
            }
        }
        Application->ProcessMessages();
        CG->AddProgress(1);
    }
    lbKeterangan->Caption="";
    CG->Progress=0;
}
//-----

void __fastcall TMukaUtama::ART1Click()
{
    float a,b,r2dalam,sinteta,costeta;
    int nteta,pusat,xai,yai;
    float dteta,teta,xa,ya;
    float dx,dx1,dy1,b00,b01,b10,b11;
    nray=int(ncol*pi/2)-1;
    for(int
    baris_ke=0;baris_ke<=nray;baris_ke++){

        citra5[baris_ke]=(float*)fcalloc(ncol+2,sizeo
f(float));
    }
    nteta=nray;
    dteta=pi/nteta;
    pusat=(ncol+1)/2;
    CG->MaxValue=nray;
    lbKeterangan->Caption="Sedang inisialisasi
    ART...";
    for(int k=1;k<=nteta;k++){
        teta=(float)(k-1)*dteta;
        sinteta=sin(teta);
        costeta=cos(teta);
        for(int i=1;i<=ncol;i++){
            for(int j=1;j<=ncol;j++){
                a=float(i-pusat);
                b=float(pusat-j);
                r2dalam=(a*a+b*b);
                if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                    xa=a*costeta+b*sinteta;
                    ya=-a*sinteta+b*costeta;
                    xai=int(xa);
                    yai=int(ya);
                    px=1;py=1;

```

```

dx=xa-xai;
if(dx<0.0)px=-1;
dy-ya-yai;
if(dy<0.0)py=-1;
dx=fabs(dx);
dy=fabs(dy);
dx1=fabs(1.0-dx);
dy1=fabs(1.0-dy);

b00=(dx1*dy1)*citra5[yai+pusat][xai+pusat];
b01=(dx1*dy)*citra5[yai+pusat+py][xai+pusat];
b10=(dx*dy1)*citra5[yai+pusat][xai+pusat+px];
b11=(dx*dy)*citra5[yai+pusat+py][xai+pusat+px];
    citra6[k][i]+=(b00+b01+b10+b11);
}
for(int j=1;j<=ncol;j++){
a=float(i-pusat);
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
xa=a*costeta+b*sinteta;
ya=-a*sinteta+b*costeta;
xai=int(xa);
yai=int(ya);
px=1;py=1;
dx=xa-xai;
if(dx<0.0)px=-1;
dy-ya-yai;
if(dy<0.0)py=-1;
dx=fabs(dx);
dy=fabs(dy);
dx1=fabs(1.0-dx);
dy1=fabs(1.0-dy);
b00=(dx1*dy1);
b01=(dx1*dy);
b10=(dx*dy1);
b11=(dx*dy);

citra5[yai+pusat][xai+pusat]+=b00*(citra3[k][i]-
    citra6[k][i])/citra4[k][i];

citra5[yai+pusat+py][xai+pusat]+=b01*(citra3[k][i]-
    citra6[k][i])/citra4[k][i];

citra5[yai+pusat][xai+pusat+px]+=b10*(citra3[k][i]-
    citra6[k][i])/citra4[k][i];

citra5[yai+pusat+py][xai+pusat+px]+=b11*(citra3[k][i]-
    citra6[k][i])/citra4[k][i];
}
}

citra6[k][i]/citra4[k][i];
}
}
Application->ProcessMessages();
CG->AddProgress(1);
}
lbKeterangan->Caption="";
CG->Progress=0;
iterasike=1;
}
//-----
void __fastcall
TMukaUtama::iterasiW1Click(TObject
*Sender)
{
status4=0;
float a,b,r2dalam,sinteta, costeta,gamma;
int nteta,pusat,xai,yai;
float dteta,teta,xa,ya,discrepancy,edistance;
float dx,dx1,dy1,b00,b01,b10,b11;
nray=int(ncol*pi/2)-1;
for(int
baris_ke=0;baris_ke<=nray;baris_ke++){
citra7[baris_ke]=(float*)falloc(ncol+2,sizeo
f(float));
citra8[baris_ke]=(float*)falloc(ncol+2,sizeo
f(float));
}
discrepancy=0.0;edistance=0.0;
gamma=0.5;
for(int i=1;i<=ncol;i++){
for(int j=1;j<=ncol;j++){
citra7[i][j]=citra5[i][j];
}
}
nteta=nray;
dteta=pi/nteta;
pusat=(ncol+1)/2;
CG->MaxValue=nray;
lbKeterangan->Caption="Sedang melakukan
iterasi ART ke "+IntToStr(iterasike)+" ...";
for(int k=1;k<=nteta;k++){
teta=(float)(k-1)*dteta;
sinteta=sin(teta);
costeta=cos(teta);
for(int i=1;i<=ncol;i++){
for(int j=1;j<=ncol;j++){
a=float(i-pusat);
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
xa=a*costeta+b*sinteta;
ya=-a*sinteta+b*costeta;
xai=int(xa);
yai=int(ya);
px=1;py=1;
}
}
}
}
}
}

```

```

dx=xa-xai;
if(dx<0.0)px=-1;
dy-ya-yai;
if(dy<0.0)py=-1;
dx=fabs(dx);
dy=fabs(dy);
dx1=fabs(1.0-dx);
dy1=fabs(1.0-dy);

b00=(dx1*dy1)*citra7[yai+pusat][xai+pusat];
b01=(dx1*dy)*citra7[yai+pusat+py][xai+pusat];
b10=(dx*dy1)*citra7[yai+pusat][xai+pusat+px];
b11=(dx*dy)*citra7[yai+pusat+py][xai+pusat+px];
citra8[k][i]+=(b00+b01+b10+b11);
}
for(int j=1;j<=ncol;j++){
a=float(i-pusat);
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
xa=a*coseteta+b*sinteta;
ya=-a*sinteta+b*coseteta;
xai=int(xa);
yai=int(ya);
px=1;py=1;
dx=xa-xai;
if(dx<0.0)px=-1;
dy-ya-yai;
if(dy<0.0)py=-1;
dx=fabs(dx);
dy=fabs(dy);
dx1=fabs(1.0-dx);
dy1=fabs(1.0-dy);
b00=(dx1*dy1);
b01=(dx1*dy);
b10=(dx*dy1);
b11=(dx*dy);

citra7[yai+pusat][xai+pusat]+=gamma*b00*(citra3[k][i]-citra8[k][i])/citra4[k][i];
if(citra7[yai+pusat][xai+pusat]<0)
citra7[yai+pusat][xai+pusat]=0;

citra7[yai+pusat+py][xai+pusat]+=gamma*b01*(citra3[k][i]-citra8[k][i])/citra4[k][i];

if(citra7[yai+pusat+py][xai+pusat]<0)
citra7[yai+pusat+py][xai+pusat]=0;

citra7[yai+pusat][xai+pusat+px]+=gamma*b10*(citra3[k][i]-
citra8[k][i])/citra4[k][i];
if(citra7[yai+pusat][xai+pusat+px]<0)
citra7[yai+pusat][xai+pusat+px]=0;

citra7[yai+pusat+py][xai+pusat+px]+=gamma*b11*(citra3[k][i]-citra8[k][i])/citra4[k][i];
if(citra7[yai+pusat+py][xai+pusat+px]<0)
citra7[yai+pusat+py][xai+pusat+px]=0;
}
}
discrepancy+=(citra3[k][i]-citra8[k][i])*citra3[k][i]-citra8[k][i])/citra4[k][i];
}
Application->ProcessMessages();
CG->AddProgress(1);
}
for(int i=1;i<=ncol;i++){
for(int j=1;j<=ncol;j++){
edistance+=(citra5[i][j]-citra7[i][j])*citra5[i][j]-citra7[i][j];
citra5[i][j]=citra7[i][j];
}
}

status4=TampilCitra(citra7,Image4);
SG->Cells[0][iterasike]=iterasike;
SG->Cells[1][iterasike]=sqrt(discrepancy/(nteta*ncol));
SG->Cells[2][iterasike]=sqrt(edistance/(ncol*ncol));
if((SG->RowCount-11)>=1)
SG->TopRow=SG->RowCount-11;
int bar=SG->RowCount;
bar++;
SG->RowCount=bar;
IbKeterangan->Caption="";
CG->Progress=0;
iterasike++;
}
//-----
void _fastcall
TMukaUtama::bobotkuadrat3Click()
{
float a,b,r2dalam,sinteta, costeta;
TBagianPixel bagian;
int nteta,pusat;
float dteta,teta;
for(int
baris_ke=0;baris_ke<=(nray+1);baris_ke++){
citra10[baris_ke]=(float*)falloc(ncol+2,size
of(float));

```

```

}
nteta=nray;
dteta=pi/nteta;
pusat=(ncol+1)/2;
CG->MaxValue=nray;
lbKeterangan->Caption="Sedang menghitung
bobot kuadrat...";
for(int k=1;k<=nteta;k++){
    teta=(float)(k-1)*dteta;
    sinteta=sin(teta);
    costeta=cos(teta);
    for(int i=1;j<=ncol;j++){
        citra10[k][i]=0;
        for(int g=-1;g<=1;g++){
            a=float(i+g-pusat);
            for(int j=1;j<=ncol;j++){
                b=float(pusat-j);
                r2dalam=(a*a+b*b);
                if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                    bagian=HitungLuasPixel1(a,b,sinteta,costeta,ncol);
                    citra10[k][i]+=SigmaLuas2(bagian);
                }
            }
        }
    }
    Application->ProcessMessages();
    CG->AddProgress(1);
}
lbKeterangan->Caption="";
CG->Progress=0;
}
//-----
void __fastcall TMukaUtama::ART3Click()
{
    float a,b,r2dalam,sinteta,costeta;
    int nteta,pusat;
    TBagianPixel bagian;
    float dteta,teta;
    for(int
    baris_ke=0;baris_ke<=(nray+1);baris_ke++){

        citra11[baris_ke]=(float*)farcalloc(ncol+2,size
        of(float));

        citra13[baris_ke]=(float*)farcalloc(ncol+2,size
        of(float));
    }
    nteta=nray;
    dteta=pi/nteta;
    pusat=(ncol+1)/2;
    CG->MaxValue=nray;
    lbKeterangan->Caption="Sedang inialisasi
    ART...";
    for(int k=1;k<=nteta;k++){
        teta=(float)(k-1)*dteta;
        sinteta=sin(teta);
        costeta=cos(teta);
        for(int i=1;i<=ncol;i++){
            citra13[k][i]=0;
            for(int g=-1;g<=1;g++){
                a=float(i+g-pusat);
                for(int j=1;j<=ncol;j++){
                    b=float(pusat-j);
                    r2dalam=(a*a+b*b);
                    if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                        bagian=HitungLuasPixel2(a,b,sinteta,costeta,ncol,pusat,citra11);
                        citra13[k][i]+=SigmaLuas1(bagian);
                    }
                }
            }
            for(int g=-1;g<=1;g++){
                a=float(i+g-pusat);
                for(int j=1;j<=ncol;j++){
                    b=float(pusat-j);
                    r2dalam=(a*a+b*b);
                    if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                        bagian=HitungLuasPixel1(a,b,sinteta,costeta,ncol);
                    }
                }
            }
            HitungART(citra11,citra3,citra13,citra10,bagian,pusat,k,i);
        }
    }
    Application->ProcessMessages();
    CG->AddProgress(1);
}
status4=TampilCitra(citra11,Image4);
lbKeterangan->Caption="";
CG->Progress=0;
iterasike=1;
}
//-----
void __fastcall
TMukaUtama::IterasiW3Click(TObject
*Sender)
{
    status4=0;
    float a,b,r2dalam,sinteta,costeta,gamma;
    TBagianPixel bagian;
    int nteta,pusat;
    float dteta,teta;
    for(int
    baris_ke=0;baris_ke<=nray;baris_ke++){
        citra7[baris_ke]=(float*)farcalloc(ncol+2,sizeo
        f(float));

        citra16[baris_ke]=(float*)farcalloc(ncol+2,size
        of(float));
    }
    discrepancy=0.0;edistance=0.0;
    gamma=1;
    for(int i=1;j<=ncol;j++){

```

```

for(int j=1;j<=ncol;j++){
    citra7[i][j]=citra11[i][j];
}
}

nteta=nray;
dteta=pi/nteta;
pusat=(ncol+1)/2;
CG->MaxValue=nray;
lbKeterangan->Caption="Sedang melakukan
iterasi ART ke "+IntToStr(iterasike)+" ...";
for(int k=1;k<=nteta;k++){
    teta=(float)(k-1)*dteta;
    sinteta=sin(teta);
    costeta=cos(teta);
    for(int i=1;i<=ncol;i++){
        citra16[k][i]=0;
        for(int g=-1;g<=1;g++){
            a=float(i+g-pusat);
            for(int j=1;j<=ncol;j++){
                b=float(pusat-j);
                r2dalam=(a*a+b*b);
                if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                    bagian=HitungLuasPixel2(a,b,sinteta,costeta,n
                    col,pusat,citra7);
                    citra16[k][i]+=SigmaLuas1(bagian);
                }
            }
        }
        for(int g=-1;g<=1;g++){
            a=float(i+g-pusat);
            for(int j=1;j<=ncol;j++){
                b=float(pusat-j);
                r2dalam=(a*a+b*b);
                if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                    bagian=HitungLuasPixel1(a,b,sinteta,costeta,n
                    col);
                }
            }
        }
        HitungART(citra7,citra3,citra16,citra10,bagian
        ,pusat,k,i,gamma,1);
    }
}
discrepancy+=(citra3[k][i]-citra16[k][i])*
(citra3[k][i]-
citra16[k][i])/citra10[k][i];
}
Application->ProcessMessages();
CG->AddProgress(1);
}
for(int i=1;i<=ncol;i++){
    for(int j=1;j<=ncol;j++){
        edistance+=(citra11[i][j]-
citra7[i][j])*(citra11[i][j]-citra7[i][j]);
        citra11[i][j]=citra7[i][j];
    }
}
status4=TampilCitra(citra7,Image4);

SG->Cells[0][iterasike]=iterasike;
SG-
>Cells[1][iterasike]=sqrt(discrepancy/(nteta*nc
ol));
SG-
>Cells[2][iterasike]=sqrt(edistance/(ncol*ncol
));
if((SG->RowCount-11)>=1)
SG->TopRow=SG->RowCount-11;
int bar=SG->RowCount;
bar++;
SG->RowCount=bar;
lbKeterangan->Caption="";
CG->Progress=0;
iterasike++;
}
//-----
void __fastcall
TMukaUtama::bobotkuadrat5Click()
{
    float a,b,r2dalam,sinteta,costeta;
    TBagianPixel bagian;
    int nteta,pusat;
    float dteta,teta;
    for(int
baris_ke=0;baris_ke<=(nray+1);baris_ke++){
        citra10[baris_ke]=(float*)falloc(ncol+2,size
of(float));
    }
    nteta=nray;
    dteta=pi/nteta;
    pusat=(ncol+1)/2;
    CG->MaxValue=nray;
    lbKeterangan->Caption="Sedang menghitung
bobot kuadrat...";
    for(int k=1;k<=nteta;k++){
        teta=(float)(k-1)*dteta;
        sinteta=sin(teta);
        costeta=cos(teta);
        for(int i=1;i<=ncol;i++){
            citra10[k][i]=0;
            for(int g=-2;g<=2;g++){
                a=float(i+g-pusat);
                for(int j=1;j<=ncol;j++){
                    b=float(pusat-j);
                    r2dalam=(a*a+b*b);
                    if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){
                        bagian=HitungLuasPixel1(a,b,sinteta,costeta,n
                        col);
                        citra10[k][i]+=SigmaLuas2(bagian);
                    }
                }
            }
        }
    }
    Application->ProcessMessages();
    CG->AddProgress(1);
}
}

```



```

lbKeterangan->Caption="";
CG->Progress=0;
}
//-----
void __fastcall TMukaUtama::ART5Click()
{
float a,b,r2dalam,sinteta,costeta;
int nteta,pusat;
TBagianPixel bagian;
float dteta,teta;
for(int
baris_ke=0;baris_ke<=(nray+1);baris_ke++){

citra11[baris_ke]=(float*)fcalloc(ncol+2,size
of(float));

citra13[baris_ke]=(float*)fcalloc(ncol+2,size
of(float));
}
nteta=nray;
dteta=pi/nteta;
pusat=(ncol+1)/2;
CG->MaxValue=nray;
lbKeterangan->Caption="Sedang inisialisasi
ART...";
for(int k=1;k<=nteta;k++){
teta=(float)(k-1)*dteta;
sinteta=sin(teta);
costeta=cos(teta);
for(int i=1;i<=ncol;i++){
citra13[k][i]=0;
for(int g=-2;g<=2;g++){
a=float(i+g-pusat);
for(int j=1;j<=ncol;j++){
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){

bagian=HitungLuasPixel2(a,b,sinteta,costeta,n
col,pusat,citra11);
citra13[k][i]+=SigmaLuas1(bagian);
}
}
}
for(int g=-2;g<=2;g++){
a=float(i+g-pusat);
for(int j=1;j<=ncol;j++){
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){

bagian=HitungLuasPixel1(a,b,sinteta,costeta,n
col);

HitungART(citra11,citra3,citra13,citra10,bagia
n,pusat,k,i);
}
}
}
}

Application->ProcessMessages();
CG->AddProgress(1);
} status4=TampilCitra(citra11,Image4);
lbKeterangan->Caption="";
CG->Progress=0;
iterasike=1;
}
//-----
void __fastcall
TMukaUtama::iterasiW51Click(TObject
*Sender)
{
status4=0;
float a,b,r2dalam,sinteta,costeta,gamma;
TBagianPixel bagian;
int nteta,pusat;
float dteta,teta;
for(int
baris_ke=0;baris_ke<=nray;baris_ke++){

citra7[baris_ke]=(float*)fcalloc(ncol+2,sizeo
f(float));

citra16[baris_ke]=(float*)fcalloc(ncol+2,size
of(float));
}
discrepancy=0.0;edistance=0.0;
gamma=1;
for(int i=1;i<=ncol;i++){
for(int j=1;j<=ncol;j++){
citra7[i][j]=citra11[i][j];
}
}

nteta=nray;
dteta=pi/nteta;
pusat=(ncol+1)/2;
CG->MaxValue=nray;
lbKeterangan->Caption="Sedang melakukan
iterasi ART ke "+IntToStr(iterasike)+" ...";
for(int k=1;k<=nteta;k++){
teta=(float)(k-1)*dteta;
sinteta=sin(teta);
costeta=cos(teta);
for(int i=1;i<=ncol;i++){
citra16[k][i]=0;
for(int g=-2;g<=2;g++){
a=float(i+g-pusat);
for(int j=1;j<=ncol;j++){
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){

bagian=HitungLuasPixel2(a,b,sinteta,costeta,n
col,pusat,citra7);
citra16[k][i]+=SigmaLuas1(bagian);
}
}
}
}
}
}
}

```

```

for(int g=-2;g<=2;g++){
a=float(i+g-pusat);
for(int j=1;j<=ncol;j++){
b=float(pusat-j);
r2dalam=(a*a+b*b);
if(r2dalam<=(ncol-0.5)*(ncol-0.5)/4){

FileAsli=SimpanFile->Files-
>Strings[0].c_str();
save=savedata(citra7,FileAsli.c_str());
}
}
//-----
}

bagian=HitungLuasPixel1(a,b,sinteta,costeta,ncol);

HitungART(citra7,citra3,citra16,citra10,bagian,
pusat,k,i,gamma,1);
}
}
}
discrepancy+=(citra3[k][i]-citra16[k][i])*
(citra3[k][i]-
citra16[k][i])/citra10[k][i];
}
Application->ProcessMessages();
CG->AddProgress(1);
}
for(int i=1;i<=ncol;i++){
for(int j=1;j<=ncol;j++){
edistance+=(citra11[i][j]-
citra7[i][j])*(citra11[i][j]-citra7[i][j]);
citra11[i][j]=citra7[i][j];
}
}
status4=TampilCitra(citra7,Image4);
SG->Cells[0][iterasike]=iterasike;
SG->Cells[1][iterasike]=sqrt(discrepancy/((nteta*ncol)));
SG->Cells[2][iterasike]=sqrt(edistance/(ncol*ncol));
}
if((SG->RowCount-11)>=1)
SG->TopRow=SG->RowCount-11;
int bar=SG->RowCount;
bar++;
SG->RowCount=bar;
lbKeterangan->Caption="";
CG->Progress=0;
iterasike++;
}
//-----
}

void __fastcall
TMukaUtama::saveClick(TObject *Sender)
{
// if (status4!=1)Application-
>MessageBox("Maaf, tidak ada data yang
dimasukkan!",
//
"Warning",MB_OK|MB_ICONWARNING);
//else
if(SimpanFile->Execute()){

void __fastcall
TMukaUtama::LebarRadiasi1Click(TObject
*Sender)
{
if(status1)
frPengaturan->ShowModal();
else Application->MessageBox("Maaf,
Silahkan buka data sinogram dulu",
"Peringatan",MB_OK|MB_ICONWARNING);
}
//-----
}

void __fastcall
TMukaUtama::PembuatProgram1Click(TObje
ct *Sender)
{
frInfo->ShowModal();
}
//-----
}

void __fastcall
TMukaUtama::Iterasi3Click(TObject *Sender)
{
ResetLaporan();
frIterasi->ShowModal();
}
//-----
}

void __fastcall
TMukaUtama::Iterasi4Click(TObject *Sender)
{
stop=0;
switch(iterasi)
{
case 1:
switch(LebarRadiasi)
{
case 1:
iterasiw1Click(MukaUtama);
}
}
}
}

```

```

        break;
    case 3:
        IterasiW3Click(MukaUtama);
        break;
    case 5:
        iterasiW51Click(MukaUtama);
        break;
    }
    break;
case 2:
    switch(LebarRadiasi)
    {
    case 1:for(int i=1;i<=50;i++){
        iterasiw1Click(MukaUtama);
        if(stop)break;
        }
        break;
    case 3:for(int i=1;i<=50;i++){
        IterasiW3Click(MukaUtama);
        if(stop)break;
        }
        break;
    case 5:for(int i=1;i<=50;i++){
        iterasiW51Click(MukaUtama);
        if(stop)break;
        }
        break;
    }
    break;
case 3:
    switch(LebarRadiasi)
    {
    case 1:do{
        iterasiw1Click(MukaUtama);
        if(stop)break;
        }while(discripancy>ndiscry);
        break;
    case 3:do{
        IterasiW3Click(MukaUtama);
        if(stop)break;
        }while(discripancy>ndiscry);
        break;
    case 5:do{
        iterasiW51Click(MukaUtama);
        if(stop)break;
        }while(discripancy>ndiscry);
        break;
    }
    break;
}
}
}
//-----
void __fastcall TMukaUtama::ResetLaporan()
{
    SG->ColWidths[0]=70;
    SG->ColWidths[1]=180;
    SG->ColWidths[2]=180;
    SG->Cells[0][0]="Iterasi";
    SG->Cells[1][0]="Discripancy";
    SG->Cells[2][0]="Euclidean Distance";
    for(int i=1;i<=(SG->RowCount-1);i++)
    for(int j=0;j<=2;j++)
    SG->Cells[j][i]="";
    SG->RowCount=2;
}

void __fastcall
TMukaUtama::FormActivate(TObject
*Sender)
{
    ResetLaporan();
}
//-----

void __fastcall
TMukaUtama::Tutup1Click(TObject *Sender)
{
    Application->Terminate();
}
//-----

void __fastcall
TMukaUtama::SimpanLaporan1Click(TObject
*Sender)
{
    if(SimpanLap->Execute()){
        FileAsli=SimpanLap->Files-
>Strings[0].c_str();
        FILE *pfp;
        pfp=fopen(FileAsli.c_str(), "w+");
        fprintf(pfp,"Iterasi\tDiscripancy\tEuclidean
Distance\n");
        for(int i=1;i<=(SG->RowCount-2);i++){
            for(int j=0;j<=2;j++)
                fprintf(pfp,"%s\t",SG->Cells[j][i]);
            fprintf(pfp,"\n");
        }
        fclose(pfp);
    }
}
//-----

void __fastcall
TMukaUtama::Stop1Click(TObject *Sender)
{
    stop=1;
}
//-----

```

LAMPIRAN C**Laporan Nilai *Discrepancy* Segitiga Lebar Kolimator $w = \Delta x$**

Iterasi	<i>Discrepancy</i>	<i>Euclidean Distance</i>
1	4,02460426049457	11,6742679889176
2	2,2257227481585	6,64941282591041
3	1,75407331858486	5,41027720336018
4	1,43527106145982	4,4433392059381
5	1,13235323287437	3,64676325719322
6	0,970364540392984	3,01814282189763
7	0,781160989719813	2,47406558272533
8	0,657169310934112	2,03745613586649
9	0,555842289027352	1,66189232560272
10	0,44970954209114	1,34690701361193
11	0,38507742503146	1,09702073670134
12	0,300100964890553	0,891695074487468
13	0,260825081269288	0,742848238868991
14	0,215679616563727	0,613616475358678
15	0,189358706700337	0,511467277670077
16	0,167979063722622	0,416663322440828
17	0,141173851068635	0,337695426348644
18	0,12574069841392	0,276310142384302
19	0,10654954743186	0,229030548702757

Laporan Nilai *Discrepancy* Segitiga Lebar Kolimator $w = 3\Delta x$

Iterasi	<i>Discrepancy</i>	<i>Euclidean Distance</i>
1	2,74049194865728	12,2176569784179
2	1,55857890048149	8,29777998661723
3	1,42777185515509	7,72113142497529
4	1,34089869606597	7,23547099911824
5	1,26165952549083	6,77365408326845
6	1,20818445219434	6,34619139116449
7	1,14818911699226	5,91220519631555
8	1,07251615135794	5,50343130583205
9	1,02166617628	5,10784934490534
10	0,954337623941472	4,73064364758947
11	0,903670423718917	4,38541436482555
12	0,842427099274036	4,05256015951493
13	0,777454966706754	3,7623931087745
14	0,739279052654792	3,46454162369964
15	0,682140685221679	3,19336103132097
16	0,641815819573553	2,93633392712986
17	0,587064507257186	2,70292525892965
18	0,539661938500909	2,50497269566485
19	0,502138922609094	2,31933802366812
20	0,463726658734289	2,15789382280767
21	0,436207347031844	2,00723065094044
22	0,405854878660783	1,87044605023089
23	0,381767315406742	1,7442749991015
24	0,353414630459784	1,6261100341475
25	0,330523852290552	1,51811166882202
26	0,307371197153445	1,41737700164428
27	0,289059519082289	1,3264124588255
28	0,272843391830194	1,23907398808644
29	0,257566036759628	1,15783376730883
30	0,245134304781272	1,07990563298019
31	0,231769869856898	1,0057233858365
32	0,220429719356696	0,937240403122591
33	0,208710160965697	0,871185215689944
34	0,197263584769433	0,810787304435236
35	0,187388858123644	0,753807179193494
36	0,176366136765409	0,701348486559793
37	0,16896470566589	0,652939798895024
38	0,158451764802573	0,607291590174585
39	0,152170388376485	0,565699327292021
40	0,142454928560077	0,526437495896117
41	0,136518197380171	0,490742565249782
42	0,128240889580459	0,45736124240465
43	0,123435858664612	0,42653788287711
44	0,116767435492078	0,397433876906388
45	0,112848355621484	0,369984102602823
46	0,107372124127574	0,343712954250735
47	0,10356241510686	0,318794454066255

Laporan Nilai *Discrepancy* Segitiga Lebar Kolimator $w = 5\Delta x$

Iterasi	<i>Discrepancy</i>	<i>Euclidean Distance</i>
1	3,10349356111177	12,5876522591776
2	1,61009921710888	8,5621005961816
3	1,4470315188294	8,02408687309281
4	1,34768635779956	7,57994928487403
5	1,27102070877306	7,16116192139796
6	1,20455245960598	6,77981476935592
7	1,16079109830854	6,39549237194802
8	1,07669463625991	6,02895683942431
9	1,02925591836595	5,68119427272442
10	0,95588757525656	5,35020865242331
11	0,913257302364519	5,04890023745639
12	0,873556784785927	4,752891507278
13	0,81472824073399	4,48645476061585
14	0,802973038848734	4,20918895804683
15	0,749569991425254	3,94266522662249
16	0,723366792152587	3,68420815651471
17	0,664041661807362	3,44338733046148
18	0,629764101774244	3,23468191366387
19	0,59162657505165	3,03434011185955
20	0,554217224586806	2,85608536211668
21	0,528928345525601	2,68557464804972
22	0,492715791307613	2,52932141305897
23	0,470758566544599	2,38645199832435
24	0,4382894721258	2,25195355513044
25	0,420050413775348	2,130934720936
26	0,397563936181307	2,01322424581968
27	0,378677841656726	1,90415664814836
28	0,363195642690873	1,79741003785396
29	0,343990951087132	1,69739053303718
30	0,329475345971943	1,60284654914596
31	0,314746600894665	1,51230330975142
32	0,302961997414463	1,4279946002479
33	0,290372029243104	1,3438522557615
34	0,279863635208191	1,26210095884953
35	0,267658629783049	1,18313872599412
36	0,254938950389269	1,10927784084158
37	0,242418680056685	1,04169941642568
38	0,230007998191443	0,980493902009191
39	0,218175231062298	0,924725382207977
40	0,210026466581277	0,872111291717167
41	0,200899852501781	0,822336666280642
42	0,193741419375378	0,774322547866967
43	0,186819613289967	0,72853987310026
44	0,179037565782838	0,684399829236812
45	0,173903932337003	0,642298601169064
46	0,166121738648685	0,601375460909886
47	0,161367439728599	0,563496420672855
48	0,154789530962348	0,526531481632855
49	0,150060592165581	0,491661444135547
50	0,144851593898445	0,458122656592321
51	0,139974273714984	0,426126810901285
52	0,134697940455124	0,396154425098464

53	0.130269238320857	0.368960605639487
54	0.125407237690782	0.343464705258449
55	0.121872883336417	0.320270291532491
56	0.117334760474717	0.298647952076749
57	0.11433980242028	0.278533742636492
58	0.110415799015111	0.259802205101538
59	0.107627215622682	0.24204650555168
60	0.104554635155001	0.225527479664201
61	0.101553579346789	0.210156946041712

Laporan Nilai *Discrepancy* Modul Lebar Kolimator $w = \Delta x$

<i>Iterasi</i>	<i>Discrepancy</i>	<i>Euclidean Distance</i>
1	9,21277583611681	18,084246157358
2	4,33033365331514	9,59211551313682
3	2,89985514832952	7,09762404805539
4	2,21546860648161	5,53108658500639
5	1,62098930948536	4,4333737361749
6	1,36627578116952	3,66442612489154
7	1,09625565867185	3,04894428423553
8	0,945644723717628	2,55588471462786
9	0,817701957327643	2,12636162898096
10	0,696540930402857	1,75249953245621
11	0,623224208142215	1,44450662849338
12	0,521340256022361	1,1808579656101
13	0,469785965172496	0,983075607301989
14	0,405279549631534	0,814744924784644
15	0,370733105880702	0,681940443129868
16	0,336565170477783	0,561040970299187
17	0,305430892243108	0,457815323423775
18	0,283665820157409	0,371989365153537
19	0,258178829911969	0,302125626973038
20	0,243362409102707	0,250486815813666
21	0,227147205727114	0,20743029054247
22	0,215615595026708	0,172719624882093
23	0,205309312690979	0,141058125303802
24	0,195044399375504	0,114368699160455
25	0,187206223747303	0,0930188187540759
26	0,178946404930905	0,0764554290752405
27	0,17255807594649	0,0642111156572004
28	0,166194749179792	0,0543626759360656
29	0,160630748293918	0,0462936786333944
30	0,155514608705044	0,0393791563727564
31	0,150555346438782	0,0335777791863473
32	0,146213707299767	0,0290029882349641
33	0,141879827027674	0,0253765031434478
34	0,138042484369509	0,0226332879729694
35	0,134304543143206	0,0204166336203881
36	0,130835866277028	0,0186315393423055
37	0,127575964076812	0,0171462414715523
38	0,124433574695696	0,01591737160798
39	0,121527736722888	0,0149274083704982
40	0,118700542248553	0,0141043236289021
41	0,11606427020577	0,013424391627131
42	0,113523037470412	0,0128347439700203
43	0,111111473981586	0,0123124443448142
44	0,108810977334765	0,0118482221709932
45	0,106601138028517	0,0114218817461427
46	0,104499775429818	0,0110375801810251
47	0,102473852159338	0,0106794572285215
48	0,100538591672174	0,0103497086105938

Laporan Nilai *Discrepancy* Modul Lebar Kolimator $w = 3\Delta x$

Iterasi	<i>Discrepancy</i>	<i>Euclidean Distance</i>
1	4,75766062741651	18,6772558123448
2	3,29896947761763	12,8595761686534
3	2,81987063001815	11,1851585467325
4	2,52218948861338	10,081793567373
5	2,22568087117461	9,1285379393641
6	2,02293219615139	8,31704305005151
7	1,83078535305146	7,58012028542075
8	1,67636446506503	6,9378909633016
9	1,54582533621516	6,32553187682325
10	1,40969968126635	5,76858281366363
11	1,31861896603639	5,26756738751697
12	1,23005619047534	4,79825429025895
13	1,14592131611628	4,38504477824747
14	1,06151737707711	3,98799999163589
15	0,982803935808033	3,64135798246697
16	0,914336953504051	3,30262180447165
17	0,839694713901241	3,00431204509148
18	0,788788492326643	2,7358618873685
19	0,724842844620486	2,49649676659458
20	0,683963918655864	2,28675710390634
21	0,633623793734107	2,09428790632608
22	0,595291158860357	1,92441227038612
23	0,562482584076694	1,76279239042748
24	0,524708890476366	1,61864238715856
25	0,503012266004374	1,48573054708467
26	0,469774701686815	1,36671758977813
27	0,448860301397	1,25839448148011
28	0,425518042173459	1,15725442131931
29	0,405768606186498	1,06718237389993
30	0,389344928482571	0,982292741196551
31	0,370017811638333	0,906918492928006
32	0,356147436641871	0,835312391849875
33	0,342604755123006	0,76731598708226
34	0,330004149180052	0,702783367362757
35	0,317913060111713	0,642352731206734
36	0,305194316531535	0,589090091952908
37	0,294027647489389	0,540066704452471
38	0,283414014319903	0,49641402032226
39	0,273745207412836	0,455203999348561
40	0,265815948181881	0,41701113154414
41	0,256911018377929	0,380948129349224
42	0,250264564139525	0,347355478888704
43	0,24210046232688	0,317640614061357
44	0,235878195628817	0,289857060959601
45	0,229286002413585	0,265092536027479
46	0,223560175401631	0,241519773496444
47	0,217945888020368	0,219916183702304
48	0,212871226931358	0,199735069165637
49	0,207617612485603	0,181342926674006
50	0,203117089312704	0,16447750625898
51	0,198446182126991	0,148641670643467
52	0,194019039564188	0,134365990023303

53	0.190057136669118	0.121142435622887
54	0.185819315496014	0.109273406855502
55	0.182147646946692	0.0985727336805543
56	0.178523589896815	0.0890756873278878
57	0.174964898993731	0.0806161178859713
58	0.171712625381524	0.07309540002437
59	0.168451040033265	0.0664489148033916
60	0.165402591470355	0.0605073561646771
61	0.162491975450138	0.0552084668126425
62	0.159590671213655	0.0505769516913253
63	0.156938415206572	0.0464294718360096
64	0.154252054059217	0.0427727015573725
65	0.151730244976393	0.0394464158223702
66	0.14928165252702	0.0364676462901098
67	0.14686317330968	0.0337471666684809
68	0.144604305628331	0.03127845832085
69	0.142314371778217	0.0290014051536919
70	0.140198796856257	0.0269387192574887
71	0.138044673462904	0.0250681850593091
72	0.136058149955926	0.0233871123618092
73	0.134023640766191	0.0218691255579422
74	0.132136041271824	0.0205017439281432
75	0.13024397962616	0.0192726438523362
76	0.128426449345063	0.0181629649060491
77	0.126665830395882	0.017170703693453
78	0.124920220577559	0.0162642865616102
79	0.123276419055406	0.0154508188576495
80	0.121615849087807	0.0147082694474698
81	0.120043032943176	0.0140394897349707
82	0.118479903886711	0.0134325358565824
83	0.116978704148133	0.0128811169682191
84	0.115492848043955	0.0123777945591827
85	0.11406149601295	0.0119203255819916
86	0.112645043491472	0.0115037487815021
87	0.11127991877868	0.0111220686155226
88	0.109930217315964	0.0107704162244472
89	0.108626443508265	0.0104465770250678
90	0.107336875922137	0.0101467970965812
91	0.106095834245192	0.0098718663748591
92	0.10486313625755	0.0096151884223767
93	0.103671774754095	0.00937601768615848
94	0.102497742976904	0.009153636857287
95	0.101355138692823	0.0089421852979077
96	0.100234866581866	0.00874552523219553

Laporan Nilai *Discrepancy* Modul Lebar Kolimator $w = 5\Delta x$

Iterasi	<i>Discrepancy</i>	<i>Euclidean Distance</i>
1	4,9629710840614	18,6922707502853
2	3,48975563994801	12,9503354366741
3	3,00830400722412	11,2683252282807
4	2,68873962419613	10,1569721535587
5	2,34554010815209	9,27042429619774
6	2,10469298222126	8,56602579984417
7	1,93413582195096	7,92328605642406
8	1,79112052820576	7,33966611920443
9	1,6594750592767	6,78273219366002
10	1,5193993326806	6,2712735475136
11	1,45090322239671	5,80026290663618
12	1,36254469010744	5,33642065946557
13	1,27172168550941	4,94059371497318
14	1,18799365856159	4,57318778486003
15	1,10998474763955	4,24666960757551
16	1,0572536031698	3,9318596999755
17	0,994076009299277	3,64366569234876
18	0,944558550543289	3,37567490146224
19	0,883567860445461	3,13150309158768
20	0,829720164875672	2,9161834500768
21	0,796714024234452	2,70950489221897
22	0,757060725986605	2,50987368747052
23	0,726628840038594	2,32563890515975
24	0,687609606789941	2,1614586690895
25	0,661393291653186	2,00003451149001
26	0,631799832773352	1,84829030276404
27	0,601494431794144	1,70710977889461
28	0,575499589637897	1,57891763807895
29	0,549357901973929	1,4684458010014
30	0,527709931537851	1,36131538930979
31	0,507498978807558	1,26365883786371
32	0,487776389609205	1,17122019262273
33	0,467279912447497	1,08692371382785
34	0,451219985327359	1,01106388196127
35	0,432844483283511	0,943367484878002
36	0,421026573551224	0,881380395356654
37	0,404831682093708	0,821875773159453
38	0,393106517259735	0,766893044612405
39	0,381022933770645	0,715075735725151
40	0,368791516843378	0,668833771877431
41	0,359075198428638	0,625217810138941
42	0,349311334753968	0,583937083856245
43	0,341139849618574	0,543777445051921
44	0,331503811444614	0,503722498136583
45	0,323317502970184	0,467118997249867
46	0,314944528835497	0,43224629440318
47	0,30766035041348	0,399577036493775
48	0,300721307933791	0,369596463933722
49	0,293788399551916	0,341140304870197
50	0,288266662310702	0,314350588679723
51	0,281480158373343	0,289034647039894
52	0,27601812683257	0,266584198808166

53	0.27000778740915	0.246220153120005
54	0.26449021841935	0.228508284205686
55	0.259708141220403	0.21280271205159
56	0.254540658915535	0.198165494368386
57	0.250130686806484	0.184511951185651
58	0.245685078566969	0.171805907494386
59	0.241394082380722	0.159948575018634
60	0.237359580925989	0.14862813885879
61	0.233621733614654	0.137813310333351
62	0.229595023040829	0.127216507873011
63	0.226237428933911	0.117065280683552
64	0.222601101927705	0.107387244242069
65	0.219146142701184	0.0983879102832396
66	0.215918763847568	0.0901955238247643
67	0.21250551322714	0.0825276777328194
68	0.209546334213481	0.0756626133586319
69	0.206292913926192	0.0695103488927839
70	0.203451841180695	0.0640502133932358
71	0.200444214521515	0.0590085263823299
72	0.197680359573858	0.0545123272864336
73	0.194811492050788	0.0505366284701517
74	0.192134610750414	0.0469946118730635
75	0.1895153824098	0.0437535474236132
76	0.186850181856836	0.0408566607558775
77	0.184390098175203	0.0382273371338099
78	0.181896165383092	0.0358505683592164
79	0.17945946066016	0.0337182203120879
80	0.177136130591678	0.0318585231386403
81	0.174784738914539	0.0301906493303791
82	0.172574616969539	0.0286666376618394
83	0.170335258052103	0.0272610008454423
84	0.168215026222634	0.0259505344645558
85	0.16607097172338	0.0247281256133388
86	0.164058350075623	0.0236053144709605
87	0.162011156431972	0.0225703191407617
88	0.160049092030931	0.0216477713404685
89	0.158127379170644	0.0207936835189244
90	0.156221134641852	0.0200387792643437
91	0.154390990914126	0.0193392662891338
92	0.152551262160135	0.0186965928193229
93	0.150795708743185	0.0180970199561172
94	0.149051086797995	0.0175396237198717
95	0.147341393139327	0.0170305991865127
96	0.145687784944929	0.0165633953887576
97	0.144019286439195	0.0161333625417574
98	0.14243285335542	0.0157331326737241
99	0.140840636022647	0.0153564885219618
100	0.13932363530673	0.0150026299404179
101	0.137794500885518	0.0146630830562752
102	0.13633599103859	0.0143495567660085
103	0.134872133437667	0.0140506863973988
104	0.13345568828884	0.0137742470105326
105	0.132051566539514	0.0135053250621446
106	0.130678016449238	0.0132534676233499
107	0.12933327204488	0.0130106551263249

108	0.128017861880572	0.0127811706498435
109	0.126716294311124	0.0125592421219103
110	0.125445486607704	0.0123446933081899
111	0.124200026753079	0.0121385104426293
112	0.122970470704542	0.0119374992751003
113	0.121745976668929	0.011741374628683
114	0.120560058831423	0.0115522006531842
115	0.119401197534513	0.0113653784666872
116	0.118253378776475	0.0111862871769132
117	0.117133169207116	0.0110097957996288
118	0.116020852298152	0.0108434549268086
119	0.114933870105382	0.0106769896942859
120	0.113865876577763	0.0105172257459801
121	0.112808336805043	0.0103567196623351
122	0.111775885896377	0.0102052649860938
123	0.110748428474479	0.0100556817527649
124	0.109748820543029	0.00990836764734326
125	0.108753577798257	0.00976593908249147
126	0.107779346775661	0.00962621033875388
127	0.106805041157549	0.00948931470020771
128	0.105873876890556	0.00935694263219208
129	0.104940579527435	0.00922620030093954
130	0.104016899808695	0.00909870149840829
131	0.103106880495184	0.00897318762056189
132	0.102217446435091	0.0088508266825204
133	0.101327594995816	0.00873109521149148
134	0.100461206995341	0.00861382227876847

LAMPIRAN D**Objek sintetis : Segitiga.dat**

No.	Lebar Kolimator yang Digunakan	Metode	Rmsd %
1.	1	ART 19 kali	0,767
2.	3	ART 47 kali	0,745
3.	5	ART 61 kali	0,721
4.	1	FBP	2,62
5.	3	FBP	4,713
6.	5	FBP	5,145

(a)

Objek sintetis : Modul.dat

No.	Lebar Kolimator yang Digunakan	Metode	Rmsd %
1.	1	ART 47 kali	1,175
2.	3	ART 96 kali	1,171
3.	5	ART 134 kali	1,142
4.	1	FBP	8,025
5.	3	FBP	12,501
6.	5	FBP	13,356

(b)

Tabel nilai rmsd (a) objek Segitiga.dat , (b) objek Modul.dat