



## **Kata Pengantar**

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan akhir berjudul “GUI untuk Rekonstruksi Gambar CT *Scan* dengan *Filtered Back Projection* dan *Algebraic Reconstruction Technique*” ini. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan mata kuliah Teknologi Pencitraan Medis tahun 2023.

Dalam penyusunan laporan ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan banyak terima kasih kepada Bapak Dr. Basari, S.T., M.Eng selaku dosen mata kuliah Teknologi Pencitraan Medis yang telah membimbing penulis selama pembuatan tugas akhir ini. Penulis juga mengucapkan terima kasih kepada teman-teman kelas Teknologi Pencitraan Medis tahun 2023 serta pihak-pihak lain yang sudah

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari pembaca. Semoga laporan ini dapat memberikan hal yang bermanfaat bagi para pembaca terkait rekonstruksi gambar pada CT *scan*.

Depok, 27 Desember 2023

Penulis

Kelompok 1

## **Abstrak**

Penelitian ini membangun sistem rekonstruksi gambar CT scan dengan GUI. Sistem ini mampu merekonstruksi gambar dari phantom dengan empat jenis metode rekonstruksi, yaitu Simple Back Projection (SBP), Filtered Back Projection (FBP), dan Algebraic Reconstruction Technique (ART).

Simulasi dilakukan untuk menerapkan setiap metode rekonstruksi pada setiap jenis phantom. Hasil simulasi menunjukkan bahwa metode FBP dengan filter Hamming dan Hanning menghasilkan kualitas gambar yang terbaik, dilihat dari nilai PSNR dan SSIM. Sementara itu, metode ART menghasilkan kualitas gambar yang lebih baik dengan jumlah iterasi yang lebih banyak, namun nilai SSIM nya masih lebih rendah dibandingkan metode FBP.

## *Abstract*

*This research builds a CT scan image reconstruction system with GUI. The system is able to reconstruct images from phantoms with four types of reconstruction methods, namely Simple Back Projection (SBP), Filtered Back Projection (FBP), and Algebraic Reconstruction Technique (ART).*

*Simulations were conducted to apply each reconstruction method to each type of phantom. The simulation results show that the FBP method with Hamming and Hanning filters produces the best image quality, judging from the PSNR and SSIM values. Meanwhile, the ART method produces better image quality with more iterations, but the SSIM value is still lower than the FBP method.*

## Daftar Isi

<b>Kata Pengantar.....</b>	<b>2</b>
<b>Abstrak.....</b>	<b>3</b>
<b>Daftar Isi.....</b>	<b>4</b>
<b>BAB 1. PENDAHULUAN.....</b>	<b>7</b>
1.1. Latar Belakang.....	7
1.2. Rumusan Masalah.....	7
1.3. Tujuan Laporan.....	8
1.4. Batasan Masalah.....	8
1.5. Sistematika Laporan.....	8
<b>BAB II. TINJAUAN PUSTAKA.....</b>	<b>11</b>
2.1. Image Reconstruction in CT Scan.....	11
2.2. Transformasi radon dan inverse radon.....	13
2.3. Filtered back projection (FBP).....	14
2.4. Algebraic reconstruction technique (ART).....	15
2.5. Filters.....	16
2.6. Image Evaluation.....	16
2.5.1 PSNR.....	17
2.5.2 SSIM.....	18
2.7. State of the Art.....	18
<b>BAB III. METODOLOGI.....</b>	<b>21</b>
<b>BAB IV. PENUTUP.....</b>	<b>30</b>
4.1. Kesimpulan.....	30
4.2. Saran dan Studi Lanjutan.....	30
<b>Referensi.....</b>	<b>32</b>
<b>Lampiran.....</b>	<b>35</b>

## Daftar Gambar

**Gambar 1.** EMI scanner pertama

**Gambar 2.** Geometri pancaran parallel dengan sistem koordinat. X-ray membuat sebuah sudut pada sumbu-y dengan jarak  $r$ . Profil intensitas dihitung untuk setiap sudut. Profil atenuasi didapatkan menggunakan log dari profil intensitas dan merupakan proyeksi dari fungsi  $\mu(x, y)$  pada sudut

**Gambar 3.** Sebuah sinogram untuk dataset 2D  $p(r, \theta)$  dengan menumpuk proyeksi  $p_\theta(r)$

**Gambar 4.** Mapping dari radon transform  $f$  pada sumbu  $x$  dan  $y$ - domain  $Rf$  pada  $(\alpha, s)$

**Gambar 5.** Penggunaan PSNR dan SSIM dalam mengevaluasi hasil gambar

**Gambar 6.** (a) Phantom 1. (b) Phantom 2. (c) Phantom 3. (d) Phantom 4.

**Gambar 7.** GUI sistem rekonstruksi gambar CT scan dan keterangan fiturnya.

**Gambar 8.** Skor PSNR dan SSIM dari metode FBP, PSNR dan SSIM dari metode ART pada Phantom 1.

**Gambar 9.** Skor PSNR dan SSIM dari metode FBP, PSNR dan SSIM dari metode ART pada Phantom 2.

**Gambar 10.** Skor PSNR dan SSIM dari metode FBP, PSNR dan SSIM dari metode ART pada Phantom 3.

## Daftar Tabel

**Tabel 1.** Tabel SOTA CT Image Reconstruction

## BAB 1. PENDAHULUAN

### 1.1. Latar Belakang

Pada awal tahun 1930, seorang radiologis Italia Alessandro Vallebona membuat sebuah metode untuk merepresentasikan sebuah *single slice* dari sebuah badan pada sebuah film radiografik. Metode ini diketahui dengan nama tomography (tomo yang berarti *slice*). CT scan komersil yang pertama kali digunakan diciptakan oleh Sir Godfrey Newbold Hounsfield di Hayes, Inggris di EMI (Electric and Musical Industries) laboratorium riset utama menggunakan x-ray. Hounsfield menemukan idenya pada tahun 1967 dan mengumumkannya pada tahun 1972. Allan Mcleod Cornmack dari Universitas Tufts, Massachusetts, Amerika Serikat menemukan hal yang serupa. sehingga pada tahun 1979, Cornmack dan Hounsfield berbagi sebuah penghargaan nobel di bidang kesehatan.



**Gambar 1.** EMI scanner pertama

Mesin CT x-ray produksi pertama hanya terbatas untuk tomografi otak, dan membutuhkan waktu sekitar 4 menit. Mesin ini beroperasi dengan cara memindai dua irisan yang berdekatan, dan waktu komputasi 7 menit per gambar dengan menggunakan komputer mini data general nova. Pemindai EMI memerlukan tangki Perspex berisi air dengan karet yang sudah dibentuk sebelumnya di bagian depan, yang menutup kepala pasien. Tangki air digunakan untuk mengurangi rentang dinamis radiasi yang mencapai detektor. Gambar-gambar yang dihasilkan memiliki resolusi yang relatif rendah, karena hanya terdiri dari matriks  $80 \times 80$  piksel.

### 1.2. Rumusan Masalah

Pertanyaan pada makalah ini, antara lain :

1. Bagaimana cara mengembangkan sebuah algoritme tentang image reconstruction dari sebuah generasi tiga proyeksi CT berdasarkan dari sebuah persamaan linear (FBP), dan bagaimana menciptakan program yang efisien?
2. Bagaimana perbedaan kualitas dari sebuah gambar yang direkonstruksikan dari metode Algebraic Reconstruction Technique (ART) dari metode Filtered Back Projection (FBP) yang lebih komersil?
3. Bagaimana efisiensi penggunaan filter yang berbeda pada sebuah gambar yang sudah diberikan *blur* dan *noise*?
4. Bagaimana perbedaan dari penggunaan filter Hamming, Hamming, Cosine, dan Ramlak?

### **1.3. Tujuan Laporan**

1. Dapat mengembangkan sebuah program yang dapat menerapkan Image Reconstruction berdasarkan algoritma Filter Back Projection (FBP)
2. Dapat mengetahui perbedaan kualitas dari gambar yang telah direkonstruksikan dengan menggunakan dua algoritma, yaitu Filter Back Projection (FBP) dan Algebraic Reconstruction Technique (ART)
3. Dapat mengetahui seberapa efisien penggunaan keempat Filters terhadap sebuah gambar yang telah dimodifikasi dengan menambahkan *noise* ataupun *blur*
4. Dapat menjelaskan perbedaan dari hasil gambar yang diberikan Filters Ramlak, Hamming, Hamming, Cosine.

### **1.4. Batasan Masalah**

1. Penelitian hanya berfokus pada proses rekonstruksi gambar pada modalitas CT 'CT Image Reconstruction'
2. Penelitian hanya menggunakan bahasa pemrograman MATLAB
3. Penelitian menerapkan dua algoritma, yaitu Filter Back Projection (FBP) dan Algebraic Reconstruction Technique (ART)
4. Penelitian juga menerapkan empat jenis Filters, berupa Ramlak, Hamming, Hamming, Cosine.
5. Hasil penelitian memberikan grafik perbandingan terhadap dua variabel pengukuran, yaitu PSNR dan SSIM.

### **1.5. Sistematika Laporan**

#### *BAB 1 : Pendahuluan*

Merupakan bagian awal laporan yang berisikan :



- Latar belakang dalam memberikan pemahaman mengapa penelitian ini penting untuk dilakukan.
- Rumusan masalah dalam menentukan permasalahan/ topik yang dijadikan fokus penelitian.
- Tujuan penelitian dalam mengidentifikasi dan memaparkan hasil apa yang diinginkan dari penelitian.
- Batasan masalah menetapkan cakupan ruang lingkup permasalahan agar fokus penelitian dapat tergambarkan dengan jelas.

## *BAB 2 : Tinjauan Pustaka*

Menguraikan segala teori dasar yang berkaitan dengan topik penelitian yang diangkat, sebagai berikut :

- Menjelaskan mengenai metode rekonstruksi gambar ‘Image Reconstruction’ khususnya pada modalitas Computed Tomography.
- Memahami konsep dasar Image Reconstruction melalui prinsip transformasi Radon dan inverse Radon.
- Menjelaskan cara kerja algoritma Filter Back Projection (FBP) pada Image Reconstruction.
- Menjelaskan cara kerja Algoritma Algebraic Reconstruction Technique (ART) pada Image Reconstruction.
- Mengetahui secara singkat mengenai Filters pada Image Reconstruction.
- Memahami dua alat ukur dalam melakukan penilaian kualitas gambar ‘Image Evaluation’ berupa PSNR dan SSIM.

## *BAB 3 : Metode*

- Mengetahui mengenai alat dan bahan yang digunakan dalam penelitian.
- Memaparkan langkah pembuatan Phantom sebagai awal dari metode penelitian.
- Memaparkan langkah terbentuknya sinogram dari Phantom dalam rangkaian metode penelitian.
- Memaparkan langkah rekonstruksi gambar dalam rangkaian metode penelitian.

- Memaparkan langkah penghitungan PSNR dan SSIM dalam rangkaian metode penelitian.
- Mensimulasikan program rekonstruksi gambar sebagai akhir dari metode penelitian.
- Menampilkan hasil penelitian berupa grafik perbandingan antara masing-masing kondisi Phantom terhadap variabel berupa PSNR dan SSIM.
- Menganalisa perbedaan yang terjadi terhadap berbagai kondisi Phantom.

#### *BAB 4 : Penutupan*

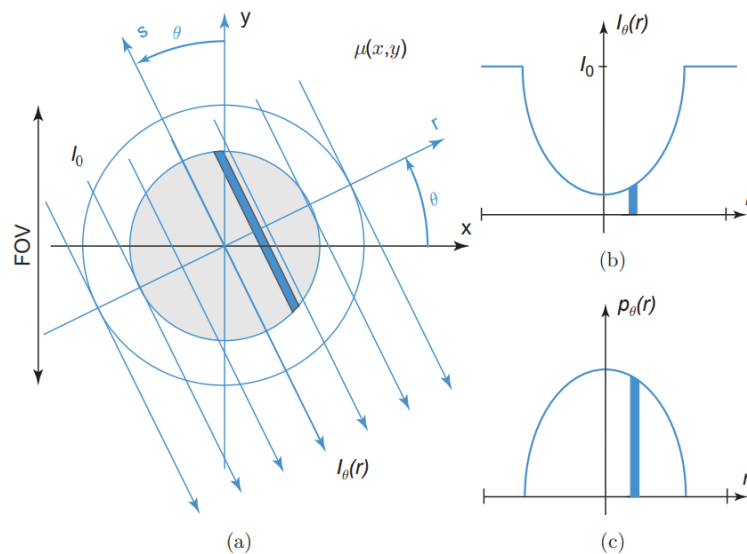
- Memberikan kesimpulan yang menjawab rumusan masalah dari penelitian tersebut.
- Membagikan saran yang dapat diterapkan dalam mengembangkan penelitian berikutnya.

## BAB II. TINJAUAN PUSTAKA

### 2.1. *Image Reconstruction in CT Scan*

*Computed tomography imaging* membutuhkan rekonstruksi dari sebuah volumetrik dataset yang tergantung pada waktu. sekuens dari sebuah CT dihitung dari dinamika contrast agent baik pada bagian vaskulatur dan parenchyma. Rekonstruksi gambar CT mengacu pada proses komputasi untuk menentukan gambar tomografi dari proyeksi x-ray pada pasien. Rekonstruksi gambar adalah tugas yang intensif secara komputasional dan salah satu langkah yang paling penting dalam proses pencitraan CT. Hasil pembentukan gambar sinar-x merupakan gambar atenuasi. Oleh karena itu, setiap piksel pada detektor merepresentasikan sebuah integral garis, yang mengakumulasi semua koefisien atenuasi x-ray disepanjang garis proyeksi. Garis proyeksi adalah garis penghubung titik fokus sinar-x dengan pusat piksel detektor. Terlepas dari sifat diskrit sebuah gambar proyeksi, sebagian besar teori rekonstruksi akan diturunkan secara matematis. pengambilan sampel diskrit kemudian dipecahkan dalam perumusan akhir algoritme rekonstruksi.

Pertama, data diakuisi diubah menjadi geometri balok paralel 2D seperti pada gambar 1. Dimana  $\mu(x, y)$  mewakili distribusi dari koefisien atenuasi linier pada bidang xy. Diasumsikan bahwa pasien berada di sepanjang sumbu-z dan  $\mu(x, y)$  adalah nol di luar bidang pandang melingkar dengan diameter FOV. sinar-X membuat sudut  $\theta$  dengan sumbu y. Intensitas yang tidak dilemahkan dari Berkas sinar-X adalah  $I_0$ . Sistem koordinat baru  $(r, s)$  adalah didefinisikan dengan memutar  $(x, y)$  pada sudut  $\theta$ .



**Gambar 2.** (a) geometri pancaran parallel dengan sistem koordinat. X-ray membuat sebuah sudut pada sumbu-y dengan jarak  $r$ . (b) profil intensitas dihitung untuk setiap sudut ( c) profil atenuasi didapatkan menggunakan log dari profil intensitas dan merupakan proyeksi dari fungsi  $\mu(x, y)$  pada sudut

kemudian angka yang didapatkan dimasukan pada rumus transformasi berikut :

$$\begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}$$

Proyeksi  $p_\theta(r)$  dapat diukur untuk  $\theta$  yang berkisar antara 0 hingga  $2\pi$ . Karena sinar yang datang bersamaan dari sisi yang berlawanan secara teoritis menghasilkan pengukuran yang identik, profil atenuasi yang diperoleh pada sisi yang berlawanan akan membawa hasil yang sama dan excess. Oleh karena itu, cukup mengukur  $p_\theta(r)$  untuk  $\theta$  yang berkisar antara 0 hingga  $\pi$ . Menumpu semua proyeksi  $p_\theta(r)$  menghasilkan sebuah 2D dataset yang disebut dengan sinogram.

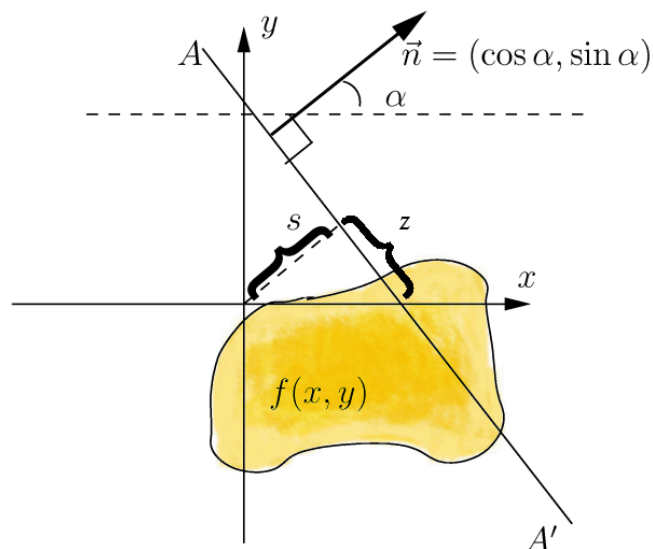


**Gambar 3.** Sebuah sinogram untuk dataset 2D  $p(r, \theta)$  dengan menumpuk proyeksi  $p_\theta(r)$

## 2.2. Transformasi radon dan inverse radon

Prinsip matematika CT pertama kali dikembangkan oleh Radon pada tahun 1917. Transformasi Radon telah membuktikan bahwa gambar suatu objek yang tidak diketahui dapat dihasilkan jika seseorang memiliki jumlah proyeksi yang tidak terbatas melalui objek tersebut, kita dapat memahami ide dasar di balik pencitraan tomografi dengan contoh diambil dari radiografi.

Fungsi proyeksi pada transformasi Radon diartikan sebagai himpunan integral garis lurus ( $y'$ ) yang memproyeksikan luas benda berdasarkan sudut. Setiap integral garis, dalam praktiknya, mewakili sifat fisik strip dengan lebar berhingga yang terutama ditentukan oleh lebar detektor. Pada setiap tampilan, kumpulan data integral garis diperoleh. Kumpulan data proyeksi lengkap dapat diperoleh dengan penilaian berulang terhadap data putaran 180 atau 360° dengan indeks sudut yang ditentukan.



**Gambar 4.** Mapping dari radon transform  $f$  pada sumbu  $x$  dan  $y$ - domain  $Rf$  pada  $(\alpha, s)$

rumus radon transformasi :

$$\begin{aligned} Rf(\alpha, s) &= \int_{-\infty}^{\infty} f(x(z), y(z)) dz \\ &= \int_{-\infty}^{\infty} f((z \sin \alpha + s \cos \alpha), (-z \cos \alpha + s \sin \alpha)) dz. \end{aligned}$$

karena sebuah objek berada pada poin (x,y) pada koordinat, radon transform akan memetakan semua poin yang diproyeksikan oleh  $y'$  terhadap sebuah koordinat baru. hasilnya akan menjadi sebuah grafik sinusoidal yang akan mengkarakteristikan sebuah proyeksi gambar.

### 2.3. Filtered back projection (FBP)

Mengingat sinogram  $p(r, \theta)$ , pertanyaannya adalah bagaimana cara merekonstruksi distribusi  $\mu(x, y)$  (atau, secara umum, fungsi  $f(x, y)$ ). Secara intuitif, kita dapat memikirkan prosedur berikut. Untuk sebuah garis tertentu  $(r, \theta)$ , tetapkan nilai  $p(r, \theta)$  untuk semua titik  $(x, y)$  di sepanjang garis tersebut. Ulangi hal ini (yaitu mengintegrasikan) untuk  $\theta$  mulai dari 0 hingga  $\pi$ . Prosedur ini disebut proyeksi balik dan diberikan oleh rumus :

$$\begin{aligned} b(x, y) &= B\{p(r, \theta)\} \\ &= \int_0^\pi p(x \cdot \cos \theta + y \cdot \sin \theta, \theta) d\theta. \end{aligned}$$

Sebuah backprojectioon memberikan sebuah jawaban aproksimasi intuitif mengenai hasil konversi dari sinogram menjadi sebuah gambar 2D. Oleh sebab itu, digunakan transformasi iradon yang digambarkan oleh rumus :

$$f(x, y) = \mathcal{R}^{-1}\{p(r, \theta)\}.$$

*Filtered back projection* adalah algoritma rekonstruksi analitik yang didesain untuk mengatasi keterbatasan proyeksi belakang konvensional; algoritme ini menerapkan filter konvolusi untuk menghilangkan keburaman. Sampai saat ini, FBP merupakan metode utama dalam rekonstruksi pencitraan penampang melintang. Metode ini menggunakan persamaan simultan dari jumlah sinar yang diambil pada sudut gelombang yang berbeda untuk menghitung nilai koefisien atenuasi dalam penampang melintang. Hal ini dicapai melalui algoritma 250.000 persamaan matematis yang dapat diselesaikan oleh komputer berkapasitas tinggi. Profil atenuasi atau proyeksi yang dihasilkan yang diwakili oleh anatomi disimpan dalam memori komputer, dipecahkan dan direkonstruksi. Setiap piksel (elemen gambar) berhubungan dengan voxel (elemen volume) gambar. Persamaan simultan digunakan untuk memastikan nilai atenuasi yang benar dalam penjumlahan sinar. Untuk merepresentasikannya dengan cara yang mudah, bayangkan sebuah kotak dengan lubang di setiap sisinya, dan ada empat huruf di dalamnya yang masing-masing sesuai dengan A, B di bagian atas dan C, D di bagian bawah.

Kemudian di dalam kotak tersebut terdapat sebuah koin yang terletak di A. Koin tersebut akan berhubungan dengan "1" dan "0".

Karena dalam teknik FBP domain awal sinogram adalah frekuensi, maka proyeksi balik langsung sinogram akan membalikkan domain tersebut menjadi spasial. Transformasi domain akan menyebabkan sedikit distorsi matematis yang terakumulasi menjadi keburaman pada citra yang direkonstruksi, yang disebut keburaman  $1/r$ . Dalam teknik FBP permasalahan ini diselesaikan dengan terlebih dahulu menambahkan fungsi filter pada sinogram sebelum diproyeksikan kembali. Persamaan umum estimasi fungsi objek,  $f(x,y)$ , dalam FBP dapat dituliskan sebagai :

$$f(x,y) = \int_0^\pi p(s,\theta) * \frac{1}{-2\pi^2 s^2} d\theta \quad \text{mit } s = x \cos \theta + y \sin \theta$$

#### 2.4. Algebraic reconstruction technique (ART)

Dalam Computed Tomography (CT), teknik rekonstruksi tiga dimensi dari proyeksi telah digunakan selama bertahun-tahun dalam radiologi. Transformasi Fourier dua dimensi adalah algoritma yang paling umum digunakan dalam radiologi. Dalam teknik ini, sejumlah besar proyeksi pada sudut yang terdistribusi secara seragam di sekitar subjek diperlukan untuk rekonstruksi gambar. Dalam bidang akselerator fisika, orang berharap bahwa distribusi berkas partikel bermuatan yang relatif sederhana dapat direkonstruksi dari yang kecil jumlah proyeksi. Dalam praktik konvensional, hanya ada dua proyeksi, biasanya horisontal dan vertikal, diukur. Ini memberikan batasan yang parah pada tingkat detail yang bisa dilakukan dicapai. Teknik Rekonstruksi Aljabar (ART) yang diperkenalkan oleh Gordan, Bender dan Herman menggunakan tiga atau lebih proyeksi untuk merekonstruksi balok 2 dimensi distribusi kepadatan. Mereka telah menunjukkan bahwa peningkatan kualitas rekonstruksi terlihat jelas ketika proyeksi ketiga ditambahkan, tetapi proyeksi tambahan menambahkan lebih sedikit pada kualitas rekonstruksi.

Algoritme ART memiliki dasar intuitif yang sederhana. Setiap proyeksi densitas dilemparkan kembali melintasi ruang rekonstruksi di mana densitas dimodifikasi secara iteratif untuk membuat setiap proyeksi yang direkonstruksi sesuai dengan proyeksi yang diukur. Dengan asumsi bahwa pola yang sedang direkonstruksi tertutup dalam ruang persegi berukuran  $n \times n$  array piksel kecil,  $p_j$   $j = 1, \dots, n^2$  adalah angka keabuan atau densitas, yang seragam di dalam piksel tetapi berbeda dengan piksel lainnya. Sebuah "sinar" adalah sebuah wilayah persegi ruang yang terletak di antara dua garis paralel. Bobot jumlah sinar tertimbang adalah total keabuan dari gambar rekonstruksi di dalam sinar. Proyeksi pada sudut tertentu

adalah jumlah sinar yang tidak tumpang tindih dan sama lebar yang menutupi gambar. Algoritma ART terdiri dari pengubahan warna abu-abu pada setiap piksel yang berpotongan dengan sinar sedemikian rupa untuk membuat jumlah sinar sesuai dengan elemen yang sesuai dari proyeksi yang diukur. Asumsikan  $P$  adalah sebuah matriks berukuran  $m \times n^2$  dan vektor kolom  $m$  komponen  $R$ . Biarkan  $p_{i,j}$  menyatakan elemen ke- $(i, j)$  dari  $P$ , dan  $R_i$  menyatakan sinar ke- $i$  dari vektor proyeksi yang direkonstruksi  $R$ . Untuk  $1 \leq i \leq m$ ,  $N_i$  adalah jumlah piksel di bawah sinar proyeksi  $R_i$ , yang didefinisikan sebagai

$$N_i = \sum_{j=1}^{n^2} p_{i,j}^2.$$

## 2.5. Filters

Jika mengasumsikan sebuah data proyeksi setelah filtering adalah

$$\tilde{p}(x_r, \theta) = p(x_r, \theta) * h(x_r)$$

dimana  $p(x_r, \theta)$  adalah data proyeksi yang dikumpulkan,  $h(x_r)$  adalah fungsi filtering. Fungsi filter yang ideal akan didapatkan melalui inverse fourier transform, yang diekspresikan sebagai

$$h(x_r) = \int_{-\infty}^{\infty} |\rho| \exp(i2\pi\rho x_r) d\rho$$

dimana  $D$  adalah frekuensi spasial. Jika bicara sesuai teori, respons frekuensi dari sebuah filter seharusnya menghasilkan  $h(D) = |D|$ . Sebuah filter yang ideal adalah sebuah filter yang memiliki frekuensi yang tak terhingga dan sebuah divergensi pada interval integral yang terbatas.

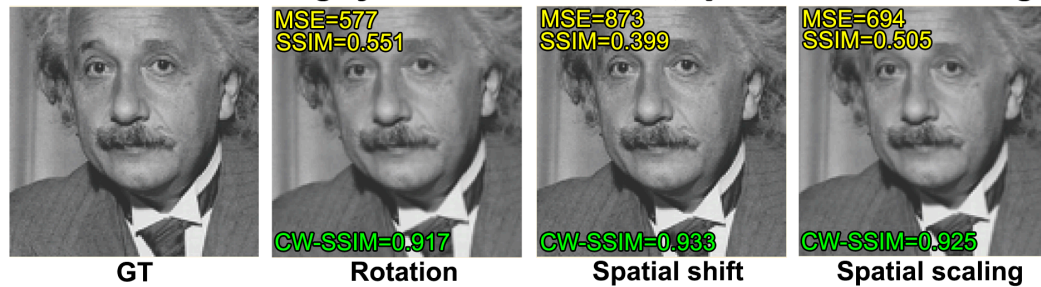
## 2.6. Image Evaluation

Peak signal to noise ratio (PSNR) dan structural index similarity (SSIM) merupakan dua alat ukur yang banyak digunakan dalam penilaian kualitas gambar. Khususnya pada gambar steganografi, kedua alat ukur ini digunakan untuk mengukur kualitas imperceptibility. PSNR digunakan lebih awal dibandingkan SSIM, mudah, telah banyak digunakan dalam berbagai pengukuran citra digital, dan telah dianggap teruji dan valid. SSIM adalah alat pengukuran yang lebih baru yang dirancang berdasarkan tiga faktor yaitu pencahayaan, kontras, dan struktur agar lebih sesuai dengan cara kerja sistem visual manusia. Beberapa penelitian telah membahas korelasi dan perbandingan kedua alat ukur ini, namun belum ada penelitian yang secara eksplisit membahas dan menyarankan alat ukur mana yang lebih cocok untuk steganografi. Penelitian ini bertujuan untuk mengkaji, membuktikan, dan menganalisis hasil pengukuran PSNR dan SSIM pada tiga metode steganografi citra berdomain spasial, yaitu LSB, PVD, dan CRT. Citra berwarna dipilih sebagai citra penampung karena penglihatan manusia lebih sensitif terhadap perubahan warna



dibandingkan dengan perubahan grayscale. Berdasarkan hasil pengujian didapatkan beberapa temuan yang bertolak belakang, dimana LSB memiliki nilai yang paling unggul berdasarkan PSNR dan PVD mendapatkan nilai yang paling unggul berdasarkan SSIM. Berdasarkan hasil pengujian dan analisis, penelitian ini menyimpulkan bahwa SSIM merupakan ukuran imperceptibility yang lebih baik dalam semua aspek dan sebaiknya pada penelitian steganografi selanjutnya setidaknya menggunakan SSIM.

#### PSNR and SSIM are highly sensitive to rotations, spatial shifts and scalings



**Gambar 5.** Penggunaan PSNR dan SSIM dalam mengevaluasi hasil gambar

### 2.5.1 PSNR

Banyak penelitian yang berkaitan dengan pemrosesan gambar dan sinyal menggunakan PSNR sebagai alat ukur kualitas. PSNR dihasilkan dari perhitungan logaritma mean square error (MSE) dari sebuah gambar. Di mana MSE secara tradisional menggunakan metode penjumlahan sebagai komponen utamanya. Pada gambar grayscale MSE dihitung berdasarkan dimensi  $M \times N$ , sedangkan pada gambar berwarna RGB MSE gambar RGB dapat dihitung berdasarkan dimensi  $M \times N \times O$ , yang dijabarkan pada rumus

$$MSE = \frac{1}{M \times N \times O} \sum_{x=1}^M \sum_{y=1}^N \sum_{z=1}^O \left[ \left( I_{(x,y,z)} - I'_{(x,y,z)} \right)^2 \right]$$

Dimana M dan N adalah resolusi citra, O adalah jumlah kanal citra,  $I(x,y,z)$  adalah nilai piksel citra asli pada koordinat x, y dan kanal z,  $I'$  adalah citra keluaran/hasil pemrosesan, dalam penelitian ini  $I'$  adalah citra stego. dapat dilihat bahwa rumus MSE dihasilkan dari penjumlahan kuadrat nilai piksel citra asli dikurangi piksel citra stego. MSE memiliki hubungan yang erat dengan PSNR karena nilai MSE digunakan untuk menghitung nilai PSNR adalah

$$PSNR = 10\log_{10} \left( \frac{\max^2}{MSE} \right)$$

Di mana max adalah nilai skala tertinggi dari skala abu-abu 8-bit. Dari rumus yang digunakan untuk menghitung PSNR, dapat dilihat bahwa PSNR terdiri dari nilai kuadrat error sebagai komponen utama. Nilai error dihasilkan oleh perbedaan nilai piksel pada koordinat dan saluran yang sama. PSNR akan menghasilkan nilai tak terhingga yang juga tidak merubah nilai piksel, sebaliknya jika semakin banyak perbedaan nilai piksel antara kedua gambar akan menghasilkan PSNR dengan nilai yang semakin kecil.

### 2.5.2 SSIM

SSIM adalah alat ukur lain yang digunakan untuk mengukur kualitas imperceptibility pada gambar steganografi. SSIM pertama kali dipublikasikan oleh, dalam penelitiannya disebutkan bahwa SSIM dibangun berdasarkan tiga faktor utama, yaitu luminance, contrast, dan structure. Ketiga faktor ini menggantikan metode penjumlahan yang digunakan sebagai dasar perhitungan PSNR. Pada gambar berwarna RGB, SSIM dapat didefinisikan dengan

$$SSIM(i, i') = l(i, i') c(i, i') s(i, i')$$

dimana,

$$s(i, i') = \frac{\sigma_{ii'} + C3}{\sigma_i \sigma_{i'} + C3} \quad l(i, i') = \frac{2\mu_i \mu_{i'} + C1}{\mu_i^2 + \mu_{i'}^2 + C1} \quad c(i, i') = \frac{2\sigma_i \sigma_{i'} + C2}{\sigma_i^2 + \sigma_{i'}^2 + C2}$$

### 2.7. State of the Art

Penulis	Tahun	Tujuan	Metode
Touraj Nikazad	2008	Menggunakan metode Diagonal Relaxed Orthogonal Projections (DROP) dan metode iterasi ART	Memperkenalkan metode baru dalam menyelesaikan persamaan dan pertidaksamaan linear

Hella Gay. Avenida & Renato V. Crisotomo	2022	Menggunakan algoritma ART dan Geometry Modelling	Mengimplementasi dan mengevaluasi algoritma ART paralel
M. Monoawarul Islam & Muftadi Ullah Arpon	2020	Menggunakan Radon Transform, Filter Back Projection, Kaiser Window Function	Mencoba mengembangkan Parallel Beam Projection dan mengimplementasikan Kaiser Window Function
Y. Haider, M. Arif, N. N. Rahman, dan M. Haseeb	2012	Menggunakan Radon Transform, Filter Back Projection, Rak-Lak filter, Shepp-Logan filter	Mengembangkan alat prototipe Computed Tomography Imaging berbasis infrared
Zakaria Z, Jaafar NH, Yazid NAM, Mansor M, Rahiman M, Rahim RA.	2010	Membahas penggunaan sinogram dalam algoritma CT Image Reconstruction menggunakan MATLAB	Menggunakan Filter Back Projection dan Normal Back Projection
Schofield R, King L, Tayal U, Castellano I, Stirrup J, Pontana F.	2020	Memberikan gambaran mengenai pada CT Image Reconstruction dengan metode FBP yang difilter dan potensi rekonstruksi berulang untuk mengurangi noise gambar dan dosis radiasi.	Menggunakan metode Filter Back Projection dan Iterative Reconstruction
Mohd. Soyapi Mohd Yusoff, Riza Sulaiman, dan Kamaruddin Shafinah	2012	Mengetahui kendala pada CT Image Reconstruction, seperti adanya efek keburaman, masalah kompleksitas komputerisasi	Menggunakan Simple Back Projection, Fourier Transform, Filter Back Projection, Summation Back Projection, ART, LST, Pemrograman Visual Basic dan C++
M.S.M. Yusoff & R. Sulaiman	2012	Melakukan validasi dari proses Image Reconstruction menggunakan Pemrograman Matlab	Menggunakan Fan-Beam Projection, Parallel-Beam Projection, Radon Transform, Pemrograman Matlab
Liubov Flores, Vicent Vidal, dan Gumersindo Verdu	2015	Membandingkan efisiensi dan kualitas gambar dari metode LSQR dengan SART	Menggunakan metode LSQR dan Filter Back Projection

Atina	2019	Membahas penggunaan Pemrograman Matlab dalam CT Image Reconstruction	Menggunakan Pemrograman Matlab
Chillarón M, Quintana-Ortí G, Vidal V, Verdú G	2020	Menyajikan metode baru dan membandingkannya dengan teknik rekonstruksi lain dalam CT Image Reconstruction.	Menggunakan metode faktorisasi QR dan algoritma ART
Wang C, Zhang H, Zeng Z, Yu J, Wang Y	2021	Meningkatkan akurasi dan stabilitas CT Image Reconstruction, terutama pada akurasi dan stabilitas parameter kalibrasi sistem CT.	Memanfaatkan Reverse Radon Transform algoritma Canny Edge Detection
Ferdinan Manuel Siahaan	2008	Menghasilkan Program Komputerisasi yang efektif dan efisien berbasis ART menggunakan Matlab pada CT Image Reconstruction	Menggunakan algoritma Algebraic Reconstruction Technique (ART), Fan-Beam Projection, dan Pemrograman Matlab

**Tabel 1.** Tabel Tinjauan Pustaka CT Image Reconstruction.

Pada penelitian ini, kami menempatkan posisi penelitian dengan mengacu pada terhadap salah satu jurnal berjudul “CT Image Reconstruction Using MATLAB” yang ditulis oleh Ferdinan Manuel Siahaan. Hal tersebut dikarenakan pada penelitian yang kami lakukan, juga akan berfokus dan menitikberatkan pada penggunaan bahasa Pemrograman MATLAB dalam melakukan rekonstruksi gambar ‘CT Image Reconstruction’ dengan membandingkan algoritma *Filter Back Projection* (FBP) dan *Algebraic Reconstruction Technique* (ART) serta dapat memberikan variasi kondisi – baik *blur* ataupun *noise* – terhadap keempat jenis Phantom yang telah dibentuk dan selanjutnya akan ditampilkan ke dalam Graphical User Interface (GUI) pada MATLAB App Designer. Tidak hanya itu, kami juga telah melakukan tinjauan pustaka terhadap beberapa jurnal dengan tujuan dalam memperkuat dan memberikan inovasi dari ide yang telah kami buat, seperti penambahan penggunaan empat jenis Filters dalam penerapan algoritma *Filter Back Projection* (FBP) berupa Ram-Lak, Hamming, Hanning, dan Cosine.

### BAB III. METODOLOGI

#### 3.1. Alat dan Bahan

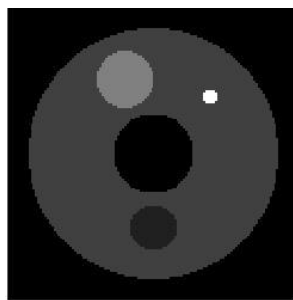
Untuk membuat sistem *CT scan image reconstruction*, digunakan laptop dengan prosesor AMD Ryzen 5 4500U dan Radeon Graphics 2.38 GHz 8 GB RAM. Seluruh algoritma dibuat menggunakan MATLAB R2022a. Kemudian, digunakan MATLAB App Designer untuk membuat *graphical user interface* (GUI) dari sistem.

#### 3.2. Metode

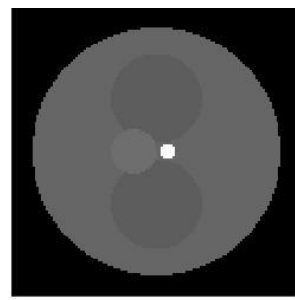
##### 3.2.1. Pembuatan *Phantom*

Dalam simulasi rekonstruksi gambar *CT scan* yang dilakukan pada sistem, digunakan empat jenis *phantom* berbentuk lingkaran yang dibangun dan di-generate langsung oleh program. Salah satu *phantom*, yaitu *phantom 4*, dibuat mengacu pada ACR *CT phantom*. Parameter dari masing-masing lingkaran, seperti posisi pusat ( $c_x$ ,  $c_y$ ), radius (rad), dan intensitas (amp), didefinisikan dalam suatu matriks.

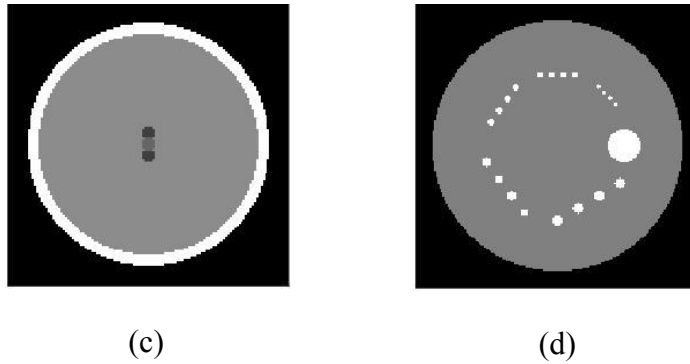
Proses pembuatan *phantom* dimulai dengan inisialisasi matriks citra *phantom* dengan ukuran  $n_x$  dan  $n_y$ . Selanjutnya, dilakukan iterasi untuk setiap lingkaran dalam matriks *circ*. Pada setiap iterasi, citra *phantom* diperbarui dengan mengatur intensitas piksel di dalam lingkaran sesuai dengan amplifikasi yang diberikan. Proses pengaturan intensitas piksel pada lingkaran menggunakan persamaan lingkaran pada geometri, dan indeks piksel yang berada di dalam lingkaran dihitung. Empat jenis *phantom* pada program dapat dilihat pada Gambar 5.



(a)



(b)



**Gambar 6.** (a) *Phantom 1*. (b) *Phantom 2*. (c) *Phantom 3*. (d) *Phantom 4*.

### 3.2.2. Penghasilan Sinogram dari *Phantom*

Phantom yang telah dibentuk di tahap awal kemudian diubah menjadi sinogram dengan menggunakan fungsi radon di MATLAB. Fungsi radon pada MATLAB digunakan untuk melakukan transformasi radon terhadap citra objek, menghasilkan sinogram yang merepresentasikan proyeksi dari objek tersebut pada berbagai sudut rotasi. Transformasi radon adalah proses matematis yang mencatat jumlah intensitas sinar-X yang melewati objek pada sudut-sudut tertentu. Parameter utama dari fungsi radon adalah citra objek yang akan diproyeksikan dan vektor sudut rotasi yang menentukan proyeksi pada sudut-sudut tertentu. Proses ini mencerminkan cara citra objek akan terlihat jika dilihat dari berbagai sudut rotasi. Sudut rotasi ( $\theta$ ) yang digunakan dalam fungsi radon adalah 0:1:179 yang berarti sudut rotasi dari 0 hingga 179 derajat. Hasil dari fungsi ini adalah matriks R, yang merupakan sinogram. Setiap kolom pada matriks ini mewakili proyeksi pada sudut tertentu, dan setiap baris menunjukkan posisi detektor pada proyeksi tersebut. Kemudian, hasil sinogram diperlihatkan dalam bentuk gambar.

### 3.2.3. Rekonstruksi Gambar

Hasil sinogram dalam tahap sebelumnya digunakan untuk rekonstruksi citra objek menggunakan algoritma rekonstruksi. Pada sistem yang dibuat dalam laporan ini, digunakan tiga jenis metode rekonstruksi, yaitu *Simple Back Projection*, *Filtered Back Projection* (FBP), dan *Algebraic Reconstruction Technique* (ART).

- *Simple Back Projection* dan *Filtered Back Projection* (FBP)

Metode Simple Back Projection (SBP) adalah pendekatan yang sederhana dan langsung dalam merekonstruksi gambar dari sinogram. Pada SBP, setiap proyeksi dari sinogram diberikan sebagai intensitas langsung

pada posisi yang sesuai dalam gambar rekonstruksi. Hal ini dilakukan dengan memproyeksikan setiap proyeksi ke dalam gambar tanpa adanya proses *filtering* atau modifikasi lainnya.

Untuk metode *Filtered Back Projection* (FBP), digunakan fungsi iradon untuk merekonstruksi gambar dari sinogram. prinsip dasar FBP adalah melakukan proses invers terhadap proyeksi untuk menghasilkan rekonstruksi citra objek. FBP bekerja dengan cara memproyeksikan sinogram ke dalam ruang Fourier, menerapkan filter untuk menekankan atau meredam beberapa frekuensi, dan kemudian melakukan inverse Fourier transform. Jenis filter yang digunakan pada sistem yang dibangun mencakup Ram-Lak, Hamming, Hanning, dan Cosine. Setiap jenis filter memiliki karakteristiknya masing-masing, sehingga dapat diimplementasikan sesuai kualitas gambar *phantom* atau sinogram.

- *Algebraic Reconstruction Technique* (ART)

Pada metode ini, gambar rekonstruksi diinisialisasi dengan nilai awal yang dapat berasal dari pendekatan awal atau secara acak. Proses iteratif dimulai, di mana residual (selisih antara proyeksi teramati dan proyeksi yang diestimasi dari gambar rekonstruksi saat ini) dihitung. Gambar rekonstruksi diperbarui menggunakan matriks sinogram dan residual, dengan tujuan mendekati proyeksi yang diamati. Iterasi dilakukan sebanyak jumlah yang ditentukan oleh *user*. Setiap iterasi melibatkan pembaruan gambar rekonstruksi berdasarkan pendekatan ART.

#### 3.2.4. Penghitungan PSNR dan SSIM

PSNR dihitung dengan mendefinisikan variabel MAX dan MSE dan dimasukkan ke dalam kalkulasi PSNR sesuai dengan rumus. Begitu juga dengan SSIM yang membandingkan hasil rekonstruksi gambar dengan gambar aslinya dari aspek kecerahan, kontras, dan struktur. Pada GUI, nilai PSNR dan SSIM akan diperbarui ketika terdapat perubahan berupa rekonstruksi gambar dengan *phantom*, nilai blur, maupun nilai *noise* yang berbeda.

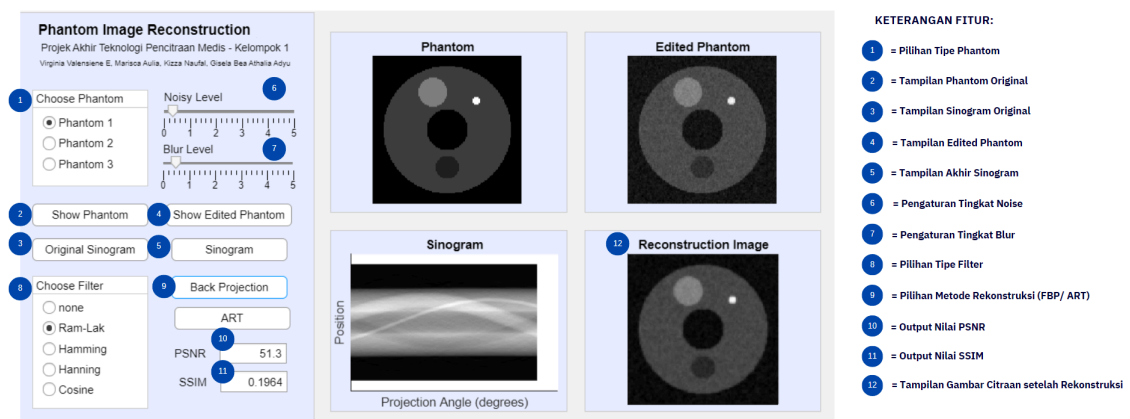
#### 3.2.5. Simulasi

Dilakukan simulasi untuk menerapkan setiap metode rekonstruksi pada setiap jenis *phantom* yang tersedia. Setiap *phantom* di-generate ke dalam empat kondisi: tanpa blur dan *noise* (original), dengan blur tingkat 1, dengan *noise* tingkat 1, dan dengan blur + *noise* masing-masing tingkat 1.

Untuk metode FBP, digunakan jenis filter *none* untuk merepresentasikan metode SBP, Ram-Lak, Hamming, Hanning, dan Cosine. Sementara itu, untuk metode ART, digunakan tiga jumlah iterasi yang berbeda: 50, 100, dan 500. Dari keseluruhan simulasi, akan dilihat hasil PSNR dan SSIM untuk melihat kemampuan masing-masing metode untuk kondisi gambar *phantom* yang berbeda.

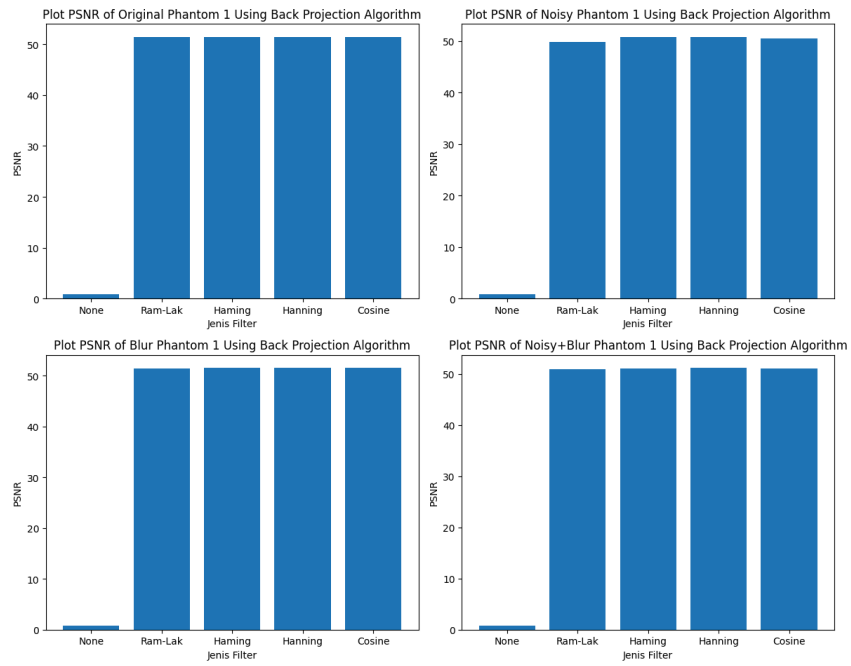
### 3.3. Hasil dan Pembahasan

Sistem rekonstruksi gambar yang dibangun pada laporan ini dikemas dengan GUI agar lebih *user-friendly* (Gambar 6). *Phantom* 1, 2, 3, dan 4 dibentuk dengan intensitas dan kontras yang berbeda. *Phantom* 1 diatur agar memiliki kontras antar objek yang paling tinggi, sementara *phantom* 2 memiliki kontras antar objek yang paling rendah. Pada *phantom* 1, terlihat bahwa hasil PSNR dari empat jenis filter pada metode FBP cukup seragam atau tidak memiliki perbedaan secara signifikan (Gambar 7(a)). Namun, PSNR dari SBP sangat rendah, bahkan di seluruh simulasi. Hal ini menjadikan metode ini menjadi metode yang paling tidak disarankan untuk digunakan. Dari skor SSIM, terlihat bahwa kondisi *noisy* pada gambar input akan menghasilkan skor SSIM paling rendah dibanding kondisi lain (Gambar 7(b)). Walaupun perbedaan hasil skor tidak terlalu signifikan, filter Hamming dan Hanning menghasilkan skor SSIM yang sedikit lebih unggul dibanding filter lainnya di kondisi *original* dan *noisy*. Sementara itu, hasil PSNR dari metode ART secara umum berbanding lurus terhadap jumlah iterasi (Gambar 7(c)). Meski, skor PSNR pada metode ART dengan 500 iterasi sedikit lebih baik dibanding metode FBP, tetapi skor SSIM nya masih jauh lebih rendah dibanding metode FBP (Gambar 7(d)).

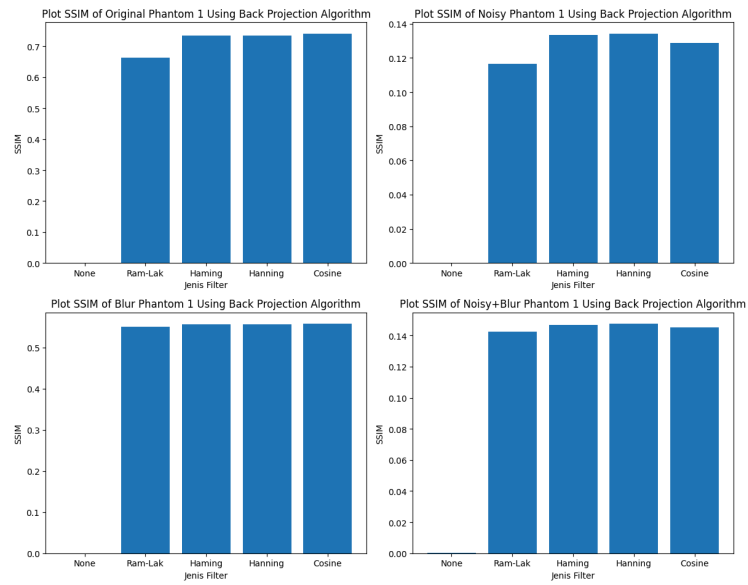


**Gambar 7.** GUI sistem rekonstruksi gambar CT *scan* dan keterangan fitur nya.

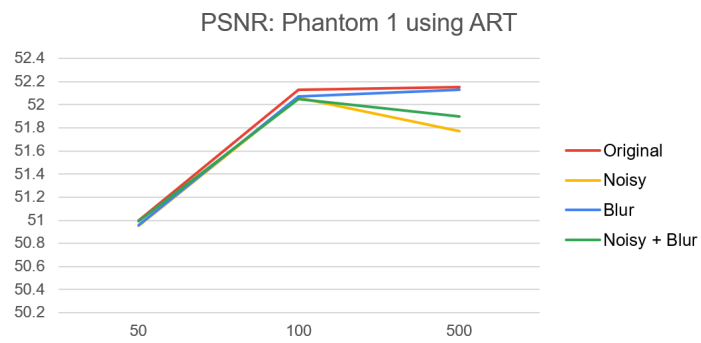




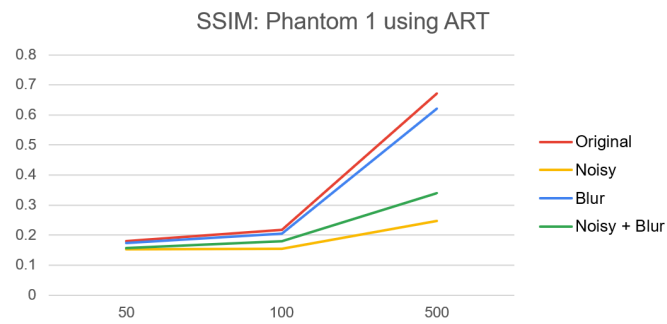
(a)



(b)



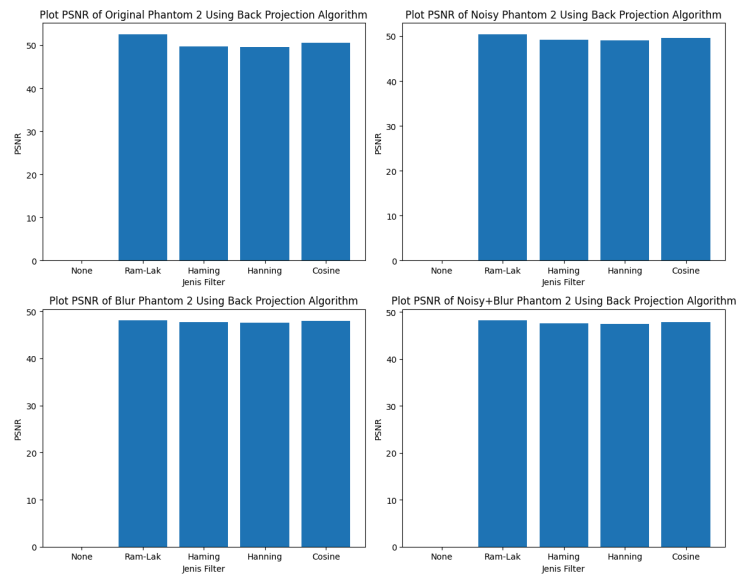
(c)



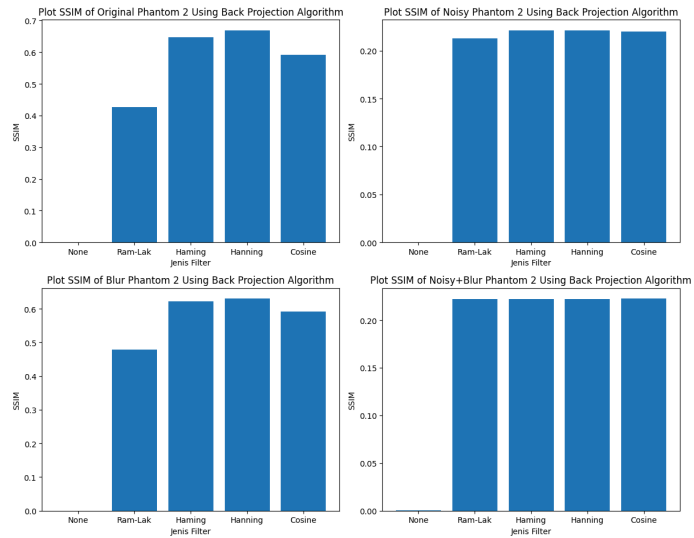
(d)

**Gambar 8.** Skor (a) PSNR dan (b) SSIM dari metode FBP, (c) PSNR dan (d) SSIM dari metode ART pada *Phantom 1*.

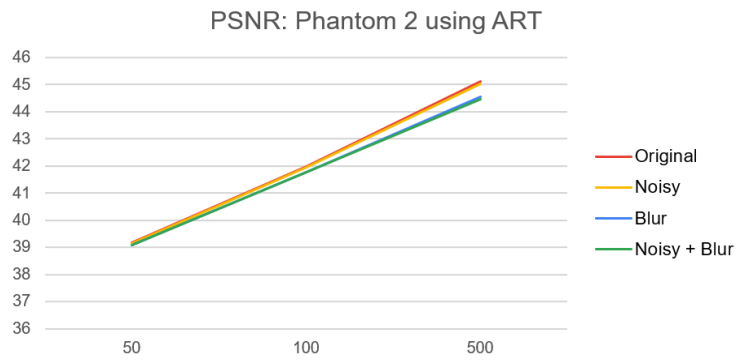
Pada *phantom 2*, skor PSNR dari setiap jenis filter juga tidak berbeda secara signifikan (Gambar 8(a)). Namun, hasil skor SSIM dari filter Ram-Lak cenderung lebih rendah dibandingkan dengan filter lainnya, terutama pada kondisi *original* dan *blur*(Gambar 8(b)). Metode ART sendiri memiliki performa yang sedikit lebih rendah dibandingkan dengan metode FBP (Gambar 8(c) dan (d)).



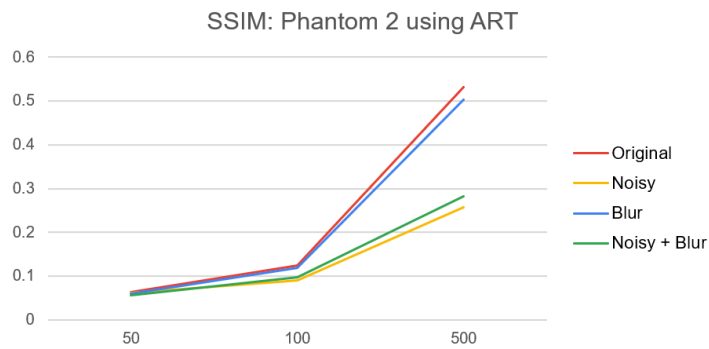
(a)



(b)



(c)

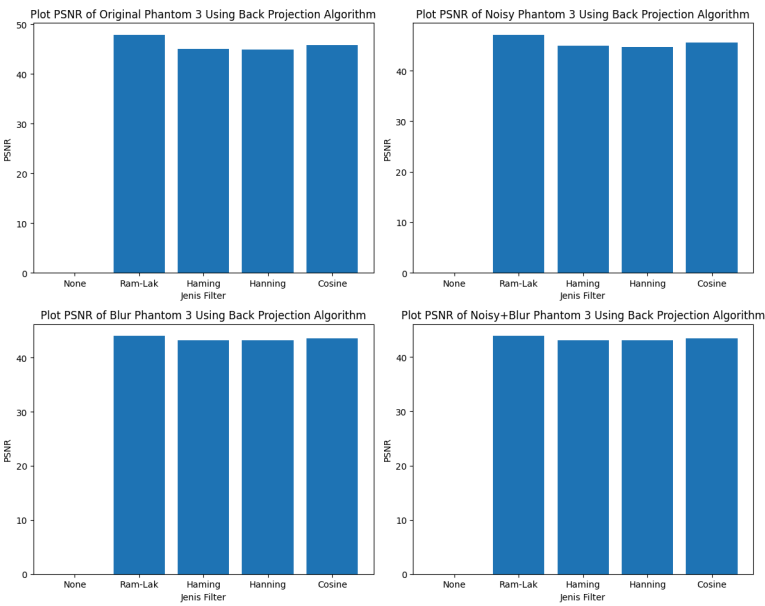


(d)

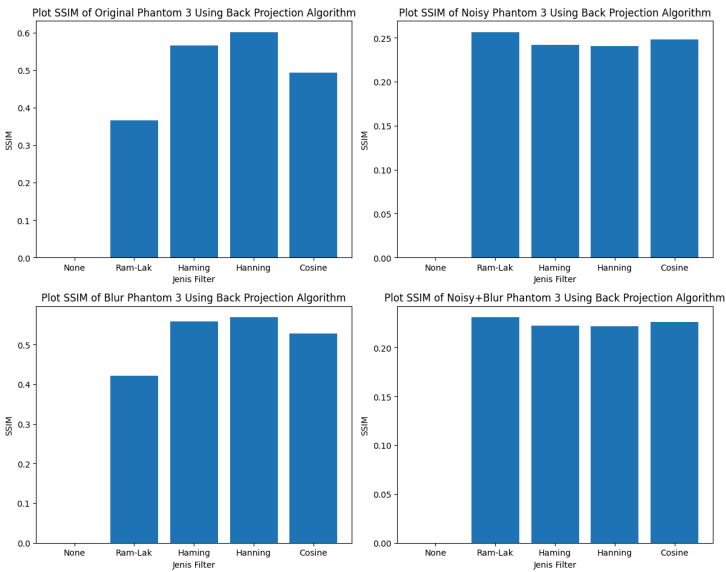
**Gambar 9.** Skor (a) PSNR dan (b) SSIM dari metode FBP, (c) PSNR dan (d) SSIM dari metode ART pada *Phantom 2*.

Pada phantom 3, metode FBP dengan jenis filter Ram-Lak masih menghasilkan skor PSNR yang paling baik (Gambar 9(a)), tetapi lagi-lagi jenis Hamming dan Hanning lebih unggul untuk menghasilkan skor SSIM yang lebih tinggi untuk metode FBP, terutama pada kondisi *original* dan *blurry* (Gambar 9(b)).

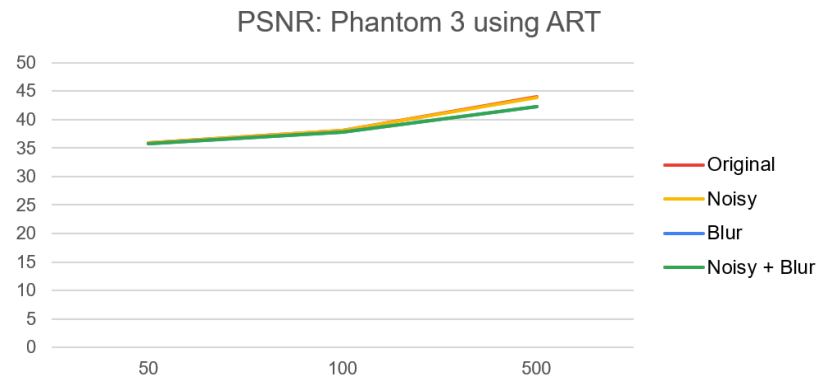
Namun, secara keseluruhan, SSIM yang ART hasilkan lebih baik, terutama pada kondisi *noisy* dan *noisy+blurry* dibandingkan dengan FBP (Gambar 9(d)).



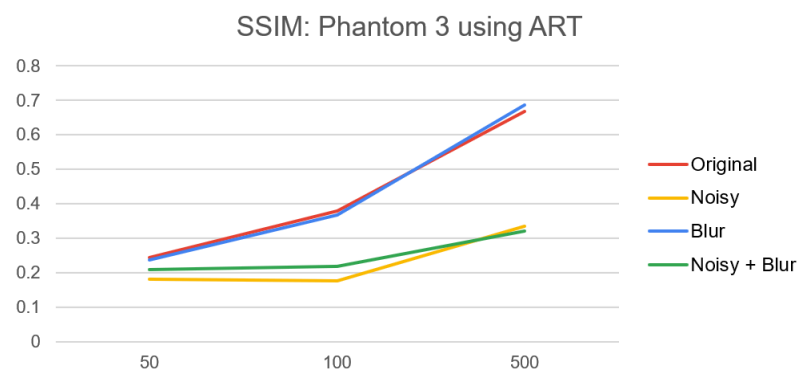
(a)



(b)



(c)



(d)

**Gambar 10.** Skor (a) PSNR dan (b) SSIM dari metode FBP, (c) PSNR dan (d) SSIM dari metode ART pada *Phantom 3*.

## BAB IV. PENUTUP

### 4.1. Kesimpulan

Dari hasil program yang sudah dibangun, dibentuk suatu GUI yang dapat merkonstruksi empat jenis *phantom* dengan dua jenis metode rekonstruksi, yaitu *Filtered Back Projection* (FBP) dan *Algebraic Reconstruction Technique* (ART), dengan penerapan berbagai jenis filter seperti Ram-Lak, Hamming, Hanning, dan Cosine. *Phantom* yang dibentuk memiliki kontras yang berbeda-beda, dengan *phantom* 1 memiliki kontras antar objek paling tinggi dan *phantom* 2 memiliki kontras antar objek paling rendah. *Phantom* 4 dibentuk sesuai dengan ACR CT *phantom*.

Berdasarkan evaluasi pada empat jenis *phantom* dengan penerapan metode rekonstruksi FBP dan ART menggunakan berbagai jenis filter, hasil menunjukkan karakteristik yang berbeda tergantung pada kondisi *phantom*. Pada kondisi kontras rendah, filter Ram-Lak mungkin memberikan skor PSNR yang tinggi karena mampu menangkap detail frekuensi tinggi. Namun, hal ini dapat disertai dengan penurunan skor SSIM karena hasil yang terlalu tajam. Di sisi lain, filter Hamming dan Hanning dapat memberikan keseimbangan yang lebih baik antara PSNR dan SSIM, karena efek penghalusan ringan membantu memperbaiki kontras rendah.

Sementara pada kondisi kontras tinggi, filter Ram-Lak dapat memberikan hasil terbaik karena mampu menangkap detail tinggi. Sedangkan filter Hamming dan Hanning, meskipun memiliki PSNR tinggi, dapat mengalami penurunan SSIM karena efek penghalusan. Filter Cosine, sebagai perantara antara Ram-Lak dan Hamming/Hanning, dapat memberikan hasil di tengah-tengah keduanya.

Pada kondisi *noisy*, filter Ram-Lak mungkin memberikan PSNR tinggi tetapi SSIM rendah karena *noise*. Filter Hamming dan Hanning dapat memberikan hasil yang lebih baik, mengurangi efek noise dan memberikan keseimbangan antara PSNR dan SSIM. Filter Cosine, sekali lagi, dapat berada di antara keduanya.

Terakhir, pada kondisi *blur*, filter Ram-Lak mungkin menghasilkan PSNR tinggi tetapi SSIM rendah karena efek tajam, sementara filter Hamming dan Hanning dapat memberikan hasil baik dengan efek penghalusan yang membantu meredakan *blur*. Filter Cosine dapat memberikan hasil yang seimbang antara PSNR dan SSIM. Dengan demikian, pemilihan filter dan metode rekonstruksi perlu disesuaikan dengan karakteristik *phantom* dan kondisi pengambilan gambar untuk mencapai hasil rekonstruksi yang optimal.

### 4.2. Saran dan Studi Lanjutan

Hasil penelitian ini memberikan beberapa saran untuk studi lanjutan yang dapat mendalami dan memperluas pemahaman dalam bidang rekonstruksi gambar CT. Pertama, perlu dilakukan penelitian lebih lanjut terkait optimasi parameter filter pada metode rekonstruksi FBP dan ART. Pengaturan parameter

tersebut memiliki dampak signifikan terhadap kualitas rekonstruksi, sehingga penelitian lebih lanjut akan memberikan wawasan yang lebih mendalam.

Selanjutnya, direkomendasikan untuk melibatkan metode rekonstruksi lainnya, seperti metode iteratif lainnya atau teknik deep learning, dalam perbandingan. Studi ini dapat memberikan pemahaman lebih mendalam tentang performa relatif berbagai metode rekonstruksi dalam konteks yang lebih luas. Analisis efek pada jenis phantom yang berbeda juga dapat menjadi fokus studi lanjutan. Melibatkan phantom dengan variasi sifat jaringan dan kontras dapat membantu memahami sejauh mana hasil rekonstruksi dipengaruhi oleh karakteristik anatomi objek yang diuji.

Terakhir, studi lebih mendalam terhadap pengaruh jumlah iterasi pada metode ART perlu dilakukan. Hal ini dapat membantu menentukan titik optimal antara akurasi rekonstruksi dan efisiensi komputasi. Penerapan hasil penelitian ini pada data klinis nyata juga menjadi langkah penting untuk memvalidasi temuan-temuan ini di lingkungan klinis sebenarnya.

## Referensi

Atina (2019) 'Aplikasi Matlab Pada Teknologi pencitraan medis', Jurnal Penelitian Fisika dan Terapannya (JUPITER), 1(1), p. 28. doi:10.31851/jupiter.v1i1.3123.

Avenido, H.G. and Crisostomo, R.V. (2022) 'Image reconstruction from a large number of projections in proton and  $^{12}\text{C}$  ions computed tomography using sequential and Parallel Art Algorithms', Procedia Computer Science, 197, pp. 126–134. doi:10.1016/j.procs.2021.12.126.

Chillarón, M. et al. (2020) 'Computed tomography medical image reconstruction on affordable equipment by using out-of-core techniques', Computer Methods and Programs in Biomedicine, 193, p. 105488. doi:10.1016/j.cmpb.2020.105488.

CT Scan (2022) Mayo Clinic. Available at: <https://www.mayoclinic.org/tests-procedures/ct-scan/about/pac-20393675> (Accessed: 12 December 2023).

Flores, L., Vidal, V. and Verdú, G. (2015) 'Iterative Reconstruction from few-view projections', Procedia Computer Science, 51, pp. 703–712. doi:10.1016/j.procs.2015.05.188.

Haider, Y. et al. (2009) 'A prototype system for infrared computed tomography for Image Reconstruction', 2009 IEEE 13th International Multitopic Conference [Preprint]. doi:10.1109/inmic.2009.5383105.

Hashmi, U.S. et al. (2020) 'Application of reconstruction and optimization algorithms in optical tomography', World Journal of Engineering and Technology, 08(02), pp. 188–203. doi:10.4236/wjet.2020.82016.

Islam, M.M. and Arpon, M.U. (2020) Image Reconstruction Techniques using Kaiser Window in 2D CT Imaging. thesis. Department Physics and Electrical Engineering.



Maier A, Steidl S, Christlein V, Joachim Hornegger, Springerlink (Online Service. Medical Imaging Systems : An Introductory Guide. Cham: Springer International Publishing; 2018.

Nikazad, T. (2008) Algebraic Reconstruction Methods. thesis. Linköpings universitet, Department of Mathematics Scientific Computing.

Romans LE. Computed tomography for technologists : a comprehensive text. Philadelphia: Wolters Kluwer; 2011.

Schofield, R. et al. (2020) 'Image reconstruction: Part 1 – understanding filtered back projection, noise and image acquisition', Journal of Cardiovascular Computed Tomography, 14(3), pp. 219–225. doi:10.1016/j.jcct.2019.04.008.

Siahaan, F.M. (2008) CT Image Reconstruction Using Matlab Programming. thesis. Faculty of Science and Technology.

Soyapi, Mohd & Yusoff, Mohd & Sulaiman, Riza & Shafinah, Kamarudin. (2012). Image Reconstruction for CT Scanner by Using Filtered Back projection Approach. 88. 797-803.

Szczykutowicz, T.P. et al. (2022) 'A review of deep learning CT reconstruction: Concepts, limitations, and promise in clinical practice', Current Radiology Reports, 10(9), pp. 101–115. doi:10.1007/s40134-022-00399-5.

Wang, C. et al. (2021) 'Application of image reconstruction based on inverse radon transform in CT system parameter calibration and Imaging', Complexity, 2021, pp. 1–10. doi:10.1155/2021/5360716.

Yusoff, M.S.M. & Sulaiman, Riza & Shafinah, Kamarudin & R., Fatihah. (2012). IMAGE RECONSTRUCTION OF COMPUTED TOMOGRAPHY USING FAN-BEAM. International Journal of Engineering, Computer Science and Technology. 0105. 06-11.

Zakaria, Z. et al. (2010) 'Sinogram concept approach in image reconstruction algorithm of a computed tomography system using MATLAB', 2010 International Conference on Computer Applications and Industrial Electronics [Preprint].  
doi:10.1109/iccaie.2010.5735132.

## Lampiran

```
classdef Kelompok_1_Image_Reconstruction < matlab.apps.AppBase
```

```
% Properties that correspond to app components
```

```
properties (Access = public)
```

```
    UIFigure          matlab.ui.Figure
```

```
    reconstruction_panel  matlab.ui.container.Panel
```

```
    UIAxes_4           matlab.ui.control.UIAxes
```

```
    edited_phantom_panel  matlab.ui.container.Panel
```

```
    UIAxes_2           matlab.ui.control.UIAxes
```

```
    control_panel       matlab.ui.container.Panel
```

```
    ARTButton           matlab.ui.control.Button
```

```
    VirginiaValensieneEMariscaAuliaKizzaNaufalGiselaBeaAthaliaLabel  
    matlab.ui.control.Label
```

```
    SSIMEditField       matlab.ui.control.NumericEditField
```

```
    SSIMEditFieldLabel  matlab.ui.control.Label
```

```
    PSNREditField       matlab.ui.control.NumericEditField
```

```
    PSNREditFieldLabel  matlab.ui.control.Label
```

```
    ProyekAkhirTeknologiPencitraanMedisKelompok1Label  matlab.ui.control.Label
```

```
    BackProjectionButton  matlab.ui.control.Button
```

```
    ChooseFilterButtonGroup  matlab.ui.container.ButtonGroup
```

```
    CosineButton          matlab.ui.control.RadioButton
```

```
    HanningButton         matlab.ui.control.RadioButton
```

```
    dot2Button            matlab.ui.control.RadioButton
```

```
    HammingButton         matlab.ui.control.RadioButton
```

```
    RamLakButton          matlab.ui.control.RadioButton
```

```

noneButton          matlab.ui.control.RadioButton
OriginalSinogramButton  matlab.ui.control.Button
SinogramButton      matlab.ui.control.Button
BlurLevelSlider     matlab.ui.control.Slider
BlurLevelSliderLabel  matlab.ui.control.Label
ShowEditedPhantomButton  matlab.ui.control.Button
NoisyLevelSlider     matlab.ui.control.Slider
NoisyLevelSliderLabel  matlab.ui.control.Label
ShowPhantomButton    matlab.ui.control.Button
ChoosePhantomButtonGroup  matlab.ui.container.ButtonGroup
Phantom4Button       matlab.ui.control.RadioButton
dotButton            matlab.ui.control.RadioButton
Phantom3Button       matlab.ui.control.RadioButton
Phantom2Button       matlab.ui.control.RadioButton
Phantom1Button       matlab.ui.control.RadioButton
PhantomImageReconstructionLabel  matlab.ui.control.Label
phantom_panel        matlab.ui.container.Panel
UIAxes               matlab.ui.control.UIAxes
sinogram_panel       matlab.ui.container.Panel
UIAxes_3             matlab.ui.control.UIAxes
end

```

```

properties (Access = private)

```

```

    circ = [ 0  0 110 2;
            -65 0 20 1;
            0  0 35 0;

```

```

        65 -25 25 4;
        50 50 7 8];
originalImage;
editedImage;
sinogram;
reconstructedImage;
reconstructedImageART;
end

methods (Access = private)

    function x = ART(~,A,AT,b,x,lambda,niter,bpos)

        if (nargin < 7)
            bpos = true;
        end

        if (nargin < 6)
            niter = 1e2;
        end

        ATA = AT(A(ones(size(x), 'single')));

        for i = 1:niter

            x = x + lambda*AT(b - A(x))./ATA;

            if (bpos)

```

```
x(x < 0) = 0;
```

```
end
```

```
figure(1); colormap gray;
```

```
imagesc(x);
```

```
axis image off;
```

```
title(num2str([i, niter], '%d / %d'));
```

```
drawnow();
```

```
end
```

```
x = gather(x);
```

```
end
```

```
% ART reconstruction function
```

```
function ARTReconstruction(app, numIterations)
```

```
% Extract relevant parameters
```

```
nx = 170;
```

```
theta = 0:1:179; % Projection angles
```

```
sgr = app.sinogram;
```

```
% Initialize the reconstructed image
```

```
app.reconstructedImageART = zeros(nx, nx);
```

```
% ART algorithm using the provided function
```

```
lambda = 0.1; % Relaxation parameter
```

```

% Create a progress bar

h = waitbar(0, 'ART Reconstruction in Progress...', 'Name', 'Please Wait');

for iter = 1:numIterations

    % Update progress bar

    waitbar(iter / numIterations, h, sprintf('Iteration %d / %d', iter, numIterations));

    % Perform ART reconstruction

    app.reconstructedImageART = app.ART(@(x) radon(x, theta), @(x) iradon(x,
theta, 'linear', 'none', 1, nx), sgr, app.reconstructedImageART, lambda, 1, true);

end

% Close the progress bar

close(h);

app.reconstructedImageART = imrotate(app.reconstructedImageART,-90);

% Display the final reconstructed image with ART in UIAxes_4

imagesc(app.UIAxes_4, app.reconstructedImageART);

colormap(app.UIAxes_4, 'gray');

axis(app.UIAxes_4, 'square');

app.UIAxes_4.DataAspectRatio = [1 1 1];

title(app.UIAxes_4, 'Reconstructed Image using ART');

end

end

```

```
% Callbacks that handle component events
```

```
methods (Access = private)
```

```
% Code that executes after component creation
```

```
function startupFcn(app)
```

```
    app.UIAxes.XTick = [];
```

```
    app.UIAxes.YTick = [];
```

```
    app.UIAxes_2.XTick = [];
```

```
    app.UIAxes_2.YTick = [];
```

```
    app.UIAxes_3.XTick = [];
```

```
    app.UIAxes_3.YTick = [];
```

```
    app.UIAxes_4.XTick = [];
```

```
    app.UIAxes_4.YTick = [];
```

```
    app.dotButton.Value = true;
```

```
    app.Phantom1Button.Value = false;
```

```
    app.Phantom2Button.Value = false;
```

```
    app.Phantom3Button.Value = false;
```

```
    app.dot2Button.Value = true;
```

```
    app.noneButton.Value = false;
```

```
    app.RamLakButton.Value = false;
```

```
    app.HammingButton.Value = false;
```

```
end
```

```
% Selection changed function: ChoosePhantomButtonGroup
```

```
function ChoosePhantomButtonGroupSelectionChanged(app, event)
```

```
    selectedButton = app.ChoosePhantomButtonGroup.SelectedObject;
```



```

if selectedButton== app.dotButton

elseif selectedButton == app.Phantom1Button

    app.circ = [ 0  0 110  2;

                -65  0 20  1;

                0  0 35  0;

                65 25 25  4;

                50 -50 7  8];

elseif selectedButton == app.Phantom2Button

    app.circ = [ 0  0 110 12;

                -47  0 40 11;

                47  0 40 11;

                0 20 20 13;

                0 -10 7 30];

elseif selectedButton == app.Phantom3Button

    app.circ = [ 0  0 110 20;

                0  0 100 11;

                10  0  7  5;

                -10  0  7  5;

                0  0  7  8];

elseif selectedButton == app.Phantom4Button

    app.circ = [ 0  0 150  1;

                % 25mm

                0 -80 20  2;

                % 2mm

                50 -70 2  2;

                57 -63 2  2;

                64 -56 2  2;

```

```

71 -49 2 2;

% 3mm

85 -21 3 2;

85 -7 3 2;

85 7 3 2;

85 21 3 2;

% 4mm

70 50 4 2;

56 60 4 2;

42 70 4 2;

28 80 4 2;

% 5mm

-20 85 5 2;

-40 70 5 2;

-60 55 5 2;

-80 40 5 2;

% 4mm

-90 0 6 2;

-75 -25 6 2;

-60 -50 6 2;

-45 -75 6 2;];

end

end

% Button pushed function: ShowPhantomButton

function ShowPhantomButtonPushed(app, event)

```

```

% Image parameters: number of pixels, size, etc.

nx = 170; ny = 170;

dx = 2;    % 2 mm / pixel

x = dx * ((1:nx)'-(nx+1)/2);

y = -dx * ((1:ny)'-(ny+1)/2);

xx = x(:,ones(1,nx));

yy = y(:,ones(1,ny))';

%

%   Generate data for disk phantom

%

phantom = zeros(nx,ny);

for ii=1:size(app.circ,1)

    cx = app.circ(ii,1); cy = app.circ(ii,2); rad = app.circ(ii,3); amp = app.circ(ii,4);

    t = find( ((xx-cx)/rad).^2 + ((yy-cy)/rad).^2 <= 1 );

    phantom(t) = amp * ones(size(t));

end

%

% Display the phantom in UIAxes

imagesc(x, y, phantom, 'Parent', app.UIAxes); % Use imagesc instead of imshow

colormap(app.UIAxes, 'gray');

axis(app.UIAxes, 'square');

app.UIAxes.DataAspectRatio = [1 1 1];

app.originalImage = phantom;

end

```

```

% Button pushed function: ShowEditedPhantomButton

function ShowEditedPhantomButtonPushed(app, event)

% Image parameters: number of pixels, size, etc.

nx = 170; ny = 170;

dx = 2;    % 2 mm / pixel

x = dx * ((1:nx)'-(nx+1)/2);

y = -dx * ((1:ny)'-(ny+1)/2);

xx = x(:,ones(1,nx));

yy = y(:,ones(1,ny))';

% Apply noise and blur to the phantom and display the edited phantom

noisyLevel = app.NoisyLevelSlider.Value;

blurLevel = app.BlurLevelSlider.Value;

% Generate data for disk phantom

phantom = zeros(nx,ny);

for ii = 1:size(app.circ, 1)

    cx = app.circ(ii,1); cy = app.circ(ii,2); rad = app.circ(ii,3); amp =
app.circ(ii,4);

    t = find( ((xx-cx)/rad).^2 + ((yy-cy)/rad).^2 <= 1 );

    phantom(t) = amp * ones(size(t));

end

% Add noise

noisyPhantom = phantom + noisyLevel * randn(size(phantom));

```

```

% Apply blur

blurredPhantom = imgaussfilt(noisyPhantom, blurLevel);

% Display the edited phantom in UIAxes_2
imagesc(x, y, blurredPhantom, 'Parent', app.UIAxes_2);
colormap(app.UIAxes_2, 'gray');
axis(app.UIAxes_2, 'square');
app.UIAxes_2.DataAspectRatio = [1 1 1];

app.editedImage = blurredPhantom;
end

```

% Button pushed function: OriginalSinogramButton

```
function OriginalSinogramButtonPushed(app, event)
```

```

    nx = 170;

    dx = 2; % 2 mm / pixel
    x = dx * ((1:nx)'-(nx+1)/2);

    theta = 0:1:179; % Projection angles
    app.sinogram = radon(app.originalImage, theta);

    % Display the sinogram in UIAxes_3
    imagesc(app.UIAxes_3, theta, x, app.sinogram);
    colormap(app.UIAxes_3, 'gray');
    xlabel(app.UIAxes_3, 'Projection Angle (degrees)');
    ylabel(app.UIAxes_3, 'Position');
end

```

% Button pushed function: SinogramButton

function SinogramButtonPushed(app, event)

nx = 170;

dx = 2; % 2 mm / pixel

x = dx \* ((1:nx)'-(nx+1)/2);

theta = 0:1:179; % Projection angles

app.sinogram = radon(app.editedImage, theta);

% Display the sinogram in UIAxes\_3

imagesc(app.UIAxes\_3, theta, x, app.sinogram);

colormap(app.UIAxes\_3, 'gray');

xlabel(app.UIAxes\_3, 'Projection Angle (degrees)');

ylabel(app.UIAxes\_3, 'Position');

end

% Selection changed function: ChooseFilterButtonGroup

function ChooseFilterButtonGroupSelectionChanged(app, event)

nx = 170;

theta = 0:1:179; % Projection angles

selectedButton = app.ChooseFilterButtonGroup.SelectedObject;

if selectedButton == app.noneButton

app.reconstructedImage = iradon(app.sinogram, theta, 'linear', 'none', 1, nx);

elseif selectedButton == app.RamLakButton

```

        app.reconstructedImage = iradon(app.sinogram, theta, 'linear', 'Ram-Lak', 1,
nx);

        elseif selectedButton == app.HammingButton

            app.reconstructedImage = iradon(app.sinogram, theta, 'linear', 'Hamming', 1,
nx);

        elseif selectedButton == app.HanningButton

            app.reconstructedImage = iradon(app.sinogram, theta, 'linear', 'Hann', 1, nx);

        elseif selectedButton == app.CosineButton

            app.reconstructedImage = iradon(app.sinogram, theta, 'linear', 'Cosine', 1,
nx);

        end

    end
end

```

```

% Button pushed function: BackProjectionButton

```

```

function BackProjectionButtonPushed(app, event)

ori = double(app.originalImage);

recons = double(flipud(app.reconstructedImage));

```

```

% Display the reconstructed image in UIAxes_4

```

```

imagesc(app.UIAxes_4, recons);

colormap(app.UIAxes_4, 'gray');

title(app.UIAxes_4, 'Reconstruction Image');

axis(app.UIAxes_4, 'square');

app.UIAxes_4.DataAspectRatio = [1 1 1];

```

```

% Calculate PSNR

```

```

MAX = 255; % Maximum possible pixel value for double image

```

```

MSE = mean((ori - recons).^2, 'all');

```

```
PSNR = 10 * log10(MAX^2 / MSE);
```

```
% Clip PSNR to the range [0, 1]
```

```
PSNR = max(min(PSNR, 255), 0);
```

```
ssimValue = ssim(recons, ori);
```

```
% Update the PSNR and SSIM values in the app
```

```
app.PSNREditField.Value = PSNR;
```

```
app.SSIMEditField.Value = ssimValue;
```

```
end
```

```
% Button pushed function: ARTButton
```

```
function ARTButtonPushed(app, event)
```

```
% Prompt the user for the number of iterations
```

```
prompt = 'Enter the number of ART iterations:';
```

```
numIterations = str2double(inputdlg(prompt, 'ART Iterations', 1, {'200'}));
```

```
if ~isnan(numIterations) && ~isempty(numIterations)
```

```
    % Call the ARTReconstruction function with the specified number of iterations
```

```
    app.ARTReconstruction(numIterations);
```

```
else
```

```
    % Display an error message if the input is invalid
```

```
        errordlg('Invalid input. Please enter a valid number of iterations.', 'Error',  
'modal');
```

```
end
```

```
ori = double(app.originalImage);
```



```

recons = double(flipud(app.reconstructedImageART));

% Calculate PSNR
MAX = 255; % Maximum possible pixel value for double image
MSE = mean((ori - recons).^2, 'all');
PSNR = 10 * log10(MAX^2 / MSE);

% Clip PSNR to the range [0, 1]
PSNR = max(min(PSNR, 255), 0);
ssimValue = ssim(recons, ori);

% Update the PSNR and SSIM values in the app
app.PSNREditField.Value = PSNR;
app.SSIMEditField.Value = ssimValue;
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Color = [0.9412 0.9412 0.9412];
app.UIFigure.Position = [100 100 879 441];

```

```
app.UIFigure.Name = 'MATLAB App';
```

```
% Create sinogram_panel
```

```
app.sinogram_panel = uipanel(app.UIFigure);
```

```
app.sinogram_panel.BackgroundColor = [0.9098 0.9294 1];
```

```
app.sinogram_panel.Position = [334 21 252 191];
```

```
% Create UIAxes_3
```

```
app.UIAxes_3 = uiaxes(app.sinogram_panel);
```

```
title(app.UIAxes_3, 'Sinogram')
```

```
app.UIAxes_3.Position = [1 3 246 183];
```

```
% Create phantom_panel
```

```
app.phantom_panel = uipanel(app.UIFigure);
```

```
app.phantom_panel.BackgroundColor = [0.9098 0.9294 1];
```

```
app.phantom_panel.Position = [334 231 252 190];
```

```
% Create UIAxes
```

```
app.UIAxes = uiaxes(app.phantom_panel);
```

```
title(app.UIAxes, 'Phantom')
```

```
app.UIAxes.Position = [1 4 246 181];
```

```
% Create control_panel
```

```
app.control_panel = uipanel(app.UIFigure);
```

```
app.control_panel.BackgroundColor = [0.9098 0.9294 1];
```

```
app.control_panel.Position = [2 0 315 442];
```

```

% Create PhantomImageReconstructionLabel

app.PhantomImageReconstructionLabel = uilabel(app.control_panel);

app.PhantomImageReconstructionLabel.HorizontalAlignment = 'center';

app.PhantomImageReconstructionLabel.FontSize = 15;

app.PhantomImageReconstructionLabel.FontWeight = 'bold';

app.PhantomImageReconstructionLabel.Position = [40 412 233 22];

app.PhantomImageReconstructionLabel.Text = 'Phantom Image
Reconstruction';

```

```

% Create ChoosePhantomButtonGroup

app.ChoosePhantomButtonGroup = uibuttongroup(app.control_panel);

app.ChoosePhantomButtonGroup.SelectionChangedFcn =
createCallbackFcn(app, @ChoosePhantomButtonGroupSelectionChanged, true);

app.ChoosePhantomButtonGroup.Title = 'Choose Phantom';

app.ChoosePhantomButtonGroup.BackgroundColor = [1 1 1];

app.ChoosePhantomButtonGroup.Position = [16 252 123 115];

```

```

% Create Phantom1Button

app.Phantom1Button = uiradiobutton(app.ChoosePhantomButtonGroup);

app.Phantom1Button.Text = 'Phantom 1';

app.Phantom1Button.Position = [11 69 80 22];

app.Phantom1Button.Value = true;

```

```

% Create Phantom2Button

app.Phantom2Button = uiradiobutton(app.ChoosePhantomButtonGroup);

app.Phantom2Button.Text = 'Phantom 2';

app.Phantom2Button.Position = [11 47 80 22];

```

```

% Create Phantom3Button

app.Phantom3Button = uiradiobutton(app.ChoosePhantomButtonGroup);

app.Phantom3Button.Text = 'Phantom 3';

app.Phantom3Button.Position = [11 25 80 22];


% Create dotButton

app.dotButton = uiradiobutton(app.ChoosePhantomButtonGroup);

app.dotButton.Visible = 'off';

app.dotButton.Text = "";

app.dotButton.Position = [91 26 25 22];


% Create Phantom4Button

app.Phantom4Button = uiradiobutton(app.ChoosePhantomButtonGroup);

app.Phantom4Button.Text = 'Phantom 4';

app.Phantom4Button.Position = [11 5 80 22];


% Create ShowPhantomButton

app.ShowPhantomButton = uibutton(app.control_panel, 'push');

        app.ShowPhantomButton.ButtonPushedFcn = createCallbackFcn(app,
@ShowPhantomButtonPushed, true);

app.ShowPhantomButton.BackgroundColor = [1 1 1];

app.ShowPhantomButton.Position = [15 215 123 24];

app.ShowPhantomButton.Text = 'Show Phantom';


% Create NoisyLevelSliderLabel

app.NoisyLevelSliderLabel = uilabel(app.control_panel);

app.NoisyLevelSliderLabel.Position = [155 341 68 22];

```

```
app.NoisyLevelSliderLabel.Text = 'Noisy Level';
```

```
% Create NoisyLevelSlider
```

```
app.NoisyLevelSlider = uislider(app.control_panel);
```

```
app.NoisyLevelSlider.Limits = [0 5];
```

```
app.NoisyLevelSlider.FontSize = 10;
```

```
app.NoisyLevelSlider.Position = [155 335 138 3];
```

```
% Create ShowEditedPhantomButton
```

```
app.ShowEditedPhantomButton = uibutton(app.control_panel, 'push');
```

```
    app.ShowEditedPhantomButton.ButtonPushedFcn = createCallbackFcn(app,  
@ShowEditedPhantomButtonPushed, true);
```

```
app.ShowEditedPhantomButton.BackgroundColor = [1 1 1];
```

```
app.ShowEditedPhantomButton.Position = [157 215 134 24];
```

```
app.ShowEditedPhantomButton.Text = 'Show Edited Phantom';
```

```
% Create BlurLevelSliderLabel
```

```
app.BlurLevelSliderLabel = uilabel(app.control_panel);
```

```
app.BlurLevelSliderLabel.Position = [154 286 59 22];
```

```
app.BlurLevelSliderLabel.Text = 'Blur Level';
```

```
% Create BlurLevelSlider
```

```
app.BlurLevelSlider = uislider(app.control_panel);
```

```
app.BlurLevelSlider.Limits = [0 5];
```

```
app.BlurLevelSlider.FontSize = 10;
```

```
app.BlurLevelSlider.Position = [154 280 138 3];
```

```

% Create SinogramButton

app.SinogramButton = uibutton(app.control_panel, 'push');

        app.SinogramButton.ButtonPushedFcn = createCallbackFcn(app,
@SinogramButtonPushed, true);

app.SinogramButton.BackgroundColor = [1 1 1];
app.SinogramButton.Position = [163 178 123 24];
app.SinogramButton.Text = 'Sinogram';


% Create OriginalSinogramButton

app.OriginalSinogramButton = uibutton(app.control_panel, 'push');

        app.OriginalSinogramButton.ButtonPushedFcn = createCallbackFcn(app,
@OriginalSinogramButtonPushed, true);

app.OriginalSinogramButton.BackgroundColor = [1 1 1];
app.OriginalSinogramButton.Position = [15 178 123 24];
app.OriginalSinogramButton.Text = 'Original Sinogram';


% Create ChooseFilterButtonGroup

app.ChooseFilterButtonGroup = uibuttongroup(app.control_panel);

        app.ChooseFilterButtonGroup.SelectionChangedFcn = createCallbackFcn(app,
@ChooseFilterButtonGroupSelectionChanged, true);

app.ChooseFilterButtonGroup.Title = 'Choose Filter';
app.ChooseFilterButtonGroup.BackgroundColor = [1 1 1];
app.ChooseFilterButtonGroup.Position = [15 21 123 141];


% Create noneButton

app.noneButton = uiradiobutton(app.ChooseFilterButtonGroup);

app.noneButton.Text = 'none';

app.noneButton.Position = [11 97 49 22];

```

```
app.noneButton.Value = true;
```

```
% Create RamLakButton
```

```
app.RamLakButton = uiradiobutton(app.ChooseFilterButtonGroup);
```

```
app.RamLakButton.Text = 'Ram-Lak';
```

```
app.RamLakButton.Position = [11 75 71 22];
```

```
% Create HammingButton
```

```
app.HammingButton = uiradiobutton(app.ChooseFilterButtonGroup);
```

```
app.HammingButton.Text = 'Hamming';
```

```
app.HammingButton.Position = [11 53 73 22];
```

```
% Create dot2Button
```

```
app.dot2Button = uiradiobutton(app.ChooseFilterButtonGroup);
```

```
app.dot2Button.Visible = 'off';
```

```
app.dot2Button.Text = '';
```

```
app.dot2Button.Position = [91 53 25 22];
```

```
% Create HanningButton
```

```
app.HanningButton = uiradiobutton(app.ChooseFilterButtonGroup);
```

```
app.HanningButton.Text = 'Hanning';
```

```
app.HanningButton.Position = [11 31 67 22];
```

```
% Create CosineButton
```

```
app.CosineButton = uiradiobutton(app.ChooseFilterButtonGroup);
```

```
app.CosineButton.Text = 'Cosine';
```

```
app.CosineButton.Position = [11 10 59 22];
```

```

% Create BackProjectionButton

app.BackProjectionButton = uibutton(app.control_panel, 'push');

    app.BackProjectionButton.ButtonPushedFcn = createCallbackFcn(app,
@BackProjectionButtonPushed, true);

app.BackProjectionButton.BackgroundColor = [1 1 1];
app.BackProjectionButton.Position = [163 138 123 24];
app.BackProjectionButton.Text = 'Back Projection';

% Create ProjekAkhirTeknologiPencitraanMedisKelompok1Label

    app.ProjekAkhirTeknologiPencitraanMedisKelompok1Label =
uicontrol(app.control_panel);

app.ProjekAkhirTeknologiPencitraanMedisKelompok1Label.HorizontalAlignment =
'center';

    app.ProjekAkhirTeknologiPencitraanMedisKelompok1Label.FontSize = 11;

    app.ProjekAkhirTeknologiPencitraanMedisKelompok1Label.Position = [17 393
279 22];

    app.ProjekAkhirTeknologiPencitraanMedisKelompok1Label.Text = 'Projek
Akhir Teknologi Pencitraan Medis - Kelompok 1';

% Create PSNREditFieldLabel

app.PSNREditFieldLabel = uicontrol(app.control_panel);
app.PSNREditFieldLabel.HorizontalAlignment = 'right';
app.PSNREditFieldLabel.Position = [161 69 39 22];
app.PSNREditFieldLabel.Text = 'PSNR';

% Create PSNREditField

app.PSNREditField = uicontrol(app.control_panel, 'numeric');

```



```

app.PSNREditField.Position = [214 69 71 22];

% Create SSIMEditFieldLabel

app.SSIMEditFieldLabel = uilabel(app.control_panel);
app.SSIMEditFieldLabel.HorizontalAlignment = 'right';
app.SSIMEditFieldLabel.Position = [166 39 35 22];
app.SSIMEditFieldLabel.Text = 'SSIM';

% Create SSIMEditField

app.SSIMEditField = uieditfield(app.control_panel, 'numeric');
app.SSIMEditField.Position = [215 39 71 22];

% Create VirginiaValensieneEMariscaAuliaKizzaNaufalGiselaBeaAthaliaLabel

app.VirginiaValensieneEMariscaAuliaKizzaNaufalGiselaBeaAthaliaLabel =
uilabel(app.control_panel);

app.VirginiaValensieneEMariscaAuliaKizzaNaufalGiselaBeaAthaliaLabel.HorizontalA
lignment = 'center';

app.VirginiaValensieneEMariscaAuliaKizzaNaufalGiselaBeaAthaliaLabel.FontSize =
8;

app.VirginiaValensieneEMariscaAuliaKizzaNaufalGiselaBeaAthaliaLabel.Position =
[17 377 279 22];

app.VirginiaValensieneEMariscaAuliaKizzaNaufalGiselaBeaAthaliaLabel.Text
= 'Virginia Valensiene E, Marisca Aulia, Kizza Naufal, Gisela Bea Athalia Adyu';

% Create ARTButton

app.ARTButton = uibutton(app.control_panel, 'push');

```

```
app.ARTButton.ButtonPushedFcn = createCallbackFcn(app,  
@ARTButtonPushed, true);
```

```
app.ARTButton.BackgroundColor = [1 1 1];
```

```
app.ARTButton.Position = [166 105 123 24];
```

```
app.ARTButton.Text = 'ART';
```

```
% Create edited_phantom_panel
```

```
app.edited_phantom_panel = uipanel(app.UIFigure);
```

```
app.edited_phantom_panel.BackgroundColor = [0.9098 0.9294 1];
```

```
app.edited_phantom_panel.Position = [604 231 251 190];
```

```
% Create UIAxes_2
```

```
app.UIAxes_2 = uiaxes(app.edited_phantom_panel);
```

```
title(app.UIAxes_2, 'Edited Phantom')
```

```
app.UIAxes_2.Position = [1 4 249 181];
```

```
% Create reconstruction_panel
```

```
app.reconstruction_panel = uipanel(app.UIFigure);
```

```
app.reconstruction_panel.BackgroundColor = [0.9098 0.9294 1];
```

```
app.reconstruction_panel.Position = [604 21 251 191];
```

```
% Create UIAxes_4
```

```
app.UIAxes_4 = uiaxes(app.reconstruction_panel);
```

```
title(app.UIAxes_4, 'Reconstruction Image')
```

```
app.UIAxes_4.Position = [2 3 248 183];
```

```
% Show the figure after all components are created
```

```

        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = Kelompok_1_Image_Reconstruction

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

```

```
        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end
```