

**PROYEK TUGAS AKHIR SEMESTER FISIKA PENCITRAAN MEDIS**

**ANALISIS SEGMENTASI JANTUNG MENGGUNAKAN  
ALGORITMA MARKER-CONTROLLED WATERSHED**

Diajukan Sebagai Tugas Akhir Semester Mata Kuliah Fisika Pencitraan Medis

Jurusan Fisika Fakultas Sains Dan Teknologi



**Oleh:**

**Annisa Nurlaili Aulia Safitri**

**(NIM. 1217030006)**

**JURUSAN FISIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUNAN GUNUNG DJATI BANDUNG  
2024**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Jantung adalah organ vital yang memiliki peran utama dalam menjaga kelangsungan hidup dengan mengatur sirkulasi darah di seluruh tubuh. Setiap kontraksi jantung memungkinkan oksigen dan nutrisi didistribusikan ke jaringan tubuh, sementara limbah metabolik seperti karbon dioksida dibuang melalui sistem pernapasan. Struktur jantung terdiri dari empat ruang utama: atrium kanan, atrium kiri, ventrikel kanan, dan ventrikel kiri. Atrium bertugas menerima darah dari tubuh dan paru-paru, sedangkan ventrikel memompa darah ke seluruh tubuh melalui arteri. Keempat ruang ini diatur oleh sistem katup yang memastikan aliran darah berjalan satu arah, sehingga efisiensi fungsi jantung dapat terjaga.

Computed Tomography (CT) adalah salah satu modalitas pencitraan medis yang sangat bermanfaat dalam menghasilkan gambaran tiga dimensi (3D) anatomi jantung. CT menggunakan sinar-X untuk mendapatkan citra penampang melintang tubuh, yang kemudian direkonstruksi menjadi model 3D. Teknik ini memungkinkan visualisasi detail atrium kiri dan vena pulmonalis dengan resolusi tinggi. Untuk mendapatkan citra yang akurat, protokol berbasis ECG-gating sering digunakan. Protokol ini meminimalkan artefak gerakan akibat denyut jantung, sehingga menghasilkan citra yang tajam dan informatif.

Namun, agar citra CT dapat digunakan secara optimal, diperlukan proses segmentasi untuk memisahkan struktur penting seperti atrium kiri dan vena pulmonalis dari bagian jantung lainnya. Segmentasi adalah proses yang memisahkan objek atau wilayah tertentu dalam citra berdasarkan intensitas piksel atau karakteristik lainnya. Dalam jantung, segmentasi menjadi tantangan karena struktur jantung yang kompleks dan variasi intensitas antar bagian. Metode segmentasi tradisional seperti region growing atau active contour models sering kali tidak cukup efektif untuk menangani struktur yang kompleks seperti vena pulmonalis.

Untuk mengatasi keterbatasan tersebut, algoritma watershed dan marker-controlled watershed sering digunakan. Watershed adalah algoritma segmentasi berbasis morfologi yang menganggap citra sebagai peta topografi, di mana piksel dengan intensitas rendah direpresentasikan sebagai

lembah dan piksel dengan intensitas tinggi sebagai puncak. Watershed membagi citra menjadi wilayah-wilayah berdasarkan gradien intensitas, tetapi sering kali menghasilkan oversegmentasi, terutama pada citra dengan kebisingan tinggi. Untuk mengatasi masalah ini, maka metode marker-controlled watershed sangat diperlukan. Metode ini menggunakan penanda atau marker untuk mengontrol proses segmentasi, sehingga hanya area yang relevan yang diproses. Marker dapat ditentukan secara manual atau otomatis berdasarkan informasi awal dari citra. Dengan menggunakan marker-controlled watershed, segmentasi menjadi lebih terarah dan hasilnya lebih akurat, menjadikannya metode yang ideal untuk memisahkan struktur kompleks seperti atrium kiri dan vena pulmonalis.

## **1.2 Tujuan**

1. Menganalisis dan mengidentifikasi struktur jantung.
2. Mengembangkan metode segmentasi yang akurat.

## **BAB II**

### **TEORI DASAR**

#### **2.1 Anatomi Jantung**

Jantung adalah organ berotot yang terletak di rongga toraks, tepat di belakang tulang dada (sternum) dan diapit oleh kedua paru-paru. Ukurannya kira-kira sebesar kepalan tangan, dengan berat sekitar 250–300 gram pada wanita dan 300–350 gram pada pria. Organ ini memainkan peran sentral dalam sistem kardiovaskular, berfungsi sebagai pompa yang mengalirkan darah ke seluruh tubuh untuk mendistribusikan oksigen dan nutrisi serta membuang limbah metabolik, seperti karbon dioksida.

Jantung dibagi menjadi empat ruang utama: atrium kanan, atrium kiri, ventrikel kanan, dan ventrikel kiri. Setiap ruang memiliki fungsi spesifik dalam sirkulasi darah:

##### **a. Atrium Kanan**

Atrium kanan adalah ruang atas di sisi kanan jantung yang menerima darah miskin oksigen dari tubuh. Darah ini dikembalikan ke jantung melalui dua vena utama:

1. Vena Cava Superior: Membawa darah dari bagian atas tubuh, termasuk kepala, leher, dan lengan.
2. Vena Cava Inferior: Membawa darah dari bagian bawah tubuh, termasuk perut, panggul, dan tungkai.

Darah yang masuk ke atrium kanan kaya akan karbon dioksida dan produk limbah lainnya karena telah digunakan oleh jaringan tubuh untuk proses metabolisme. Atrium kanan mengalirkan darah ini ke ventrikel kanan melalui katup trikuspid.

##### **b. Atrium Kiri**

Atrium kiri adalah ruang atas di sisi kiri jantung yang menerima darah kaya oksigen dari paru-paru. Darah ini tiba di atrium kiri melalui empat vena pulmonalis:

1. Dua vena dari paru-paru kanan.
2. Dua vena dari paru-paru kiri.

Darah yang masuk ke atrium kiri telah menjalani proses oksigenasi di paru-paru selama pertukaran gas. Dari atrium kiri, darah mengalir ke ventrikel kiri melalui katup mitral.

#### c. Ventrikel Kanan

Ventrikel kanan adalah ruang bawah di sisi kanan jantung yang bertugas memompa darah ke paru-paru untuk proses oksigenasi. Darah yang diterima dari atrium kanan melewati katup trikuspid ke ventrikel kanan. Saat ventrikel kanan berkontraksi, darah dipompa ke paru-paru melalui arteri pulmonalis. Aliran darah ini melewati katup pulmonal untuk memastikan darah hanya mengalir ke arah paru-paru dan tidak kembali ke ventrikel kanan.

Di paru-paru, darah melepaskan karbon dioksida dan mengambil oksigen melalui proses pertukaran gas di alveolus.

#### d. Ventrikel Kiri

Ventrikel kiri adalah ruang bawah di sisi kiri jantung yang bertugas memompa darah kaya oksigen ke seluruh tubuh. Ruang ini menerima darah dari atrium kiri melalui katup mitral. Dengan dinding otot yang lebih tebal dibandingkan ventrikel kanan, ventrikel kiri menghasilkan tekanan yang cukup kuat untuk mendorong darah ke seluruh tubuh melalui aorta, arteri terbesar di tubuh manusia.

Aorta bercabang menjadi arteri-arteri yang lebih kecil, yang kemudian mengantarkan darah ke organ-organ, otot, dan jaringan di seluruh tubuh. Setelah oksigen dan nutrisi digunakan oleh jaringan, darah kembali ke atrium kanan untuk memulai siklus lagi.

#### e. Sistem Katup Jantung

Jantung memiliki empat katup utama yang memastikan aliran darah berjalan dalam satu arah dan mencegah aliran balik:

1. Katup Trikuspid: Terletak antara atrium kanan dan ventrikel kanan, mengatur aliran darah dari atrium kanan ke ventrikel kanan.
2. Katup Mitral: Terletak antara atrium kiri dan ventrikel kiri, mengatur aliran darah dari atrium kiri ke ventrikel kiri.
3. Katup Pulmonal: Terletak antara ventrikel kanan dan arteri pulmonalis, mencegah darah kembali ke ventrikel kanan setelah dipompa ke paru-paru.
4. Katup Aorta: Terletak antara ventrikel kiri dan aorta, mencegah darah kembali ke ventrikel kiri setelah dipompa ke seluruh tubuh.

Sistem katup ini bekerja seiring dengan kontraksi dan relaksasi jantung untuk memastikan bahwa darah terus mengalir dengan efisien melalui sirkulasi pulmonal (paru-paru) dan sistemik (seluruh tubuh).

## **2.2 Computed Tomography**

Computed Tomography (CT) adalah teknologi pencitraan medis yang memanfaatkan sinar-X untuk menghasilkan gambar penampang melintang tubuh secara detail. Proses kerja CT melibatkan pemindaian tubuh pasien dari berbagai sudut menggunakan sumber sinar-X yang berputar di sekitar pasien. Sinar-X ini melewati tubuh dan sebagian energinya diserap oleh jaringan, sementara sisanya diteruskan. Detektor di sisi berlawanan menangkap sinar yang diteruskan dan mengukur intensitasnya. Data ini kemudian diproses oleh komputer untuk membuat gambar dua dimensi (2D) dari potongan tubuh. Dengan mengambil serangkaian gambar 2D dari berbagai sudut, komputer dapat merekonstruksi model tiga dimensi (3D) yang memberikan visualisasi menyeluruh tentang struktur internal tubuh.

Teknologi CT memiliki kemampuan yang sangat penting dalam dunia kedokteran, khususnya dalam pencitraan jantung. Resolusi spasial yang tinggi dari CT memungkinkan deteksi struktur kecil dan detail anatomi kompleks dengan akurasi tinggi. Dalam konteks jantung, struktur seperti atrium kiri, vena pulmonalis, dan arteri koroner dapat divisualisasikan dengan jelas. Detil ini sangat penting untuk mendiagnosis kelainan seperti penyumbatan pada pembuluh darah, aneurisma, atau malformasi lainnya.

Salah satu tantangan utama dalam pencitraan jantung adalah gerakan konstan yang dihasilkan oleh kontraksi dan relaksasi jantung. Untuk mengatasi ini, protokol ECG-gating digunakan selama pemindaian. Dengan merekam aktivitas listrik jantung menggunakan elektrokardiogram (ECG), CT dapat menyinkronkan pengambilan gambar dengan fase tertentu dari siklus jantung, seperti diastol, saat jantung berada dalam kondisi relaksasi maksimal. Teknik ini meminimalkan artefak gerakan yang dapat mengaburkan detail gambar, sehingga meningkatkan kejelasan dan keakuratan hasil pencitraan.

Selain itu, CT memungkinkan pengambilan gambar dengan sangat cepat, terutama dengan teknologi pemindai multi-slice atau dual-source CT. Dalam hitungan detik, ratusan gambar dapat dihasilkan, mencakup seluruh jantung. Kecepatan ini sangat penting, terutama untuk pasien dengan detak jantung tidak teratur atau tinggi. Dengan bantuan zat kontras berbasis yodium, visualisasi pembuluh darah, dinding jantung, dan jaringan di sekitarnya dapat diperjelas. Zat kontras ini meningkatkan perbedaan antara berbagai struktur, sehingga membantu dokter dalam menganalisis kondisi perfusi darah dan fungsi jantung secara keseluruhan.

CT juga digunakan secara luas dalam berbagai aplikasi kardiologi, termasuk Coronary CT Angiography (CCTA) untuk mendeteksi penyempitan atau penyumbatan arteri koroner, pemetaan anatomi jantung untuk perencanaan prosedur ablasi kateter, dan penilaian skor kalsium untuk mengidentifikasi risiko penyakit jantung koroner. Dengan kemampuan untuk merekonstruksi gambar dalam berbagai sudut pandang, termasuk model volumetrik tiga dimensi, CT memberikan wawasan mendalam tentang hubungan spasial antara struktur jantung dan pembuluh darah.

### **2.3 Segmentasi**

Segmentasi adalah proses mendasar dalam analisis citra medis yang bertujuan untuk memisahkan objek atau wilayah tertentu dari citra agar dapat dianalisis lebih lanjut. Dalam konteks medis, segmentasi membantu dokter atau peneliti memfokuskan perhatian pada struktur tertentu, seperti organ atau jaringan yang relevan dengan diagnosis atau pengobatan. Pada pencitraan jantung, segmentasi digunakan untuk mengisolasi struktur penting seperti atrium kiri, ventrikel, dan vena pulmonalis, sehingga memberikan informasi yang sangat dibutuhkan untuk prosedur diagnostik dan intervensi, termasuk ablasi kateter frekuensi radio (RFCA).

Proses segmentasi bergantung pada berbagai parameter, seperti intensitas piksel, gradien, tekstur, atau fitur lainnya yang dapat dibedakan dalam citra. Hasil segmentasi yang berhasil adalah pemisahan struktur yang jelas dan tepat sesuai kebutuhan klinis. Sebagai contoh, pada citra jantung, atrium kiri dan vena pulmonalis sering menjadi fokus karena peran mereka dalam berbagai gangguan kardiovaskular, termasuk fibrilasi atrium (AF).

Namun, segmentasi pada pencitraan jantung menghadapi sejumlah tantangan khusus. Salah satu tantangan utama adalah variasi anatomi individu. Setiap pasien memiliki bentuk, ukuran, dan posisi anatomi jantung yang unik. Atrium kiri dan vena pulmonalis, misalnya, dapat menunjukkan perbedaan signifikan dalam dimensi, geometri, dan orientasi. Variasi ini tidak hanya memengaruhi hasil segmentasi tetapi juga membuat pendekatan berbasis satu ukuran untuk semua menjadi tidak efektif.

Selain itu, kesamaan intensitas antarstruktur jantung juga menjadi tantangan. Pada citra medis seperti CT atau MRI, banyak struktur jantung memiliki intensitas piksel yang serupa karena komposisi jaringan yang mirip. Misalnya, dinding atrium kiri mungkin memiliki tingkat intensitas yang mendekati jaringan sekitarnya seperti dinding ventrikel atau pembuluh darah besar. Kesamaan ini menyulitkan algoritma segmentasi berbasis threshold, yang bergantung pada perbedaan intensitas untuk memisahkan wilayah yang berbeda dalam citra.

Kesulitan segmentasi juga diperburuk oleh artefak pencitraan, seperti noise (kebisingan), blur (kabur), atau distorsi gerakan yang dihasilkan oleh denyut jantung atau pernapasan. Artefak ini dapat mengaburkan batas antara struktur yang berdekatan, sehingga membuat proses segmentasi menjadi lebih rumit.

Untuk mengatasi tantangan ini, berbagai pendekatan segmentasi telah dikembangkan. Beberapa pendekatan yang sering digunakan termasuk:

1. Region-Based Segmentation: Metode ini memulai dari titik awal tertentu (seed point) dan meluas ke piksel tetangga yang memiliki karakteristik serupa, seperti intensitas.
2. Model Aktif: Menggunakan kontur atau kurva fleksibel yang menyesuaikan bentuknya untuk mengikuti tepi struktur berdasarkan informasi gradien.



3. Metode Berbasis Gradien: Mendeteksi batas struktur berdasarkan perubahan intensitas yang tajam.

Metode yang lebih canggih, seperti marker-controlled watershed, menggunakan kombinasi teknik berbasis marker untuk membatasi wilayah segmentasi, sehingga mengurangi risiko oversegmentasi akibat variasi intensitas atau noise.

## **2.4 Algoritma Watershed**

Algoritma watershed didasarkan pada konsep bahwa citra dapat dianggap sebagai peta topografi, di mana intensitas piksel diinterpretasikan sebagai ketinggian:

- Piksel dengan intensitas rendah dianggap sebagai lembah atau catchment basins.
- Piksel dengan intensitas tinggi dianggap sebagai puncak atau ridge lines.

Proses segmentasi dimulai dengan mengidentifikasi lembah yang terpisah dalam citra. Ketika air "mengisi" lembah ini, wilayah-wilayah dengan intensitas rendah secara alami membentuk catchment basins. Garis pemisah antara dua lembah yang berdekatan dikenal sebagai watershed lines. Dengan prinsip ini, watershed dapat memisahkan objek dalam citra berdasarkan perbedaan intensitas atau gradien.

Keunggulan utama dari algoritma ini adalah kemampuannya untuk mendeteksi tepi atau batas-batas objek dengan baik. Namun, algoritma watershed memiliki kelemahan utama, yaitu rentan terhadap oversegmentasi. Pada citra dengan tingkat kebisingan tinggi atau perbedaan intensitas yang kecil, algoritma ini cenderung menghasilkan terlalu banyak wilayah segmentasi. Hal ini membuat hasil menjadi tidak praktis untuk analisis medis.

## **2.5 Marker-Controlled Watershed**

Untuk mengatasi kelemahan oversegmentasi dalam algoritma watershed, metode marker-controlled watershed diperkenalkan. Pendekatan ini menggunakan marker sebagai penanda awal untuk membatasi wilayah yang akan disegmentasi, sehingga menghasilkan hasil yang lebih terarah dan akurat. Marker dapat dihasilkan secara manual oleh pengguna berdasarkan pengetahuan sebelumnya tentang citra, atau secara otomatis melalui algoritma pendukung.

Proses marker-controlled watershed terdiri dari beberapa langkah utama:

1. **Pembuatan Marker:** Marker adalah titik awal yang digunakan untuk menentukan wilayah yang relevan dalam citra. Pada citra jantung, marker sering ditempatkan pada struktur penting seperti atrium kiri, vena pulmonalis, atau dinding ventrikel. Marker ini membantu membatasi proses segmentasi hanya pada area yang diinginkan, mengurangi risiko oversegmentasi.
2. **Pengolahan Gradien:** Setelah marker dibuat, gradien intensitas dihitung untuk menonjolkan batas-batas struktur dalam citra. Gradien ini menunjukkan perubahan intensitas yang tajam antara dua wilayah, sehingga membantu algoritma dalam menentukan batas segmentasi. Teknik ini juga membantu mengatasi tantangan seperti kesamaan intensitas antarstruktur.
3. **Transformasi Watershed:** Algoritma watershed diterapkan pada hasil gradien yang telah diproses, tetapi dengan batasan yang ditentukan oleh marker. Dengan menggunakan marker sebagai panduan, transformasi watershed membagi citra menjadi wilayah yang sesuai dengan struktur anatomi yang diinginkan. Proses ini memastikan bahwa segmentasi hanya terjadi di sekitar area yang telah ditentukan oleh marker, menghilangkan wilayah yang tidak relevan.

Marker-controlled watershed adalah metode segmentasi citra yang menggabungkan kekuatan algoritma watershed dengan pendekatan berbasis marker, menghasilkan segmentasi yang lebih akurat dan efisien. Salah satu keunggulan utama dari metode ini adalah pengendalian terhadap oversegmentasi. Dengan menggunakan marker, algoritma membatasi wilayah yang diproses, sehingga menghindari pembagian yang berlebihan pada citra yang sering terjadi pada algoritma watershed konvensional.

Selain itu, metode ini juga menawarkan presisi yang tinggi, terutama dalam segmentasi struktur yang memiliki intensitas serupa, seperti atrium kiri dan vena pulmonalis dalam pencitraan jantung. Fleksibilitas juga menjadi keuntungan besar dari metode ini, karena marker dapat disesuaikan untuk berbagai jenis citra dan tujuan segmentasi, baik secara manual maupun otomatis. Tak kalah pentingnya, marker-controlled watershed lebih efisien dalam penggunaan

waktu dan sumber daya komputasi, karena dengan membatasi area yang diproses, algoritma ini membutuhkan lebih sedikit sumber daya dibandingkan algoritma watershed konvensional.

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Alat dan Bahan**

<b>No.</b>	<b>Alat dan Bahan</b>	<b>Jumlah</b>
1.	Laptop	1 Buah
2.	Dataset	-
3.	Website Google Colab	-

#### **3.2 Prosedur Penelitian**

Penelitian ini bertujuan untuk mengembangkan metode segmentasi gambar medis berbasis komputer untuk mengidentifikasi struktur jantung pada data pasien. Data yang digunakan berupa gambar medis dalam format PNG, yang dikumpulkan dari folder pasien. Setiap folder berisi serangkaian gambar yang akan diurutkan berdasarkan ID pasien untuk memastikan konsistensi analisis.

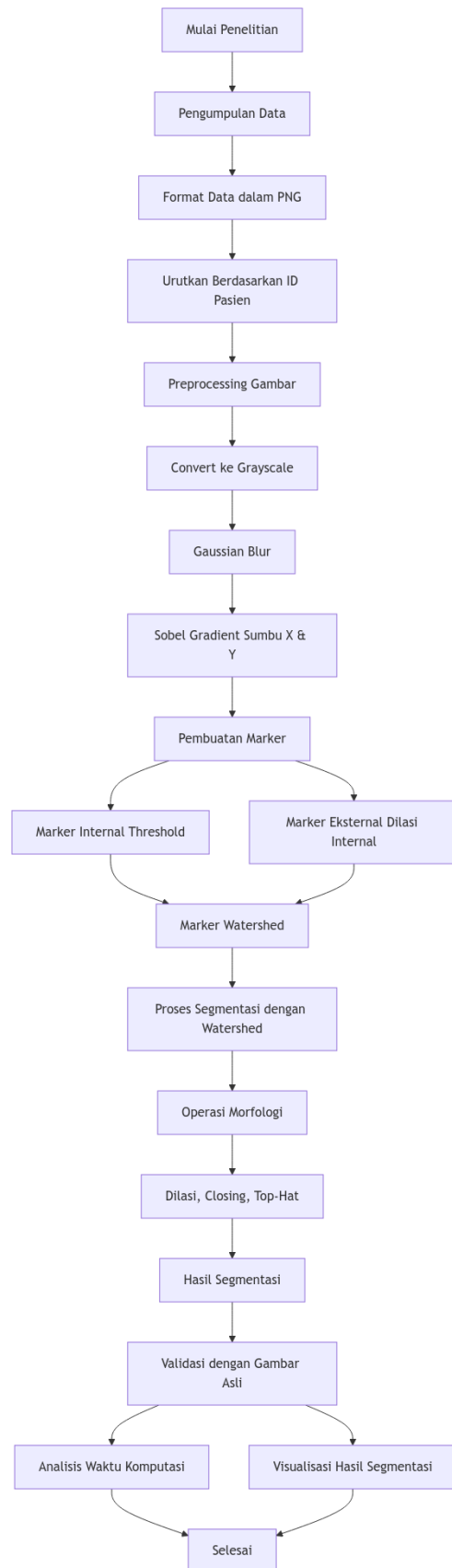
Langkah pertama adalah memproses gambar menjadi grayscale untuk menyederhanakan analisis intensitas, diikuti dengan penerapan Gaussian Blur untuk mengurangi noise sekaligus meningkatkan kejelasan fitur. Setelah itu, gradien Sobel diterapkan untuk menghitung perubahan intensitas pada sumbu X dan Y, menghasilkan gambar yang menonjolkan tepi atau kontur struktur anatomi.

Proses segmentasi dimulai dengan membuat tiga jenis marker: marker internal, eksternal, dan watershed. Marker internal dibuat dengan menggunakan ambang intensitas tertentu untuk menandai area inti struktur. Marker eksternal dihasilkan melalui proses dilasi pada marker internal,

memperluas area yang menjadi kandidat struktur. Gabungan kedua marker ini membentuk marker watershed, yang digunakan sebagai peta label untuk algoritma segmentasi.

Segmentasi utama dilakukan menggunakan algoritma watershed, yang memisahkan struktur berdasarkan gradien intensitas dan marker yang telah dibuat. Operasi morfologi seperti dilasi, closing, dan top-hat diterapkan untuk memperbaiki hasil segmentasi, memastikan struktur yang tersegmentasi akurat dan bebas dari noise atau elemen yang tidak relevan.

Hasil segmentasi kemudian divalidasi dengan membandingkannya terhadap gambar asli untuk memastikan keakuratan dan detail. Proses ini juga melibatkan analisis waktu komputasi pada berbagai iterasi untuk mengevaluasi efisiensi algoritma. Visualisasi dibuat untuk menampilkan berbagai hasil, termasuk gradien Sobel, peta watershed, kontur, filter jantung, dan citra hasil segmentasi akhir. Visualisasi ini tidak hanya membantu memastikan hasil segmentasi, tetapi juga memberikan wawasan mengenai bagaimana struktur jantung teridentifikasi secara otomatis.



## BAB IV

### DATA DAN PEMBAHASAN

#### 4.1 Kode Program

```
[3] %matplotlib inline

import numpy as np
import pandas as pd
import pydicom
import os
import scipy.ndimage as ndimage
from skimage import measure, morphology, segmentation
import matplotlib.pyplot as plt
from skimage.segmentation import watershed

import time
```

```
▶ INPUT_FOLDER = '/content/pasien'

patients = os.listdir(INPUT_FOLDER)
patients.sort()

print("Some examples of patient IDs:")
print(",\n".join(patients[:6]))
```

```
[181] def load_image(path):
    """
    Loads PNG image as grayscale.

    Parameters: path (Image path)

    Returns: image (NumPy array)
    """
    from skimage.io import imread
    image = imread(path, as_gray=True)
    return image

patient_folder_path = os.path.join(INPUT_FOLDER, patients[5]) # Path pasien pertama
for file in os.listdir(patient_folder_path):
    file_path = os.path.join(patient_folder_path, file)
    try:
        image = load_image(file_path)
        print(f"Successfully read: {file}")
    except Exception as e:
        print(f"Error reading {file}: {e}")
```

```

[182] # Memastikan pasien tersedia
if len(patients) > 0:
    patient_folder_path = os.path.join(INPUT_FOLDER, patients[5])

    # Memuat data PNG
    test_patient_image = load_image(os.path.join(patient_folder_path, os.listdir(patient_folder_path)[0]))

    if test_patient_image is not None:
        plt.imshow(test_patient_image, cmap='gray')
        plt.title("Input Image")
        plt.savefig('slice_index.png') # Simpan sebagai PNG
        plt.show()

```

```

[203] def generate_markers(image):
    """
    Generates markers for a given image.

    Parameters:
    image (numpy array): Input image for marker generation.

    Returns:
    tuple: Internal Marker, External Marker, and Watershed Marker.
    """

    # Creation of the internal Marker
    marker_internal = image < 0.10 # Threshold for internal marker
    marker_internal = segmentation.clear_border(marker_internal) # Remove border artifacts
    marker_internal_labels = measure.label(marker_internal) # Label connected components

    # Sort regions by area and remove the smallest regions
    areas = [r.area for r in measure.regionprops(marker_internal_labels)]
    areas.sort()

    if len(areas) > 2:
        # Menghitung indeks tengah
        middle_index = len(areas) // -2
        middle_value = areas[middle_index]

        for region in measure.regionprops(marker_internal_labels):
            if region.area < middle_value: # Membandingkan dengan nilai tengah
                for coordinates in region.coords:
                    marker_internal_labels[coordinates[0], coordinates[1]] = 0 # Hapus wilayah kecil

```



```
[203]     marker_internal = marker_internal_labels > 0 # Keep only labeled regions

        # Creation of the External Marker
        external_a = ndimage.binary_dilation(marker_internal, iterations=10) # Dilate internal marker
        external_b = ndimage.binary_dilation(marker_internal, iterations=55) # Dilate further
        marker_external = external_b ^ external_a # XOR operation to create external marker

        # Creation of the Watershed Marker
        marker_watershed = np.zeros(image.shape, dtype=int) # Initialize empty marker (use int for dtype)
        marker_watershed += marker_internal * 200 # Internal marker in watershed
        marker_watershed += marker_external * 100 # External marker in watershed

        return marker_internal, marker_external, marker_watershed

[204] # Use the input image for marker generation
test_patient_internal, test_patient_external, test_patient_watershed = generate_markers(test_patient_image)

# Plot the markers
f, (ax1, ax2, ax3) = plt.subplots(1, 3, sharey=True, figsize=(15,15))

ax1.imshow(test_patient_internal, cmap='gray')
ax1.set_title("Internal Marker")
ax1.axis('off')

ax2.imshow(test_patient_external, cmap='gray')
ax2.set_title("External Marker")
ax2.axis('off')

ax3.imshow(test_patient_watershed, cmap='gray')
ax3.set_title("Watershed Marker")
```

```
▶ # Use the input image for marker generation
test_patient_internal, test_patient_external, test_patient_watershed = generate_markers(test_patient_image)

# Plot the markers
f, (ax1, ax2, ax3) = plt.subplots(1, 3, sharey=True, figsize=(15,15))

ax1.imshow(test_patient_internal, cmap='gray')
ax1.set_title("Internal Marker")
ax1.axis('off')

ax2.imshow(test_patient_external, cmap='gray')
ax2.set_title("External Marker")
ax2.axis('off')

ax3.imshow(test_patient_watershed, cmap='gray')
ax3.set_title("Watershed Marker")
ax3.axis('off')

plt.savefig('markers_output.png') # Simpan sebagai PNG
```

```
[205] # Lists to store computation times and iterations
      computation_times = []
      iteration_titles = []
```

```
▶ def segment_cardiac(image, iterations=1):
    """
    Segments cardiac region using various techniques.

    Parameters: image (Scan image), iterations (more iterations, more accurate mask)

    Returns:
    - Segmented Cardiac
    - Cardiac Filter
    - Outline Cardiac
    - Watershed Cardiac
    - Sobel Gradient
    """

    # Store the start time
    start = time.time()

    marker_internal, marker_external, marker_watershed = generate_markers(image)

    ...

    Creation of Sobel Gradient
    ...
```

```
# Store the start time
start = time.time()

marker_internal, marker_external, marker_watershed = generate_markers(image)

...

Creation of Sobel Gradient
...

# Sobel-Gradient
sobel_filtered_dx = ndimage.sobel(image, 1)
sobel_filtered_dy = ndimage.sobel(image, 0)
sobel_gradient = np.hypot(sobel_filtered_dx, sobel_filtered_dy)
sobel_gradient *= 255.0 / np.max(sobel_gradient)

...

Using the watershed algorithm
We pass the image convoluted by sobel operator and the watershed marker
to watershed from skimage.segmentation to get a matrix labeled using the watershed segmentation algorithm.
...

watershed_result = watershed(sobel_gradient, marker_watershed)

...

Reducing the image to outlines after Watershed algorithm
...

outline = ndimage.morphological_gradient(watershed_result, size=(3,3))
outline = outline.astype(bool)

...

Black Top-hat Morphology:
...
```



```
# Structuring element used for the filter
blackhat_struct = [[0, 0, 1, 1, 1, 0, 0],
                   [0, 1, 1, 1, 1, 1, 0],
                   [1, 1, 1, 1, 1, 1, 1],
                   [1, 1, 1, 1, 1, 1, 1],
                   [1, 1, 1, 1, 1, 1, 1],
                   [0, 1, 1, 1, 1, 1, 0],
                   [0, 0, 1, 1, 1, 0, 0]]

blackhat_struct = ndimage.iterate_structure(blackhat_struct, iterations)

# Perform Black Top-hat filter
outline += ndimage.black_tophat(outline, structure=blackhat_struct)

...
Generate cardiac filter using internal marker and outline.
...
cardiac_filter = np.bitwise_or(marker_internal, outline)
cardiac_filter = ndimage.binary_closing(cardiac_filter, structure=np.ones((5,5)), iterations=3)

...
Segment cardiac region using cardiac filter and the image.
...

# Ensure both arrays have the same shape
if image.shape != cardiac_filter.shape:
    image = np.resize(image, cardiac_filter.shape)

# Segment the image
segmented = np.where(cardiac_filter == 1, image, -2000*np.ones(cardiac_filter.shape))
```

```
[206] # Append computation time
      end = time.time()
      computation_times.append(end - start)
      iteration_titles.append("{num} iterations".format(num=iterations))

      return segmented, cardiac_filter, outline, watershed_result, sobel_gradient
```

```
[207] import cv2
      import numpy as np
      import matplotlib.pyplot as plt

      # Baca gambar asli
      image = cv2.imread('/content/pasien/5/IM-0155-0001.dcm.png', cv2.IMREAD_GRAYSCALE)

      # Terapkan Gaussian Blur untuk mengurangi noise
      blurred_image = cv2.GaussianBlur(image, (5, 5), 0)

      # Terapkan Sobel Gradient
      sobel_x = cv2.Sobel(blurred_image, cv2.CV_64F, 1, 0, ksize=3)
      sobel_y = cv2.Sobel(blurred_image, cv2.CV_64F, 0, 1, ksize=3)
      sobel_gradient = cv2.magnitude(sobel_x, sobel_y)

      # Normalisasi hasil Sobel Gradient
      sobel_normalized = cv2.normalize(sobel_gradient, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)
```

```
[207] # Plot hasil
plt.figure(figsize=(10, 5))
plt.subplot(1, 3, 1)
plt.title("Original Image")
plt.imshow(image, cmap='gray')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.title("Blurred Image")
plt.imshow(blurred_image, cmap='gray')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.title("Sobel Gradient")
plt.imshow(sobel_normalized, cmap='gray')
plt.axis('off')

plt.tight_layout()
plt.show()
```

```
# Periksa bentuk gambar
if test_patient_image.ndim == 3: # Jika memiliki dimensi batch
    single_image = test_patient_image[0]
else:
    single_image = test_patient_image

# Jika gambar berwarna, konversi ke grayscale
if single_image.ndim == 3:
    single_image = np.mean(single_image, axis=2)

# Jalankan segmentasi
for itrs in range(1, 9):
    test_segmented, test_cardiacfilter, test_outline, test_watershed, test_sobel_gradient = segment_cardiac(single_image, itrs)
```

```

itr_dict = {'Iterations': iteration_titles, 'Computation Times (in seconds)': computation_times}

colors = ['#30336b',] * 8
colors[0] = '#eb4d4b'

import plotly.graph_objects as go

# Plotting the iteration vs computation time chart
fig = go.Figure(data=[go.Bar(
    x=itr_dict['Iterations'],
    y=itr_dict['Computation Times (in seconds)'],
    marker_color=colors
)])
fig.update_traces(texttemplate='%{y:.3s}', textposition='outside')

fig.update_layout(
    title='Iterations vs Computation Times (Cardiac)',
    yaxis=dict(
        title='Computation Times (in seconds)',
        titlefont_size=16,
        tickfont_size=14,
    ),
    autosize=False,
    width=800,
    height=800
)

fig.show()

```

```

# Displaying Sobel Gradient and Watershed images for cardiac data
import matplotlib.pyplot as plt

f, (ax1, ax2) = plt.subplots(1, 2, sharey=True, figsize=(12, 12))

ax1.imshow(sobel_normalized, cmap='gray')
ax1.set_title("Sobel Gradient (Cardiac)")
ax1.axis('off')

ax2.imshow(test_watershed, cmap='gray')
ax2.set_title("Watershed (Cardiac)")
ax2.axis('off')

plt.show()

```

## 1. Import Library

1. %matplotlib inline: Memastikan grafik ditampilkan langsung di dalam notebook (seperti Jupyter Notebook).
2. numpy dan pandas digunakan untuk manipulasi data numerik dan tabular.
3. pydicom untuk membaca file DICOM, yaitu format standar gambar medis.
4. os untuk mengelola file dan folder dalam sistem.
5. scipy.ndimage untuk operasi pemrosesan gambar, seperti morfologi dan filter.
6. skimage digunakan untuk operasi segmentasi dan pengukuran pada gambar.
7. matplotlib digunakan untuk menampilkan visualisasi data.
8. time digunakan untuk menghitung waktu eksekusi fungsi.

## 2. Memuat Data Gambar

1. Folder utama tempat menyimpan data pasien ditentukan sebagai 'INPUT\_FOLDER'.
2. Semua folder atau file dalam direktori utama dibaca dan disimpan dalam variabel patients.
3. Daftar pasien diurutkan secara alfabetis menggunakan fungsi sorting.
4. Enam ID pasien pertama dicetak untuk memastikan data pasien tersedia.

## 3. Fungsi untuk Memuat Gambar

1. Fungsi load\_image digunakan untuk membaca gambar dari jalur file yang diberikan.
2. Gambar dimuat dalam format grayscale untuk mempermudah analisis.
3. Fungsi mengembalikan gambar sebagai array numerik.

## 4. Memuat dan Memeriksa Gambar Pasien

1. Salah satu folder pasien dipilih (dalam hal ini, folder pasien kelima dalam daftar).
2. Semua file dalam folder pasien tersebut dibaca.
3. Setiap file diperiksa:
  - Jika file berhasil dibaca, pesan sukses ditampilkan.
  - Jika ada kesalahan saat membaca, pesan error ditampilkan.

## 5. Menampilkan Gambar Input

1. Jika daftar pasien tidak kosong, folder pasien yang dipilih digunakan untuk memuat gambar.
2. Gambar pertama dari folder tersebut diambil untuk dianalisis.
3. Gambar ditampilkan menggunakan skema warna grayscale, dengan judul "Input Image".
4. Gambar juga disimpan dalam file PNG untuk referensi lebih lanjut.

## 6. Membuat Marker untuk Segmentasi

1. Fungsi generate\_markers menghasilkan marker internal, marker eksternal, dan marker watershed.
  - Marker Internal:
    - Dibuat berdasarkan threshold intensitas piksel.
    - Artefak tepi dihapus menggunakan metode clear\_border.
    - Komponen terhubung diberi label menggunakan fungsi label.
    - Wilayah kecil dihapus berdasarkan ukuran area tengah.
  - Marker Eksternal:
    - Marker internal diperluas dengan dilasi morfologi.
    - Marker eksternal dibuat dengan operasi XOR antara dua hasil dilasi.
  - Marker Watershed:
    - Sebuah array kosong diinisialisasi untuk menyimpan marker watershed.
    - Marker internal dan eksternal digabungkan untuk menghasilkan array marker akhir.

## 7. Visualisasi Marker

1. Gambar marker internal, eksternal, dan watershed ditampilkan menggunakan subplot.
2. Setiap marker diberi judul dan disimpan dalam file PNG untuk referensi.

## 8. Fungsi untuk Segmentasi Area Cardiac

1. Fungsi `segment_cardiac` melakukan segmentasi area cardiac menggunakan beberapa teknik:
  - Gradien Sobel:
    - Gradien intensitas gambar dihitung dalam arah horizontal dan vertikal.
    - Gradien ini digunakan untuk mendeteksi tepi objek dalam gambar.
  - Segmentasi Watershed:
    - Algoritma watershed diterapkan pada gradien Sobel dan marker watershed untuk memisahkan area berdasarkan intensitas.
  - Outline:
    - Wilayah hasil watershed dikurangi menjadi outline menggunakan filter morfologi.
  - Black Top-Hat Morphology:
    - Artefak kecil dalam outline dihapus menggunakan filter top-hat.
  - Cardiac Filter:
    - Kombinasi marker internal dan outline digunakan untuk menghasilkan filter akhir.
    - Filter ini membantu menyoroti area cardiac dalam gambar.
  - Segmentasi:
    - Area cardiac yang terdeteksi diisolasi dari gambar asli menggunakan filter yang dibuat.
2. Waktu komputasi dihitung untuk setiap iterasi dan disimpan dalam daftar.

## 9. Visualisasi Waktu Komputasi

1. Grafik batang dibuat untuk menunjukkan hubungan antara jumlah iterasi segmentasi dan waktu komputasi.
2. Setiap iterasi diberi label, dan warna berbeda digunakan untuk menyoroti iterasi pertama.

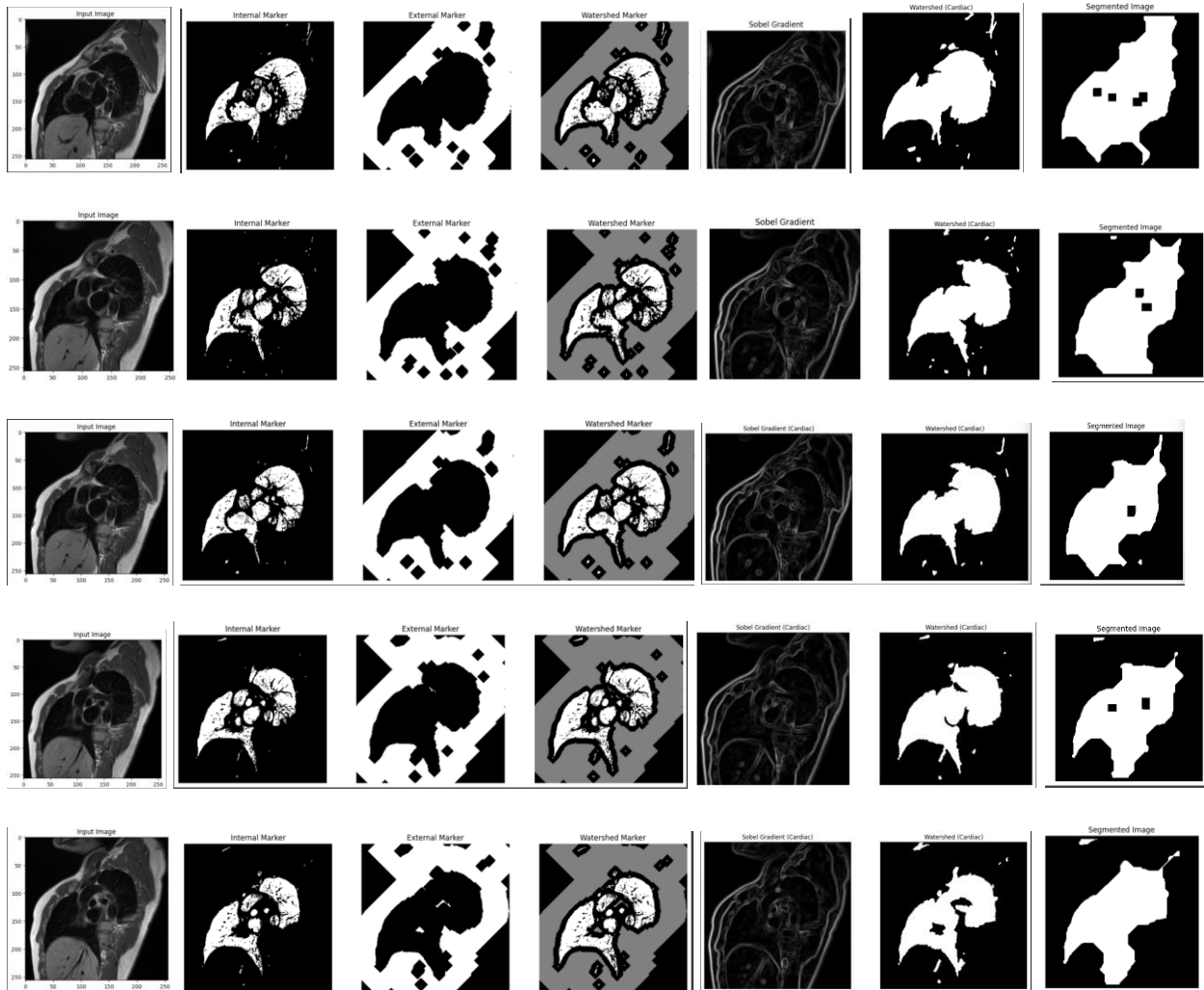
## 10. Visualisasi Sobel Gradient dan Watershed

1. Gradien Sobel dan hasil watershed ditampilkan berdampingan menggunakan subplot.
2. Gambar ini memberikan gambaran visual tentang hasil segmentasi dan deteksi tepi.

## 4.2 Hasil

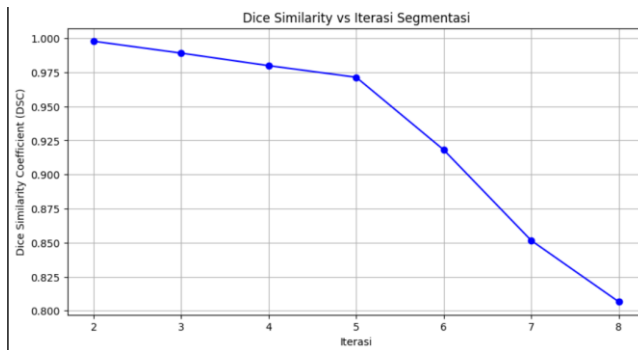
### 4.2.1 Hasil Segmentasi

Digunakan 5 dataset citra jantung, sehingga menghasilkan:



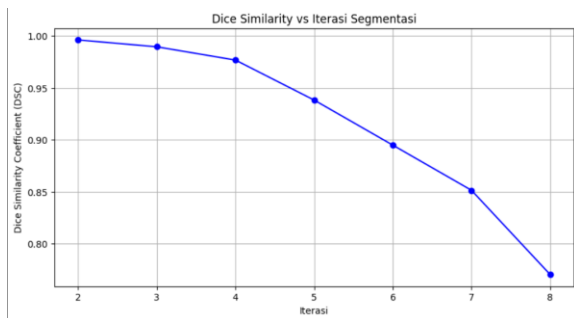
### 4.2.2 Akurasi



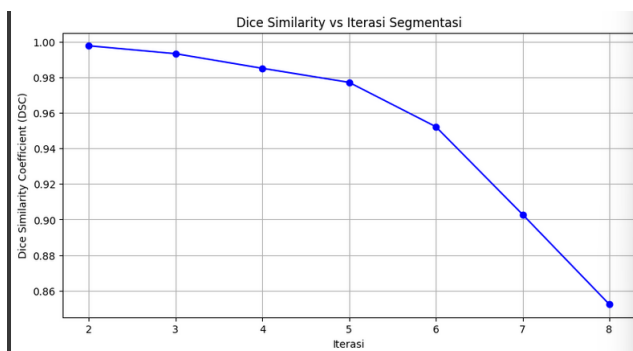


Region 1:

Area	24130.0
Perimeter	1046.9919907732865
Eccentricit	0.8562432070810579
Solidity	0.773794253463314

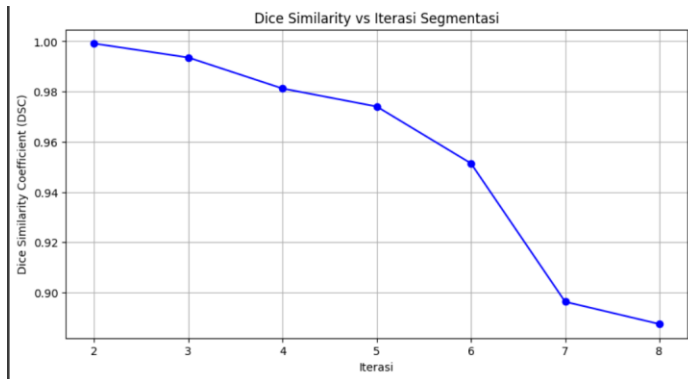


	Region 1	Region 2
Area	62.0	25842.0
Perimeter	28.242640687119284	930.6244584051392
Eccentricit	0.8379923813145083	0.8546361386054461
Solidity	1.0	0.8209803983861232

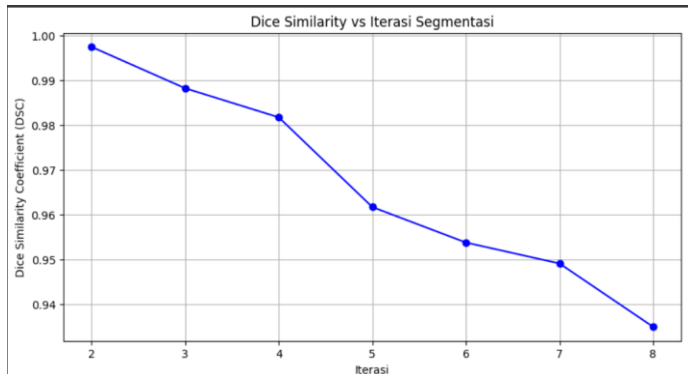


	Region 1	Region 2
Area	22083.0	51.0

Perimeter	865.9797974644666	24.970562748477143
Eccentricit	0.8595530695532972	0.608031799595528
Solidity	0.7810355803918795	1.0



	Region 1	Region 2
Area	105.0	20119.0
Perimeter	45.65685424949238	907.080302527855
Eccentricit	0.9507158786192875	0.777437924893341
Solidity	0.9375	0.7485860991218931



	Region 1	Region 2
Area	183.0	18571.0
Perimeter	64.59188309203678	807.9087296526012
Eccentricit	0.9433530731692129	0.7905594683530199
Solidity	0.8551401869158879	0.701666225866173

### 4.3 Pembahasan

Pada proses segmentasi jantung menggunakan algoritma watershed yang terdiri dari lima baris, masing-masing merepresentasikan tahapan segmentasi pada gambar input berbeda. Setiap baris

memulai proses dari gambar asli (input image) hingga menghasilkan gambar segmentasi akhir. Algoritma watershed digunakan untuk memisahkan wilayah target, yaitu jantung, dari latar belakang atau struktur lain di sekitar jantung dalam citra medis.

Pada tahap awal, gambar input yang merupakan citra medis mentah dari pemindaian ditampilkan. Gambar ini menunjukkan struktur internal tubuh termasuk jantung, jaringan sekitarnya, dan latar belakang. Tahap berikutnya adalah pembuatan penanda internal, yang berfungsi menandai bagian inti dari objek target. Penanda ini dihasilkan dengan teknik seperti thresholding untuk mengidentifikasi area jantung. Di sisi lain, penanda eksternal digunakan untuk membatasi wilayah non-target, seperti latar belakang atau jaringan di luar jantung, sehingga membantu algoritma mengenali batas antara objek target dan bagian lainnya.

Peta watershed kemudian dibentuk dengan menggabungkan penanda internal dan eksternal, serta gradien Sobel. Peta ini memvisualisasikan area transisi atau batas antara jantung dan jaringan lain sebagai wilayah abu-abu. Gradien Sobel sendiri diterapkan untuk mendeteksi tepi atau batas pada gambar melalui perubahan intensitas, memberikan informasi tambahan kepada algoritma watershed untuk menentukan batas objek dengan lebih presisi.

Setelah itu, algoritma watershed diterapkan untuk membagi citra menjadi wilayah yang terpisah berdasarkan gradien intensitas dan penanda awal. Hasil dari tahap ini adalah ekstraksi wilayah target, di mana jantung mulai terlihat lebih jelas terisolasi dari latar belakang. Pada gambar segmentasi akhir, objek target, yaitu jantung, ditampilkan sebagai area putih yang telah berhasil dipisahkan dari elemen lain.

Pada proses segmentasi, terdapat dua jenis wilayah yang dihasilkan, yaitu Region 1 dengan area yang lebih besar dan Region 2 dengan area yang lebih kecil. Setiap wilayah memiliki karakteristik unik yang mencerminkan hasil dari algoritma segmentasi.

Region 1 mencerminkan wilayah utama yang berhasil diidentifikasi oleh algoritma, dengan ukuran area yang besar menunjukkan keberhasilan dalam menangkap objek target secara keseluruhan. Area yang luas ini biasanya mencakup bagian utama dari jantung, sesuai dengan tujuan segmentasi. Namun, area yang terlalu besar dapat menjadi indikasi adanya over-segmentasi, di mana batas objek meluas ke area non-target.

Panjang perimeter wilayah ini juga relatif besar, mencerminkan kompleksitas bentuk dari objek utama. Nilai eccentricity, yang mendekati 1, menunjukkan bahwa objek ini cenderung berbentuk memanjang atau elips. Selain itu, nilai solidity yang tinggi, mendekati 1, mengindikasikan bahwa wilayah tersebut padat dan tidak berongga, yang sesuai dengan harapan untuk objek utama seperti jantung.

Sementara itu, Region 2 merepresentasikan wilayah kecil yang umumnya muncul sebagai outlier atau artefak dari segmentasi. Wilayah ini memiliki area dan perimeter yang kecil, sering kali berupa bagian-bagian yang kurang relevan atau mungkin hasil dari noise pada citra awal.

Wilayah kecil ini biasanya memiliki nilai solidity yang tinggi, mendekati 1, yang menunjukkan bahwa bentuknya cenderung sederhana dan tidak berongga. Eccentricity dari wilayah kecil ini dapat bervariasi tergantung pada bentuknya, tetapi karena ukurannya yang kecil, sering kali bentuknya lebih stabil dan tidak kompleks.

Berdasarkan data, terdapat analisis segmentasi pada beberapa dataset dengan karakteristik yang berbeda untuk setiap iterasi. Segmentasi ini menghasilkan dua jenis wilayah utama, yaitu Region 1 (wilayah besar) dan Region 2 (wilayah kecil), yang dianalisis berdasarkan metrik seperti area, perimeter, eccentricity, dan solidity.

Pada iterasi awal, wilayah besar (Region 1) memiliki area yang cukup luas dan nilai solidity yang tinggi, menunjukkan bahwa segmentasi menghasilkan objek yang cukup solid dan representatif. Namun, seiring bertambahnya iterasi, area ini cenderung sedikit mengecil, sementara nilai solidity juga menurun. Hal ini dapat diartikan bahwa proses segmentasi mulai mengalami degradasi akurasi, terutama karena munculnya over-segmentasi atau ketidakmampuan algoritma mempertahankan bentuk wilayah target.

Sebaliknya, wilayah kecil (Region 2) menunjukkan karakteristik yang lebih sederhana, dengan nilai solidity mendekati 1 pada sebagian besar iterasi. Wilayah ini cenderung stabil tetapi sering kali merupakan outlier atau hasil dari noise pada proses segmentasi.

Pada iterasi terbaik, yaitu dataset ke-2, Region 1 memiliki area yang besar dengan nilai solidity cukup tinggi, menunjukkan segmentasi yang optimal. Namun, pada iterasi berikutnya, wilayah target mulai kehilangan bentuk aslinya, dan area yang disegmentasi menjadi kurang solid.

Secara keseluruhan, proses segmentasi menunjukkan bahwa algoritma dapat bekerja dengan baik di awal, tetapi hasil segmentasi dapat menurun jika iterasi dilakukan terlalu banyak. Iterasi yang optimal adalah yang mampu mempertahankan keseimbangan antara luas area dan soliditas bentuk wilayah target.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Analisis dan identifikasi struktur jantung menggunakan algoritma marker-controlled watershed berhasil memberikan gambaran yang lebih jelas mengenai berbagai bagian anatomi jantung, seperti ventrikel, atrium, dan pembuluh darah. Hal ini memungkinkan pemahaman yang lebih mendalam tentang bentuk dan ukuran struktur jantung, yang sangat penting untuk diagnosis medis yang lebih lanjut.

Selain itu, metode segmentasi marker-controlled watershed terbukti efektif dalam menghasilkan segmentasi yang lebih akurat dan tepat pada citra jantung. Dengan memanfaatkan marker untuk memandu proses segmentasi, metode ini mampu mengurangi kesalahan segmentasi, seperti oversegmentasi atau undersegmentasi, yang sering terjadi pada algoritma konvensional, dan menghasilkan hasil yang lebih handal dalam aplikasi medis.

## REFERENSI

Gharipour, R. (2013). *Segmentation of the Left Atrium Using the Marker-Controlled Watershed Algorithm*. Master of Science Thesis, Chalmers University of Technology. Available from Chalmers University, Gothenburg, Sweden

Cristoforetti, A., Faes, L., Ravelli, F., Centonze, M., Del Greco, M., Antolini, R., & Nollo, G. (2008). Isolation of the left atrial surface from cardiac multi-detector CT images based on marker-controlled watershed segmentation. *Medical Engineering & Physics*, 30(1), 48–58. doi:10.1016/j.medengphy.2007.01.003

Wantanajittikul, K., Saekho, S., Phrommintikul, A., Theera-Umpon, N., & Auephanwiriyakul, S. (2015). Fully Automatic Cardiac T2\* Relaxation Time Estimation Using Marker-controlled Watershed. In *2015 IEEE International Conference on Control System, Computing and Engineering* (pp. 377–380). IEEE. doi:10.1109/ICCSCE.2015.7482210

Parvati, K., Rao, B. S. P., & Das, M. M. (2009). Image Segmentation Using Gray-Scale Morphology and Marker-Controlled Watershed Transformation. *Discrete Dynamics in Nature and Society*, 2009, Article ID 384346, 8 pages. doi:10.1155/2008/384346