

***CONTENT BASED IMAGE RETRIEVAL CITRA SIDIK JARI
MENGUNAKAN METODE *WAVELET* DAUBECHIES
DAN DIAGRAM *NOHIS-Tree****

SKRIPSI

Oleh :

AGUNG SATRIO BUWONO

NIM : 09650145



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM
MALANG
2014**

**CONTENT BASED IMAGE RETRIEVAL CITRA SIDIK JARI
MENGUNAKAN METODE *WAVELET* DAUBECHIES
DAN DIAGRAM *NOHIS-Tree***

SKRIPSI

**Diajukan kepada :
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam Memperoleh Gelar Sarjana
Komputer (S. Kom)**

**Oleh :
AGUNG SATRIO BUWONO
NIM : 09650145**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM
MALANG
2014**

HALAMAN PERSETUJUAN***CONTENT BASED IMAGE RETRIEVAL CITRA SIDIK JARI
MENGUNAKAN METODE WAVELET DAUBECHIES
DAN DIAGRAM NOHIS-Tree*****SKRIPSI****Oleh :**

Nama : Agung Satrio Buwono
NIM : 09650145
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Telah Disetujui, 07 Februari 2014

Dosen Pembimbing I**Dosen Pembimbing II**

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

Dr. M. Amin Hariyadi, MT
NIP. 19670118 200501 1 001

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

CONTENT BASED IMAGE RETRIEVAL CITRA SIDIK JARI MENGUNAKAN METODE WAVELET DAUBECHIES DAN DIAGRAM NOHIS-Tree

SKRIPSI

Oleh :

Agung Satrio Buwono

NIM. 09650145

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)

Tanggal, 28 Februari 2014

Susunan Dewan Penguji :

Tanda Tangan

- | | | | | |
|--------------------|---|--|---|---|
| 1. Penguji Utama | : | <u>Dr. M. Faisal, M.T</u>
NIP. 19740510 200501 1 007 | (|) |
| 2. Ketua Penguji | : | <u>Irwan Budi Santoso, M.Kom</u>
NIP. 19770103 201101 1 004 | (|) |
| 3. Sekretaris | : | <u>Dr. Cahyo Crysdian</u>
NIP. 19740424 200901 1 008 | (|) |
| 4. Anggota Penguji | : | <u>Dr. M. Amin Hariyadi, MT</u>
NIP. 19670118 200501 1 001 | (|) |

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

**HALAMAN PERNYATAAN
ORISINALITAS PENELITIAN**

Saya yang bertandatangan di bawah ini :

Nama : Agung Satrio Buwono
NIM : 09650145
Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika
Judul Penelitian : *Content Based Image Retrieval* Citra Sidik Jari
Menggunakan Metode *Wavelet Daubechies* dan
Diagram *NOHIS-Tree*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 Januari 2014
Yang Membuat Pernyataan,

Agung Satrio Buwono
NIP. 09650145

HALAMAN MOTTO

وَأَسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ ^ج وَإِنَّهَا لَكَبِيرَةٌ إِلَّا عَلَى

الْخَاشِعِينَ

“Jadikanlah sabar dan shalat sebagai penolongmu. dan Sesungguhnya yang demikian itu sungguh berat, kecuali bagi orang-orang yang khusyu'.” (QS : Al-Baqarah-45)

HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

*Segala Puji bagi Allah SWT tuhan semesta alam,
Semoga Shalawat dan Salam senantiasa tercurahkan kepada
Baginda Rosullulloh Saw dan Para Generasi Penerusnya,
kupersembahkan sebuah karya sederhana untuk orang-orang
yang paling kusayangi dan aku banggakan
Ayah dan Mamah
Subandi Arifin, B.Sc dan Desiwyaty Choerunnisa
Para Guru-Guru Spiritualku
Ust.Shochib Abdullah, Ust.Khuzaini, Ust. Imam Muslimin
Atas segala bimbingannya selama ku menuntut ilmu Agama
Kakandaku yang terhormat
Eka Havid Nugraha
yang selalu membuatku bersemangat berjuang
Serta Seluruh keluarga besarku
Kakek Misri, Nenek, Paman Pipin, Tante, Kakak & Adik
Semoga Allah SWT melindungi dan menjaga mereka semua.*

KATA PENGANTAR



Segala puji bagi Allah SWT berkat rahmat, taufiq dan hidayah-Nya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “*Content Based Image Retrieval* Citra Sidik Jari Menggunakan Metode *Wavelet Daubechies* dan Diagram *NOHIS-Tree*” dengan baik. Shalawat dan salam semoga tercurah kepada Nabi Muhammad SAW, para sahabat dan para pengikut setia Beliau.

Content Based Image Retrieval adalah salah satu metode yang digunakan untuk mencari gambar dalam database besar. Dalam penelitian ini disajikan sebuah *CBIR* Citra Sidik Jari dengan menggunakan *wavelet daubechies* sebagai metode untuk ekstraksi fitur citra sidik jari. Hasil dari ekstraksi fitur yang berupa nilai energi aproksimasi dan nilai koefisien detail tersebut kemudian diklasifikasikan menggunakan teknik pengindeksan berdasarkan Diagram *NOHIS-Tree*.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati penulis mengucapkan terima kasih terutama kepada Bapak Dr. Cahyo Crysdiyan, selaku dosen pembimbing I dan Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang; Dr. M. Amin Hariyadi, MT, selaku dosen pembimbing II dan dosen wali; Irwan Budi Santoso, M. Kom selaku dosen penanggung jawab lab. *computer vision*; Seluruh dosen Jurusan Teknik

Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang; serta teman-teman seperjuangan Habib Abdullah, S.Kom; Mar'atul Azizah, S.T; Ahmad Husein; Yousron Amin; A.Farid Rahman; Aunul Mubarak; Miftahul Huda, S.Kom dan teman-teman Teknik Informatika angkatan 2009 yang tidak dapat penulis sebutkan satu-persatu. Penulis berterima kasih atas bantuan, masukan, dukungan serta motivasi kepada penulis.

Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna. Tentunya *tak ada gading yang tak retak*. Akhir kata, semoga penelitian ini dapat bermanfaat dan merupakan alternatif referensi yang berguna bagi peminat dan pengembang bidang *Computer Vision* khususnya *Content Based Image Retrieval*.

Malang, 5 Januari 2014

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	6
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	7
1.6 Sistematika Penelitian	7
BAB II	9
TINJAUAN PUSTAKA	9
BAB III	20
PERANCANGAN DAN IMPLEMENTASI APLIKASI	20
3.1 Desain Proses	20
3.2 Desain GUI (<i>Graphical User Interface</i>)	56
3.3 Implementasi Aplikasi	65
BAB IV	68
EKSPERIMEN DAN PEMBAHASAN	68
4.1 Langkah-Langkah Eksperimen	68
4.2 Eksperimen atau Uji Coba	69
4.3 Pembahasan	82

4.4 Integrasi <i>Content Based Image Retrieval</i> Citra Sidik Jari menggunakan metode <i>Wavelet Daubechies</i> dan diagram <i>NOHIS-Tree</i> dengan Al-Qur'an	97
BAB V	101
KESIMPULAN DAN SARAN	101
5.1 Kesimpulan	101
5.2 Saran	102
DAFTAR PUSTAKA	104
LAMPIRAN 1 Database Citra Sidik Jari	106
LAMPIRAN 2 Listing <i>Source Code</i> Proses <i>Similarity Search</i> menggunakan <i>NOHIS-Search</i>	117
LAMPIRAN 3 Listing <i>Source Code</i> <i>Visualisasi Ranking</i> Identifikasi	123

DAFTAR GAMBAR

<i>Gambar 2.1 Gambar (a) Pola Utama Sidik jari (a) loop, (b) whorl, (c) arches .</i>	12
<i>Gambar 3.1 Blok Diagram Proses Aplikasi CBIR Citra Sidik Jari dengan menggunakan Metode Wavelet Daubechies dan Diagram NOHIS-Tree.....</i>	21
<i>Gambar 3.2 Hasil scanning citra sidik jari.....</i>	22
<i>Gambar 3.3 Hasil cropping citra sidik jari berukuran 512x512 piksel.....</i>	23
<i>Gambar 3.4 Diagram Alur Akuisisi Citra Sidik Jari dan Preprocessing.....</i>	24
<i>Gambar 3.5 Pemilihan threshold secara analisis visual histogram.....</i>	26
<i>Gambar 3.6 Source Code dari Pencarian Nilai T sebagai Batas Ambang Threshold.....</i>	27
<i>Gambar 3.7 Hasil Threshold Citra Sidik Jari</i>	28
<i>Gambar 3.8 Diagram alur ROI citra blackwhite sidik jari.....</i>	29
<i>Gambar 3.9 listing source code untuk membaca citra sidik jari blackwhite</i>	30
<i>Gambar 3.10 Hasil ROI citra blackwhite dengan nilai piksel B/W = 0 terluar....</i>	30
<i>Gambar 3.11 Hasil preprocessing yang meliputi proses thresholding, ROI dan resize citra sidik jari</i>	31
<i>Gambar 3.12 Source Code dari Proses Akuisisi Citra Sidik Jari</i>	32
<i>Gambar 3.13 Diagram Alur Ekstraksi Fitur menggunakan Wavelet Daubechies</i>	37
<i>Gambar 3.14 Dekomposisi menggunakan Wavelet Daubechies level 1</i>	38
<i>Gambar 3.15 Hasil Perhitungan Energi</i>	40
<i>Gambar 3.16 Source code untuk transformasi Wavelet Daubechies berserta perhitungan energi wavelet daubechies.....</i>	42
<i>Gambar 3.17 Proses ekstaksi feature menggunakan wavelet daubechies pada akuisisi citra sidik jari input</i>	42
<i>Gambar 3.18 Ekstraksi feature menggunakan wavelet daubechies pada proses akuisisi citra sidik jari query.....</i>	43
<i>Gambar 3.19 Diagram alur penyimpanan hasil ekstraksi fitur ke database</i>	44
<i>Gambar 3.20 Pengisian identitas dan ekstaksi feature citra input.....</i>	44
<i>Gambar 3.21 Source code untuk button proses simpan menggunakan syntax SQL</i>	45
<i>Gambar 3.22 Diagram Alur Teknik Pengindeksan dengan menggunakan Diagram NOHIS-Tree</i>	46
<i>Gambar 3.23 Arsitektur dari Nohis Search (Taileb, 2013).....</i>	48
<i>Gambar 3.24 Teknik Pengindeksan pada 2 Dimensi (a) Penentuan Titik Pusat (w), (b) Hyper-plane Orthogonal, (c) Minimum Bounding Rectangle, (d) Non Overlapping Index (Taileb, M. Lamrous, S., and Touati, S. 2008)</i>	49
<i>Gambar 3.25 Teknik Klasifikasi Ekstraksi Fitur pada NOHIS-Tree (a) Retangle with the childrends, (b) NOHIS-tree (Taileb, M. Lamrous, S., and Touati, S. 2008)</i>	50
<i>Gambar 3.26 Diagram Alur Proses Similarity Search dengan NOHIS Search....</i>	55
<i>Gambar 3.27 Contoh Proses Similarity Search pada diagram NOHIS-tree level 4</i>	56
<i>Gambar 3.28 Antarmuka input citra sidik jari</i>	57

Gambar 3.29 Tampilan proses input citra sidik jari	58
Gambar 3.30 Proses buka file dan thresholding pada proses input.....	59
Gambar 3.31 Pengisian identitas dan ekstraksi feature citra input.....	60
Gambar 3.32 Antarmuka proses identifikasi content based image retrieval citra sidik jari	61
Gambar 3.33 Proses buka file dan threshold pada proses identifikasi	62
Gambar 3.34 Proses ekstraksi feature citra sidik jari	63
Gambar 3.35 Contoh Proses Similarity Search pada diagram NOHIS-tree level 4	64
Gambar 3.36 Tampilan Ranking hasil identifikasi Content Based Image Retrieval citra sidik jari	65
 Gambar 4.1 Persebaran Data 2 Nilai Koefisien Aproksimasi dan Koefisien Detail	70
Gambar 4.2 Titik Pusat menggunakan Persamaan Regresi Linear	71
Gambar 4.3 Separating Hyper-plane Orthogonal.....	72
Gambar 4.4 Pembagian Kuadran (Minimum Bounding Rectangle)	73
Gambar 4.5 Non-Overlapping Index.....	74
Gambar 4.6 Kode sumber untuk menampilkan teknik pengindeksan berdasarkan diagram NOHIS-Tree	76
Gambar 4.7 Hasil proses identifikasi CBIR citra sidik jari menggunakan wavelet daubechies dan diagram NOHIS-Tree	77
Gambar 4.8 Hasil proses identifikasi pada diagram NOHIS-tree citra sidik jari	77
Gambar 4.9 Hasil proses identifikasi menggunakan daubechies db4 matlab dan metode brute force	78
Gambar 4.10 Studi Perbandingan Tingkat Akurasi Wavelet Daubechies dengan Fungsi db4 Matlab.....	95
Gambar 4.11 Grafik Studi Perbandingan Efisiensi waktu pencarian antara NOHIS-Tree dengan Brute Force	97

DAFTAR TABEL

Tabel 3.1 <i>Orthogonal Daubechies Coefficients D2-D12 (Low-Pass Filter)</i>	33
Tabel 3.2 <i>Orthonormal Daubechies Coefficients db4</i>	34
Tabel 3.3 Struktur tabel Data	45
Tabel 4.1 Hasil uji citra sidik jari	79
Tabel 4.2 Kesesuaian Data Uji 9 Hasil Teratas.....	82
Tabel 4.3 Kesesuaian Data Uji 6 Hasil Teratas.....	84
Tabel 4.4 Kesesuaian Data Uji 3 Hasil Teratas.....	85
Tabel 4.5 Kesesuaian Data Uji 1 Hasil Teratas.....	87
Tabel 4.6 Perbandingan Efisiensi dan Akurasi Proses Identifikasi.	90

ABSTRAK

Buwono, Agung Satrio 2014. ***Content Based Image Retrieval Citra Sidik Jari Menggunakan Metode Wavelet Daubechies Dan Diagram NOHIS-Tree***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing : (I) Dr. Cahyo Crysdian (II) Dr. M. Amin Hariyadi, MT

Kata Kunci : *Content Based Image Retrieval, Wavelet Daubechies, NOHIS-Tree*

Penerapan komputer untuk pengenalan pola telah banyak dilakukan, seperti pengenalan karakter, pengenalan suara, namun untuk pengenalan citra sidik jari dari penelitian-penelitian yang sudah ada masih banyak kelemahannya. Hal tersebut disebabkan oleh karakteristik citra sidik jari yang memiliki kemiripan yang sangat tinggi tetapi apabila menggunakan salah satu metode pengenalan pola, dimana salah satunya yaitu metode *wavelet* dan diaplikasikan dengan komputer, maka citra sidik jari tersebut dapat dikenali.

Kendala lainnya adalah semakin meningkatnya kebutuhan akan kumpulan citra dalam jumlah besar menyebabkan pemrosesan citra digital semakin berat. Sebagai solusi para peneliti mengusulkan temu kenali citra berbasis konten atau *Content Based Image Retrieval (CBIR)* yang merupakan salah satu metode yang digunakan untuk mencari gambar dalam database besar.

Penelitian ini bertujuan mengimplementasikan metode *wavelet daubechies* sebagai *preprocessing* digabungkan dengan penerapan teknik pengindeksan menggunakan diagram *NOHIS-Tree* pada aplikasi *CBIR* citra sidik jari untuk mengidentifikasi citra sidik jari dan mengukur seberapa baik performa akurasi serta mengetahui pengaruh pengelompokan terhadap tingkat efisiensi waktu pencarian. Penggunaan *Wavelet Daubechies* sebagai ekstraksi fitur untuk mendekomposisi citra sidik jari. Hasil dari ekstraksi fitur yang berupa nilai energi aproksimasi dan nilai koefisien detail tersebut kemudian diklasifikasikan menggunakan teknik pengindeksan berdasarkan Diagram *NOHIS-Tree*. Sembilan hasil *similarity search* teratas dianggap sebagai hasil identifikasi citra sidik jari.

Hasil dari penelitian menunjukkan bahwa aplikasi *CBIR* Citra sidik jari dengan menggunakan *Wavelet Daubechies* dan Diagram *NOHIS-Tree* memiliki tingkat akurasi tertinggi sebesar 62,50% dengan waktu pencarian rata-rata 9,6 detik. Sedangkan dengan menggunakan *wavelet db4* matlab dan metode *Brute Force* memiliki akurasi tertinggi sebesar 84,375 % dengan waktu pencarian rata-rata 182,25 detik atau 3,04 menit.

ABSTRACT

Buwono, Agung Satrio 2014. *Content Based Image Retrieval on Fingerprint Image Using Wavelet Daubechies Method and NOHIS-Tree Diagram*. Thesis. Informatics Department of Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University, Malang.
 Adviser : (I) Dr. Cahyo Crysdiar (II) Dr. M. Amin Hariyadi, MT

Keywords : Content Based Image Retrieval , Daubechies Wavelet , NOHIS - Tree

The application of computers for pattern recognition has been widely applied, such as character recognition, speech recognition, but for the introduction of the fingerprint image from the studies already exist still many weaknesses. This is due to the characteristics of fingerprint image which has a very high similarity, but when using one of the methods of pattern recognition, where one of them is wavelet method and applied to the computer, then the fingerprint image can be recognized.

Another constraint is the increasing need for large amounts of image collection of digital image processing causing increasingly severe. As a solution the researchers propose a content-based image retrieval recognize or Content Based Image Retrieval (CBIR) which is one of the methods used to search for images in large databases.

This study aims to implement the Daubechies wavelet method as preprocessing coupled with the application of indexing techniques using diagrams NOHIS - Tree in CBIR applications fingerprint image to identify the fingerprint image and measure how well the performance accuracy and determine the effect of clustering on the efficiency of search time. The use of Daubechies wavelet decomposition as feature extraction for fingerprint image. The results of the extraction of features in the form of the energy value of the approximation and detail coefficients values were then classified using indexing techniques based NOHIS - Tree Diagram. Nine top results of similarity search is regarded as the identification of the fingerprint image.

The results of the study show that the CBIR applications fingerprint image using Daubechies Wavelet and NOHIS Diagram - Tree has the highest accuracy rate of 62.50 % with an average search time of 9.6 seconds. While using wavelet db4 matlab and method Brute Force has the highest accuracy of 84.375 % with an average search time of 182.25 seconds or 3.04 minutes.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi, terutama di bidang dunia digital membawa perubahan cukup besar. Salah satunya dengan adanya digitalisasi data citra. Hampir semua sistem analog digantikan dengan sistem komputerisasi. Perkembangan pengolahan citra digital juga semakin luas, diantaranya adalah pengenalan pola (*pattern recognition*) pada citra digital. Pola dari citra yang diolah adalah guratan-guratan sidik jari. Ciri khas yang dimiliki suatu guratan pola sidik jari tersebut berbeda antara satu orang dengan orang lain. Perbedaan pola dari sidik jari tersebut bisa digunakan sebagai identifikasi (Hendarko, 2010).

Dalam Al-Qur'anul Karim dijelaskan bahwa kelak di hari akhir indera penglihatan, pendengaran, dan kulit akan menjadi saksi atas apa yang dikerjakan selama manusia hidup di dunia. Kulit manusia pada jari-jemari manusia (sidik jari) diciptakan oleh Allah SWT dengan sebaik-baiknya berfungsi sebagai identitas pribadi seseorang yang berbeda dengan manusia lainnya. Allah SWT berfirman dalam surat Al-Qiyamah Ayat 4 :

بَلَىٰ قَدَرِينَ عَلَىٰ أَن نُّسَوِيَ بَنَاتَهُ ۖ

Artinya : “bukan demikian, sebenarnya Kami Kuasa menyusun (kembali) jari jemarinya dengan sempurna.” (QS : 75 - 4).

Penerapan komputer untuk pengenalan pola telah banyak dilakukan, seperti pengenalan karakter, pengenalan suara, namun untuk pengenalan citra sidik jari dari penelitian-penelitian yang sudah ada masih banyak kelemahannya. Hal tersebut disebabkan oleh karakteristik citra sidik jari yang memiliki kemiripan yang sangat tinggi. Pengenalan pola merupakan pengenalan suatu objek dengan menggunakan berbagai metode dimana dalam proses pengenalannya memiliki tingkat akurasi yang tinggi. Memiliki tingkat akurasi yang tinggi maksudnya adalah suatu objek yang secara manual tidak dapat dikenali tapi bila menggunakan salah satu metode pengenalan pola, dimana salah satunya yaitu metode *wavelet* dan diaplikasikan dengan komputer, maka citra tersebut dapat dikenali (Sedyono, Nataliani, Mariana, 2009).

Berbagai penelitian tentang sidik jari telah banyak dikembangkan, berdasarkan penelitian-penelitian yang telah ada menunjukkan bahwa salah satu teknik ekstraksi fitur yang memiliki tingkat pengenalan pola dengan tingkat akurasi baik adalah *wavelet daubechies*. Salah satunya adalah penelitian yang dilakukan oleh Minarni (2004), hasil penelitiannya menunjukkan bahwa pengenalan citra dan ekstraksi ciri *wavelet*

Daubechies meningkat unjuk kerjanya sebesar 1% dibandingkan dengan *wavelet Haar*.

Pada penelitian lain yang dilakukan oleh Fitriana Nelvi (2013), yaitu membuat suatu aplikasi yang mampu mengidentifikasi citra sidik jari. Namun pada penelitian ini, proses identifikasi menggunakan algoritma *Wavelet Haar* masih menghasilkan tingkat akurasi sekitar 50% dan proses identifikasi citra sidik jari menggunakan *Euclidian Distance* dan *Brute Force* ini membutuhkan waktu yang relatif lama dalam proses pencariannya.

Untuk mengatasi masalah tingkat akurasi dan proses identifikasi citra sidik jari yang membutuhkan waktu lebih lama tersebut, salah satu solusinya adalah dengan menggunakan *wavelet daubechies* sebagai pengekstraksi fitur citra sidik jari agar diperoleh hasil pengenalan pola sidik jari dengan tingkat akurasi yang lebih baik. Permasalahan tentang lamanya proses identifikasi citra sidik jari pada penelitian tersebut dikarenakan banyaknya jumlah data yang diolah sehingga membutuhkan teknik klasifikasi untuk diterapkan pada sebuah sistem *image retrieval* yang dikenal dengan *CBIR (Content Based Image Retrieval)* atau Sistem temu kenali citra berbasis konten.

Dengan semakin meningkatnya kebutuhan akan kumpulan citra dalam jumlah besar, kendala yang dihadapi dalam pemrosesan citra digital semakin berat. Sebagai solusi para peneliti mengusulkan temu kenali citra

berbasis konten atau *Content Based Image Retrieval* (Karmilasari dan Sumarna, 2011).

Sistem *Content Based Image Retrieval* sangat penting untuk menggunakan teknik pengindeksan yang efisien untuk melakukan dan mempercepat pencarian dalam database besar. Teknik pengindeksan yang digunakan juga harus mendukung tingginya dimensi gambar fitur. Sebuah indeks berdimensi tinggi dapat dibuat dengan menggunakan teknik tradisional dari pengindeksan seperti R-tree, atau dengan menggunakan sebuah algoritma *clustering* untuk membentuk *cluster* atau kelompok dari *descriptors* (hasil ekstraksi fitur), dan kelompok yang didukung oleh suatu struktur hirarkis sebagai contoh Birch menggunakan CF-tree, DBSCAN menggunakan R*-tree dan X-tree. Banyak struktur indeks berdimensi tinggi telah diusulkan, yang paling dikenal dan sering digunakan adalah *data-partitioning* (partisi data) berbasis struktur indeks seperti SS-tree, SR-tree, X-tree, dianggap sebagai ekstensi dari R-tree, dan *space-partitioning* (partisi ruang) berdasarkan struktur indeks seperti k-d-B-tree, hB-tree, dan LSDh-tree yang berasal dari kd-tree. Struktur R-tree berbasis struktur indeks masih menghasilkan *overlapping* diantara batasan-batasan *bounding*-nya dan fanouts yang rendah, hal ini berpengaruh negatif pada hasil pemrosesan *query*. Kd-tree berdasarkan struktur indeks memiliki kelemahan yang pada dasarnya tidak ada jaminan menggunakan ruang yang dialokasikan, hal ini menyebabkan adanya *cluster* yang sedikit atau kosong. Berdasarkan kekurangan tersebut dan dengan suatu tujuan untuk

mempercepat proses pencarian dalam struktur indeks berdimensi tinggi maka teknik yang disebut *NOHIS-tree* diusulkan sebagai solusi (Taileb and Touati, 2011).

Dalam penelitian ini diusulkan suatu model sistem temu kenali citra (*Content Based Image Retrieval*) citra sidik jari dengan teknik klasifikasi menggunakan struktur indeks diagram pohon *NOHIS-tree* untuk mendapatkan hasil dengan tingkat efektivitas yang lebih baik. Banyak metode telah digunakan dalam penanganan di dalam konteks *CBIR*, khususnya pada pencarian citra-citra di dalam suatu image database, salah satu yang populer adalah menggunakan wavelet sebagai preprocessing-nya untuk mentransformasi nilai-nilai pixel (warna) pada citra. Wavelet yang digunakan dalam penelitian ini adalah *Daubechies Wavelet Transform* yang digunakan untuk mendekomposisi nilai-nilai pixel dari suatu citra. Penggunaan *wavelet daubechies* sebagai ekstraksi fitur dan diagram *NOHIS-tree* sebagai teknik klasifikasi dan identifikasi dari sistem *Content Based Image Retrieval* citra sidik jari ini diharapkan dapat menghasilkan pengenalan pola sidik jari dengan tingkat akurasi yang lebih baik dan proses identifikasi yang lebih optimal dan efisien.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini antara lain :

- a. Apakah metode *Wavelet Daubechies* dan Diagram *NOHIS-Tree* bisa diterapkan pada aplikasi *Content Based Image Retrieval* Citra Sidik Jari?

- b. Seberapa baik performa *Content Based Image Retrieval* Citra Sidik Jari dengan menggunakan metode *Wavelet Daubechies*?
- c. Seberapa baik performa *Content Based Image Retrieval* Citra Sidik Jari dengan menggunakan metode Diagram *NOHIS-Tree*?

1.3 Batasan Masalah

Agar penelitian ini tidak menyimpang dari permasalahan maka perlu adanya batasan masalah, yaitu antara lain :

- a. Citra yang digunakan berupa gambar sidik jari dalam format gambar '.jpg', '.tif', '.bmp', '.png', dan '.gif'. yang telah melalui proses cropping terlebih dahulu.
- b. Penelitian dikhususkan pada penggunaan alihragam *wavelet daubechies* orde 4 sebagai pengekstraksi ciri dan diagram *NOHIS-tree* sebagai sistem identifikasi citra sidik jari.
- c. Perangkat lunak yang digunakan dalam penelitian ini adalah MATLAB R2011b.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini antara lain :

- a. Membuktikan penerapan metode *Wavelet Daubechies* dan Diagram *NOHIS-Tree* pada aplikasi *Content Based Image Retrieval* Citra Sidik Jari.
- b. Mengukur performa *Wavelet Daubechies* pada aplikasi *Content Based Image Retrieval* Citra Sidik Jari.
- c. Mengukur performa Diagram *NOHIS-Tree* pada aplikasi *Content Based Image Retrieval* Citra Sidik Jari.

1.5 Manfaat Penelitian

Hasil dalam penelitian ini diharapkan akan memberikan kontribusi terhadap perkembangan pengolahan citra digital khususnya dalam identifikasi sidik jari dan membantu pihak yang berkepentingan dalam mengidentifikasi sidik jari serta membuat suatu perangkat lunak yang mampu mengenali pola sidik jari yang memiliki tingkat akurasi dan efektivitas lebih baik.

1.6 Sistematika Penelitian

Penulisan skripsi ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pendahuluan, membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penyusunan tugas akhir, metodologi, dan sistematika penyusunan tugas akhir.

BAB II Landasan Teori

Landasan teori berisikan beberapa teori yang mendasari dalam penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah dasar teori yang berkaitan dengan pembahasan tentang sidik jari, transformasi *Wavelet Daubechies* dan Diagram *NOHIS-Tree*.

BAB III Analisa dan Perancangan

Menganalisa kebutuhan sistem untuk membuat aplikasi meliputi spesifikasi kebutuhan software dan langkah-langkah pembuatan Aplikasi *Content Based Image Retrieval* Citra Sidik Jari Menggunakan Metode *Wavelet Daubechies* dan Diagram *NOHIS-Tree*.

BAB IV Eksperimen dan Pembahasan

Menjelaskan tentang pengujian Aplikasi *Content Based Image Retrieval* Citra Sidik Jari Menggunakan Metode *Wavelet Daubechies* dan Diagram *NOHIS-Tree* yang telah diterapkan.

BAB V Penutup

Berisi kesimpulan dan saran



BAB II

TINJAUAN PUSTAKA

I. *Content based image retrieval (CBIR)* atau Temu Kenali Citra Berbasis Konten.

Content Based Image Retrieval (CBIR) atau temu kenali citra adalah metode yang digunakan untuk melakukan pencarian citra digital pada suatu database citra. Yang dimaksud dengan "Content-Based" di sini adalah : objek yang dianalisa dalam proses pencarian itu adalah actual contents (kandungan aktual) sebuah citra. Istilah konten pada konteks ini merujuk pada warna, bentuk, tekstur, atau informasi lain yang didapatkan dari citra tersebut (Anonym, 2010).

Proses umum dari CBIR adalah pada citra yang menjadi *query* dilakukan proses ekstraksi fitur (*image contents*), begitu halnya dengan citra yang ada pada basis data citra juga dilakukan proses seperti pada citra *query*. Parameter fitur citra yang dapat digunakan untuk *retrieval* pada *system* ini dapat berupa histogram, susunan warna, tekstur, dan *shape*, tipe spesifik dari objek, tipe event tertentu, nama individu, lokasi, emosi (Sumarna, 2010).

Awalnya teknik yang dipakai bukan mencari fitur melainkan berdasarkan penambahan deskripsi mengenai citra dalam bentuk teks. Dengan kata lain pertama citra diberi teks berdasarkan citra tersebut kemudian dilakukan pencarian berdasarkan teks (*text based*) menggunakan sistem database manajemen tradisional. Namun karena pembangkitan teks secara otomatis, mengenai deskripsi

spektrum citra, secara detail sulit untuk dilakukan kebanyakan aplikasi *text based image retrieval* saat itu melakukan pemberian teks deskripsi citra secara manual.

Penelitian lain dan pengembangan *image retrieval* dimulai pada sekitar 1970-an. Pada tahun 1979, sebuah konferensi mengenai *Database Techniques for Pictorial Application* diadakan di Florida. Sejak saat itu aplikasi dalam melakukan manajemen database citra menarik perhatian peneliti.

Beberapa piranti lunak dibuat untuk mencoba menghasilkan RCBK (Retrieval Citra Berbasis Konten) yang efisien, misalnya *QBIC (IBM)*, *PicHunter (NEC Research Institute)*, *VisualSEEK (Columbia University Centre for Telecommunication Research)*, *Photobox (MIT's Vision and Modeling Group)*, *Chabot*, *Excalibur*, dan lain-lain. Dari piranti-piranti tersebut RCBK dengan pendekatan ekstraksi fitur otomatis semakin berkembang, studi tentang berbagai fitur dari suatu citra dipelajari lebih lanjut.

Fitur yang digunakan dalam RCBK dapat dikategorikan menjadi dua, yaitu fitur umum dan khusus. Fitur umum biasanya meliputi warna, tekstur serta bentuk dan fitur yang khusus digunakan pada area aplikasi yang khusus, misalnya pengenalan wajah atau sidik jari (Jeffrey, 2010).

Wahyu Hidayat (2007) menerangkan dalam penelitiannya tentang penerapan *K-Nearest Neighbour* untuk klasifikasi gambar *landscape* berdasarkan fitur warna dan tekstur, bahwa *CBIR* bekerja dengan cara mengukur kemiripan gambar *query* dengan semua gambar yang ada dalam database sehingga *query cost* berbanding lurus dengan jumlah gambar dalam database. Membatasi *range* pencarian gambar dengan cara melakukan klasifikasi merupakan salah satu cara

untuk mengurangi *query cost* pada *CBIR*. Dalam penelitian ini metode *Colour Histogram* dan *Edge Histogram Descriptor* digunakan sebagai pengekstraksi fitur kemudian hasilnya diklasifikasikan menggunakan *K-Nearest Neighbor*. Hasil pengujiannya menunjukkan bahwa kombinasi fitur warna dan tekstur memberikan tingkat akurasi yang lebih tinggi dibandingkan dengan klasifikasi berdasarkan fitur warna saja atau tekstur saja namun membutuhkan waktu klasifikasi yang lebih lama.

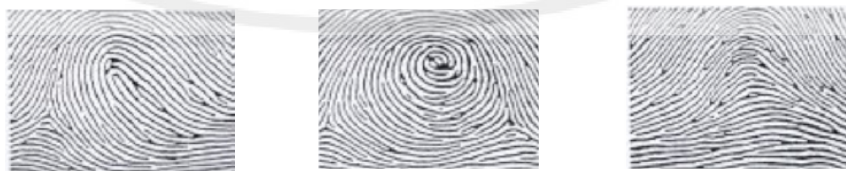
Agus Sumarna (2010) dalam penelitian ini dilakukan pencarian citra (*CBIR*) berdasarkan ekstraksi fitur warna dengan menggunakan 2 tipe histogram yaitu *Global Color Histogram*(GCHs) dan *Local Color Histogram*(LCHs). Proses *CBIR* dengan metode *Local Color Histogram* menghasilkan citra hasil pencarian yang jumlahnya banyak secara kuantitas, dan hasil pencariannya terpengaruh oleh citra geometri misalnya citra yang dirotasi, sehingga tidak efektif secara kualitas. Sedangkan metode *Global Color Histogram* menghasilkan citra hasil pencarian yang lebih sedikit secara kuantitatif dibandingkan dengan hasil pencarian *Local Color Histogram*, dan hasil pencariannya tidak dipengaruhi oleh citra geometri. Sehingga lebih efektif karena hanya menampilkan citra yang mempunyai penyebaran warna yang mirip dengan citra *query*-nya. Aplikasi *CBIR* dalam penelitian ini memiliki kekurangan yaitu jika *query* citra yang ada didatabase citra ukurannya atau dimensinya terlalu besar maka *CBIR* akan berhenti melakukan *query* atau terjadi *error*, sehingga harus diadakan *resize* atau perubahan ukuran citra database agar proses *query* bisa berjalan cepat.

II. Identifikasi Sidik Jari.

Sistem keamanan menggunakan sidik jari ini disebut sebagai sistem identifikasi biometrik. Sidik jari telah terbukti cukup akurat, aman, mudah dan nyaman untuk dipakai sebagai identifikasi bila dibandingkan dengan sistem biometrik lainnya seperti retina mata atau DNA. Hal ini dapat dilihat pada sifat yang dimiliki oleh sidik jari, antara lain :

1. Perennial nature, yaitu guratan-guratan pada sidik jari yang melekat pada kulit manusia seumur hidup.
2. Immutability, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapat kecelakaan yang serius.
3. Individuality, pola sidik jari adalah unik dan berbeda untuk setiap orang (Elvayandri, 2002).

Secara umum, sidik jari dibagi menjadi 3 tipe pola utama, yaitu arches, loops, dan whorls. Tipe loops merupakan pola yang paling banyak ditemukan pada manusia, Menurut Francis Galton, sekitar 60% sidik jari bertipe loops, 30% bertipe whorls, dan 10% bertipe arches (Falasev *et al*, 2009).



Gambar 2.1 Gambar (a) Pola Utama Sidik jari (a) loop, (b) whorl, (c) arches

Eko Sedyono, Yessica, Chrisanty (2009) menjelaskan dalam penelitiannya tentang Klasifikasi Sidik Jari dengan menggunakan metode *wavelet symlet* bahwa *wavelet symlet 6* mampu mengolah citra dan menampilkan ciri-ciri khusus citra, namun cara pengambilan sidik jari dengan tinta stempel yang dicap pada kertas kemudian di *scan* merupakan cara yang kurang baik. Walaupun dengan menerapkan pendekatan binerisasi dengan pola *input 9 bit* terhadap citra sidik jari secara manual, penerapan mode *wavelet symlet 6* dapat memberikan hasil ketepatan mencapai 80%.

Hendarko (2010) dalam penelitiannya tentang identifikasi sidik jari membandingkan prosentase tingkat keberhasilan pada masing-masing tapis yang menggunakan *wavelet Haar*, *Daubechies*, *Symlet* maupun *Coiflet*. Hasilnya menunjukkan bahwa jenis *wavelet* yang memiliki prosentase tingkat keberhasilan paling tinggi adalah jenis *wavelet daubechies* tipe *daubechies 8 (db8)* yang mampu mengidentifikasi citra uji sidik jari sampai 91%. Namun penelitian ini masih bisa disempurnakan lagi dengan beberapa cara diantaranya, perlu dikembangkan lagi alat pemindai sidik jari yang langsung terhubung dengan perangkat lunak, sehingga sistem dapat memproses secara waktu nyata (*real time*). Metode penelitian terhadap jenis *wavelet* selain keempat jenis yang telah diujikan untuk memperoleh jenis *wavelet* yang paling optimal dan metode pengenalan jarak *euclidian distance* bisa diganti misalnya dengan metode Jaringan Syaraf Tiruan (JST) kemudian diperbandingkan untuk memperoleh metode pengenalan manakah yang lebih baik.

Fitriana Nelvi (2013) dalam penelitian ini yaitu membuat suatu aplikasi yang mampu mengidentifikasi citra sidik jari rotasi. Proses identifikasinya meliputi empat tahap utama yaitu akuisisi citra, *preprocessing*, *main processing* dan identifikasi citra sidik jari. Akuisisi citra diambil dengan membubuhkan jari yang telah dikenai *stamp pad ink* ke kertas template. Metode *preprocessing* yang telah dikenai *stamp pad ink* ke kertas template. Metode *preprocessing* yang digunakan antara lain *cropping* dan *tresholding*. *Main process* menggunakan *Analytical Geometry* sebagai metode untuk merotasi dan *Wavelet Transform* sebagai metode untuk mengekstraksi *feature* yang terdapat pada citra. Nilai pada *feature* yang di peroleh kemudian dibandingkan dengan nilai *feature* yang terdapat pada database. Delapan hasil identifikasi teratas dianggap sebagai hasil identifikasi citra sidik jari. Terdapat beberapa kelemahan pada aplikasi sidik jari ini yaitu proses identifikasi (*brute force*) membutuhkan waktu yang relatif lama dan metode *Wavelet Haar* dan *Euclidian Distance* yang digunakan pada identifikasi citra sidik jari rotasi ini masih menghasilkan tingkat akurasi di bawah 60%.

III. *Wavelet Daubechies.*

Ide tentang *wavelet* telah ada sejak awal abad 20, tetapi pengembangannya baru dicapai pada tahun 1980-an. Disamping *paper* oleh Fraizer dan Jawert (1985), *wavelet* juga populer disekolah “*France School*” di Perancis yang diketuai oleh J. Morlet, A. Grossman, dan Y. Meyer (Hendarko 2010).

Daubechies adalah salah satu keluarga *Wavelet*, lebih tepatnya keluarga *Wavelet* yang bersifat orthogonal. *Wavelet* ini ditemukan oleh Ingrid Daubechies.

Minarni (2004) dalam penelitian ini membahas tentang sistem pengklasifikasian sidik jari menggunakan transformasi wavelet sehingga menghasilkan multiresolusi dari citra aslinya. Transformasi wavelet ini digunakan sebagai metode ekstraksi ciri sekaligus mereduksi dimensi citra masukan. Citra tereduksi selanjutnya diproses untuk klasifikasinya. Pengenalan dan klasifikasinya dengan menerapkan jaringan syaraf tiruan Learning Vektor Quantizations (LVQ) mengelompokkan sidikjari ke salah satu pola utama sidikjari (*whorl, left, loop, right, loop, arch, dan tented arch*). Hasil penelitian ini menunjukkan pengenalan dan ekstraksi ciri wavelet Daubechies meningkat unjuk kerjanya sebesar 1% dibandingkan dengan wavelet Haar.

Hal ini juga diperkuat dengan penelitian-penelitian lainnya diantaranya, yaitu oleh Tarigan (2009) yaitu membuat aplikasi pengenalan sidik jari menggunakan wavelet *symlet* dan *wavelet daubechies* menggunakan jaringan saraf tiruan perambatan balik. Hasil dari penelitian ini menunjukkan bahwa berdasarkan hasil data uji *wavelet* induk yang memberikan tingkat pengenalan paling baik adalah jaringan yang menggunakan *wavelet daubechies* 4 (db4) dengan penguraian bertingkat 2 yaitu sebesar 92%.

Hartanto (2010) dalam penelitiannya tentang Pengenalan Iris Mata menggunakan Alihragam *Wavelet Daubechies* Orde 4 menjelaskan bahwa iris dapat digunakan sebagai dasar untuk pengenalan biometrik. Untuk mengidentifikasi tekstur iris dalam gambar mata, metode tekstur analisis dapat digunakan. Ada beberapa metode analisis tekstur, salah satunya adalah dengan menggunakan wavelet berdasarkan pada fitur image energi ekstraksi. Analisis ini

menggunakan karakteristik energi yang terkandung dalam transformasi wavelet. Berdasarkan alasan itu, dalam penelitian program aplikasi ini untuk mengidentifikasi iris mata berdasarkan Transformasi *Wavelet Daubechies* 4. Image Eye digunakan dalam penelitian ini diperoleh dan diproses, mulai mengambil pada karakteristik dan tekstur dari citra iris yang dikonversi ke dalam bentuk polar. Kemudian fitur ekstraksi dilakukan dengan menggunakan transformasi *Wavelet Daubechies* order 4. Karakteristik yang diperoleh dalam bentuk nilai energi. Tahap berikutnya adalah pengakuan menggunakan terdekat dinormalisasi Jarak Euclidean. Pengujian yang dilakukan dalam penelitian ini terdiri dari empat jenis : pengaruh database contoh, pengaruh tingkat dekomposisi transformasi *Wavelet Daubechies* orde 4, pengaruh format gambar input yang berbeda, dan pengujian pada mata gambar yang tidak ada dalam database. Dari hasil pengujian ini, dapat disimpulkan bahwa tingkat pengenalan tertinggi dengan parameter yang ditunjukkan dalam pengujian *Daubechies Wavelet Transform* order 4 tingkat 4 dengan dua sampel iris gambar yang tersimpan adalah 86,66 %. Tingkat pengenalan terendah ditunjukkan dalam tes dengan *Daubechies Wavelet Transform* order 4 level 6 dengan satu gambar sampel iris disimpan adalah 62,5 %. Kemudian dari hasil pengujian pengaruh format gambar input yang berbeda, dapat disimpulkan bahwa sampel diambil dari 40 individu yang satu sampel adalah mengambil untuk setiap orang, menggunakan format BMP serta dengan menggunakan format JPEG. Sedangkan dari hasil pengujian untuk gambar mata yang tidak ada dalam database dengan ambang batas 0,3559, tingkat pengenalan adalah 96 %.

IV. *Non Overlapping Hierarchical Index Structure-Tree (NOHIS-Tree).*

Mounira Taileb and Sid Lamrous and Sami Touati (2008), pada forum *World Academy of Science, Engineering and Technology, International Science Index 13* dalam penelitiannya yang berjudul “*Non-Overlapping Hierarchical Index Structure for Similarity Search*” yang bertujuan untuk mempercepat pencarian kesamaan dalam *database* berdimensi tinggi. Diusulkan sebuah metode pengindeksan hirarkis baru, terdiri dari fase *offline* dan fase *online*. Pada fase *offline*, setelah mengumpulkan seluruh data dalam cluster dan membangun indeks hirarkis, orisinalitas utama mengembangkan metode dengan membangun bentuk loncat dari cluster untuk menghindari tumpang tindih. Untuk tahap *online*, ide ini untuk meningkatkan akurasi pencarian kesamaan. Namun untuk tahap kedua, dalam penelitian ini juga telah dikembangkan sebuah algoritma pencarian yang disesuaikan. Metode ini disebut *NOHIS (Non Overlapping Hierarchical Index Structure-Tree)* menggunakan *Principal Direction Divisive Partitioning (PDDP)* atau *Principal Arah Memecah belah Partisi* sebagai algoritma clustering. Prinsip *PDDP* adalah untuk membagi data yang secara rekursif menjadi dua *sub-cluster*, pembagian dilakukan dengan menggunakan *hyper-plane* ortogonal terhadap arah utama berasal dari matriks kovarians dan melewati centroid cluster untuk membagi. Data dari masing-masing dua *sub-cluster* diperoleh dengan membentuk *Minimum Bounding Rectangle (MBR)*. Kedua MBRs diarahkan sesuai dengan arah utama. Sehingga *Non-Overlapping* antara dua bentuk terjamin. Percobaan menggunakan *database* yang berisi deskripsi gambar. Hasil penelitian ini

menunjukkan bahwa metode yang diusulkan lebih baik dari scan sekuensial dan SR-tree dalam pengolahan menggunakan *K-Nearest Neighbors*.

Mounira Taileb and Sami Touati (2011) dalam penelitian yang berjudul “*NOHIS-Tree: High-Dimensional Index Structure for Similarity Search*”, menerapkan sistem *Content Based Image Retrieval* yang menggunakan pengindeksan yang efisien untuk melakukan dan mempercepat pencarian dalam database besar. Dalam penelitian ini disajikan indeks hirarkis *NOHIS-tree* (*Non Overlapping Hierarchical Index Structure* atau bebas tumpang tindih indeks struktur hirarkis) ketika berskala ke database yang sangat besar. Penelitian ini juga menghadirkan sebuah studi dari pengaruh pengelompokan pada waktu pencarian. Hasil tes kinerja menunjukkan bahwa *NOHIS-tree* melakukan lebih baik daripada *SR-tree*. Pengujian juga menunjukkan bahwa *NOHIS-tree* membuat kinerja dalam ruang dimensi tinggi. Terakhir disertakan tes kinerja yang mencoba untuk menentukan jumlah kluster di *NOHIS-tree* yang memiliki waktu pencarian terbaik.

Mounira Taileb (2013) dalam penelitiannya tentang “*Content Based Image Retrieval System Using NOHIS-tree*” menjelaskan bahwa *CBIR* telah menjadi salah satu area penelitian terpenting dalam *computer vision*. Ini adalah metode yang banyak digunakan untuk mencari gambar dalam database besar. Dalam penelitian ini disajikan sebuah sistem *CBIR* yang disebut *NOHIS-Search*. Sistem ini didasarkan pada teknik pengindeksan *NOHIS-tree*. Dua fase dari sistem dijelaskan dan kinerja sistem digambarkan dengan database gambar *ImageEval*. Sistem *NOHIS-Search* dibandingkan dengan dua sistem *CBIR* lain, yang pertama

yang menggunakan algoritma pengindeksan *PDDP* (*Principal Direction Divisive Partitioning*) dan sistem kedua menggunakan Pencarian Sekuensial (*Sequential Search*). Hasil penelitian menunjukkan bahwa sistem *NOHIS-search* melebihi dua sistem lain.



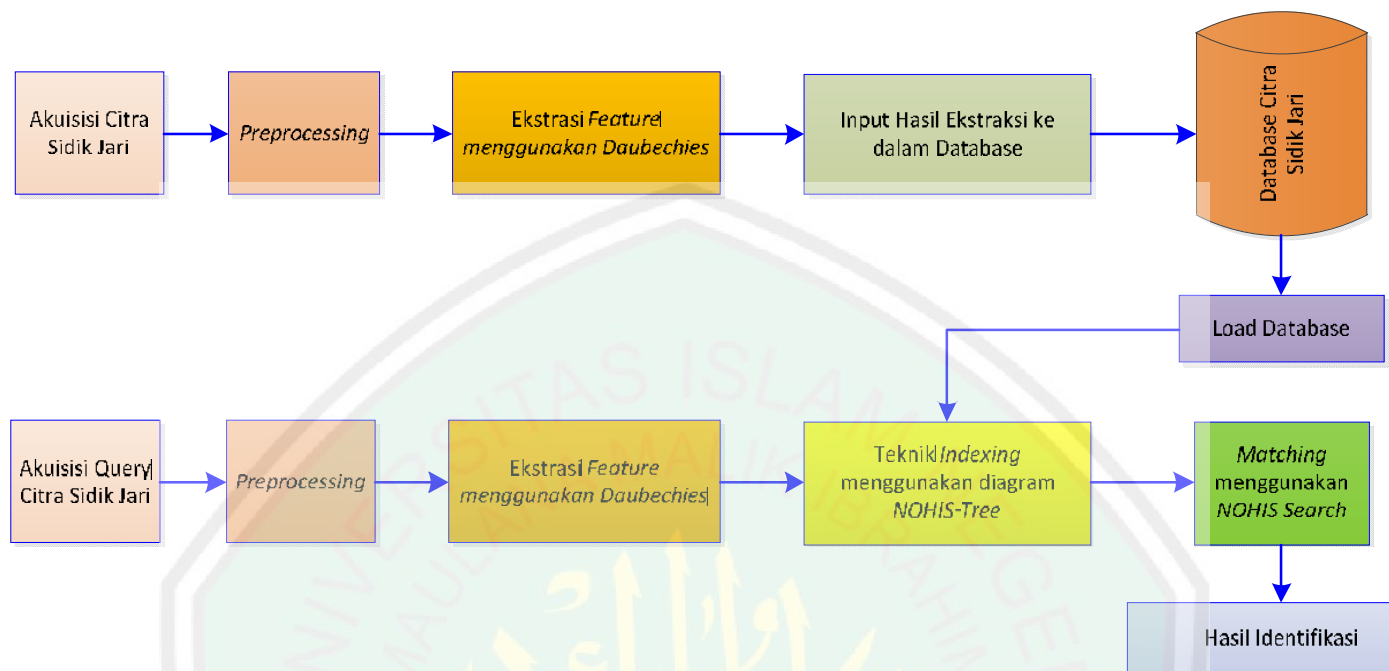
BAB III

PERANCANGAN DAN IMPLEMENTASI APLIKASI

3.1 Desain Proses

Desain Proses pada penelitian ini memiliki input berupa citra sidik jari yang berukuran 512 x 512 piksel. Citra sidik jari tegak lurus digunakan sebagai citra yang disimpan di dalam database. Citra inputan melalui proses *thresholding* terlebih dahulu. Proses ini bertujuan untuk menyederhanakan nilai matriks yang ada didalam sebuah citra. Setelah citra diubah ke dalam skala *black and white*, citra diekstraksi menggunakan metode transformasi *wavelet daubechies* sehingga menghasilkan empat nilai koefisien yaitu nilai aproksimasi, nilai horizontal, nilai vertikal dan nilai diagonal.

Hasil perhitungan nilai energi yang berupa nilai aproksimasi dan nilai koefisien detail horizontal, vertikal, serta diagonal ini yang kemudian akan disimpan ke dalam database. Dalam penyusunannya, data input yang di masukkan ke dalam database akan dianalisa manakah yang cocok untuk penerapan teknik pengindeksan berdasarkan diagram *NOHIS-Tree*. Hasil dari analisa berupa dua nilai koefisien yang sesuai akan menjadi titik acuan dari data untuk dikelompokkan atau diklasifikasikan menggunakan diagram pohon *NOHIS tree* berdasarkan nilai-nilai tersebut. Proses identifikasinya menggunakan Algoritma *NOHIS-search* dengan memperbandingkan menggunakan *euclidian distance* antara citra *query* dengan citra pada database untuk memperoleh waktu pencarian yang optimal. Berikut ini diagram blok dari aplikasi yang akan dibangun :



Gambar 3.1 Blok Diagram Proses Aplikasi CBIR Citra Sidik Jari dengan menggunakan Metode Wavelet Daubechies dan Diagram NOHIS-Tree

Urutan penjelasan dari desain proses aplikasi yang akan dibangun tersebut antara lain, yaitu akuisisi citra, preprocessing, dan ekstraksi fitur yang dilakukan terhadap citra input dan citra *query*. Selanjutnya, proses penyimpanan hasil ekstraksi fitur citra sidik jari ke *database*. Proses terakhir yaitu klasifikasi atau teknik pengindeksan berdasarkan diagram *NOHIS-Tree* dan proses terakhir adalah *Similarity Search* atau proses identifikasi menggunakan *NOHIS-Search*.

3.1.1. Akuisisi Citra Sidik Jari

Akuisisi citra merupakan proses awal untuk mendapatkan citra digital. Pada penelitian ini, akuisisi citra dimulai dengan pengambilan citra sidik jari dengan cara menggunakan tinta stempel yang ditempelkan pada kertas kemudian di *scan*. Pada penelitian ini, alat *scanner* yang digunakan adalah Brother DCP195C dengan resolusi 400dpi dan menghasilkan citra berukuran 4677 x 3400 piksel. Hasil *scanning* ditunjukkan pada gambar 3.2.



Gambar 3.2 Hasil scanning citra sidik jari

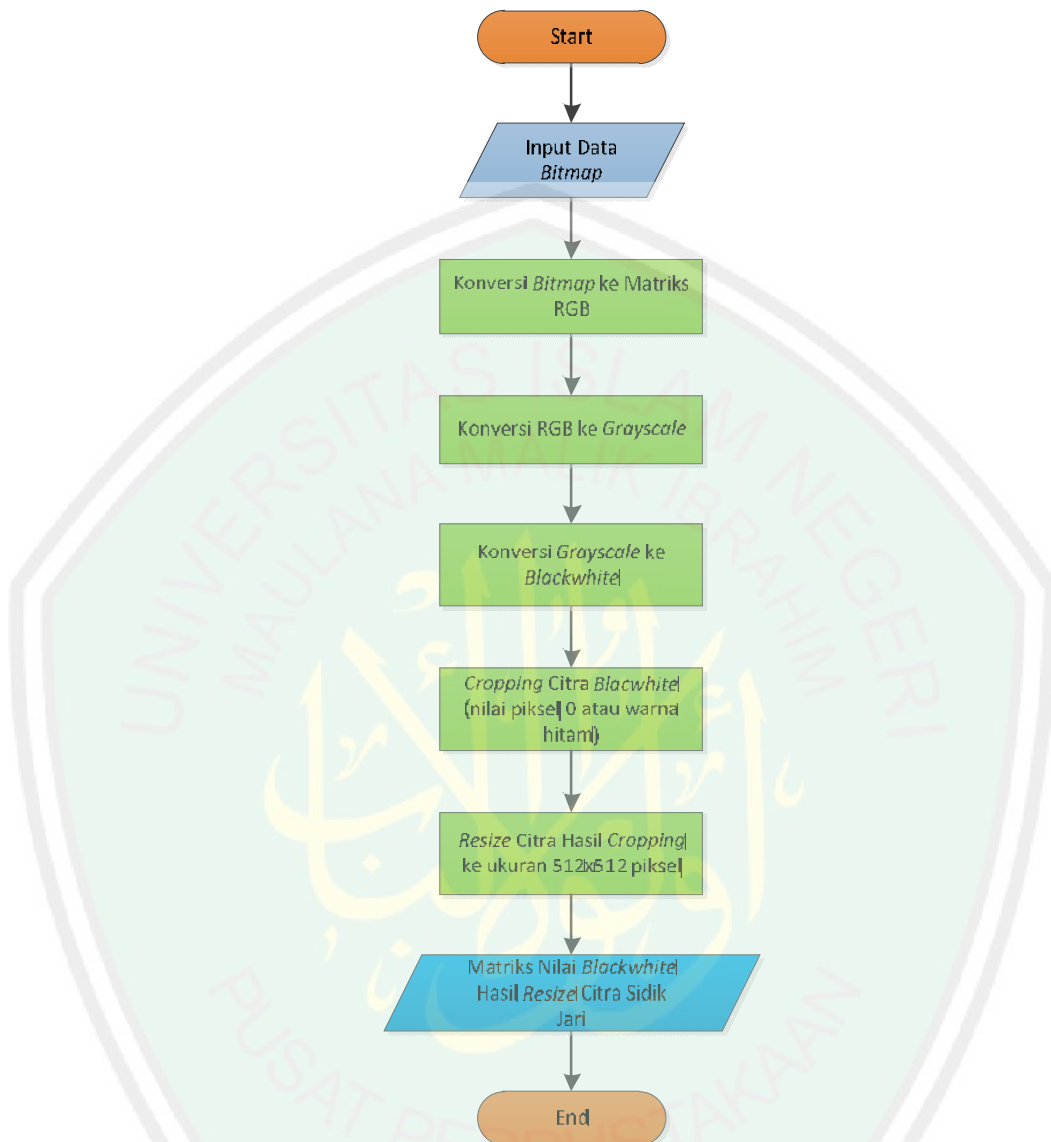
Pada tahap ini, citra hasil *scanning* di-*crop* pada masing-masing sidik jari dengan ukuran setiap sidik jari 512x512 pixel. Gambar 3.3 menunjukkan hasil *cropping* dengan ukuran 512x512 pixel.



Gambar 3.3 Hasil cropping citra sidik jari berukuran 512x512 piksel

3.1.2. *Preprocessing* Citra Sidik Jari

Sebelum citra masukan diproses lebih lanjut, perlu dilakukan proses awal (*preprocessing*) terlebih dahulu dengan tujuan agar mendapatkan hasil yang maksimal disaat proses identifikasi untuk dapat menghasilkan identifikasi yang terbaik. Berikut ini adalah diagram alur untuk proses *preprocessing* pada akuisisi citra sidik jari pada aplikasi untuk proses input citra sidik jari :



Gambar 3.4 Diagram Alur Akuisisi Citra Sidik Jari dan Preprocessing

Preprocessing disini terdapat beberapa hal-hal penting yang akan dilakukan di antaranya adalah : *thresholding*, *cropping* dan *resize*. Dalam proses *threshold* terdapat beberapa proses seperti dijelaskan pada diagram alur akuisisi dan *preprocessing* citra sidik jari. Tiga proses utama pada *thresholding* yaitu konversi data bitmap ke matriks citra RGB (*Red Green Blue*), konversi citra RGB ke citra *grayscale* dan konversi citra *grayscale* ke citra *blackwhite*. Sedangkan

proses *cropping* dan *resize* termasuk pada proses selanjutnya pada akuisisi citra sidik jari dan *preprocessing*.

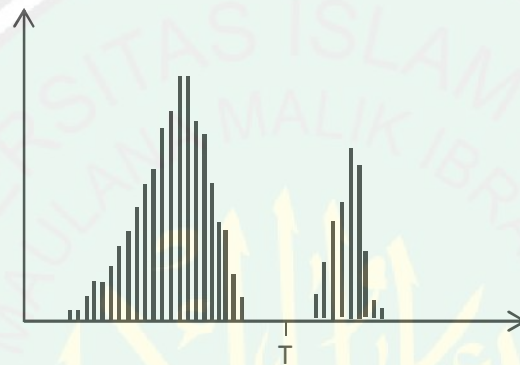
1. *Thresholding*

Thresholding adalah memisahkan citra ke dalam daerah intensitasnya masing-masing sehingga bisa dibedakan antara objek dan *background*. Citra sidik jari yang telah berukuran 512x512 piksel masih dalam merupakan citra warna (*true color*) yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap piksel dari citra *true color* diwakili oleh 3 *byte*, dimana masing-masing *byte* mempresentasikan warna merah (*Red*), hijau (*Green*), dan biru (*Blue*). Pada tahap segmentasi, Citra RGB ini kemudian di konversi ke citra *grayscale* atau citra keabuan kemudian dilanjutkan dengan konversi dari citra *grayscale* ke citra biner. Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner juga disebut sebagai citra B & W (*black and white*) atau citra monokrom. Hanya dibutuhkan 1 *byte* untuk mewakili nilai setiap piksel dari citra biner.

Histogram yang ditunjukkan pada gambar 3.5 yang berkaitan dengan citra (x,y) yang terdiri dari objek terang pada *background* gelap, maka piksel objek dan *background* mempunyai level intensitas yang dikelompokkan ke dalam dua mode domain. Satu cara yang jelas untuk mengekstrak objek dari *background* adalah dengan memilih *threshold* T yang membagi mode-mode ini. Kemudian sembarang titik (x, y) untuk dimana $f(x, y) \geq T$ disebut *object point*. Sedangkan yang lain disebut *background point*. Dengan kata lain, citra yang di-*threshold* $g(x,y)$ dedefinisikan sebagai :

$$g(x, y) = \begin{cases} 1 & \text{jika } f(x, y) \geq T \\ 0 & \text{jika } f(x, y) < T \end{cases}$$

Piksel yang diberi nilai 1 berkaitan dengan objek sedangkan piksel yang diberi nilai 0 berkaitan dengan *background*. Ketika T adalah konstanta, pendekatan ini disebut *global thresholding* (Eko Prasetyo, 2011:221).



Gambar 3.5 Pemilihan *threshold* secara analisis visual histogram

Salah satu cara untuk memilih *thresholding* adalah dengan pemeriksaan visual histogram citra. histogram dalam gambar 3.5 secara jelas mempunyai dua mode yang berbeda. Sebagai hasilnya, mudah untuk memilih *threshold* T yang membaginya. Metode yang lain dalam memilih T adalah dengan *train and error*, mengambil beberapa *threshold* berbeda sampai satu nilai T yang memberikan hasil yang baik sebagai keputusan observer ditemukan.

Untuk pemilihan *threshold* secara otomatis, prosedurnya dijelaskan sebagai berikut (Eko Prasetyo, 2011 : 222) :

- Pilih nilai T awal, disarankan perkiraan awal adalah titik tengah antara nilai intensitas minimum dan maksimum citra.

- Mensegmentasi citra menggunakan T . Ini akan menghasilkan dua kelompok piksel : G_1 , yang berisi semua nilai dengan nilai intensitas $\geq T$, dan G_2 , yang berisi semua piksel dengan nilai intensitas $< T$.
- Menghitung nilai rata-rata intensitas μ_1 dan μ_2 masing-masing untuk piksel dalam region G_1 dan G_2 .
- Hitung nilai *threshold* yang baru dengan rumus $T = \frac{\mu_1 + \mu_2}{2}$
- Ulangi langkah 2 sampai 4 hingga nilai rata-rata μ_1 dan μ_2 tidak berubah lagi.

Berikut ini adalah *source code* dari pencarian nilai T yang mana nilai T tersebut digunakan sebagai batas ambang *threshold*.

```
function T= metode_threshold(data_proses)

T= 0.5* (double(min(data_proses(:))) +
double(max(data_proses(:))));
done= false;
while ~done
    g= data_proses >= T;
    Tnext = 0.5 * (mean(data_proses(g)) + mean
        (data_proses(~g)));
    done = abs (T-Tnext) < 0.5;
    T = Tnext;
End
```

Gambar 3.6 *Source Code* dari Pencarian Nilai T sebagai Batas Ambang *Threshold*.

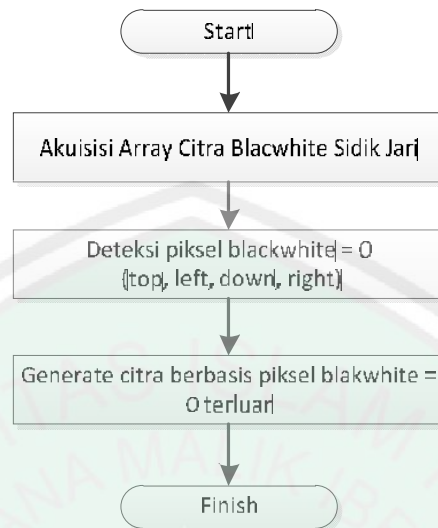
Berikut ini menunjukkan proses thresholding pada citra sidik jari.



Gambar 3.7 Hasil Threshold Citra Sidik Jari

2. *Region of Interest (ROI)* dan *Resize*

Perbaikan citra atau *image restoration* dalam penelitian ini digunakan untuk mendapatkan hasil yang lebih optimal sebelum citra sidik jari melalui proses transformasi *wavelet daubechies*. Tahapan perbaikan citra tersebut yaitu *ROI* citra blackwhite dan *resize* citra hasil *ROI*. *Region of Interest* citra *blackwhite* ini bertujuan untuk membaca atau mengambil citra yang bernilai piksel hitam saja dan menghilangkan noise atau menghilangkan nilai piksel putih yang ada disekitarnya sehingga diperoleh gambar sidik jari yang lebih akurat atau mengambil objeknya saja dan mengabaikan background yang tidak digunakan. Berikut ini adalah diagram alur proses *ROI*:



Gambar 3.8 Diagram alur ROI citra blackwhite sidik jari

Berikut ini listing *source code* untuk membaca objek citra sidik jari dari citra biner atau citra *blackwhite* sidik jari.

```
function [x1,y1,w,h] = metode_cropping(data_crop)
```

```
baris=size(data_crop,1);
kolom=size(data_crop,2);
```

```
Xterendah = baris;
Yterendah = kolom;
Xtertinggi = 0;
Ytertinggi = 0;
```

```
for i=1:baris
    for j=1:kolom
        nilai = data_crop(j,i);
        if(nilai==0)
            if(i<Xterendah)
                Xterendah=i;
            end
            if(j<Yterendah)
                Yterendah=j;
            end
            if(i>Xtertinggi)
                Xtertinggi=i;
            end
            if(j>Ytertinggi)
                Ytertinggi=j;
            end
        end
    end
end
```

```

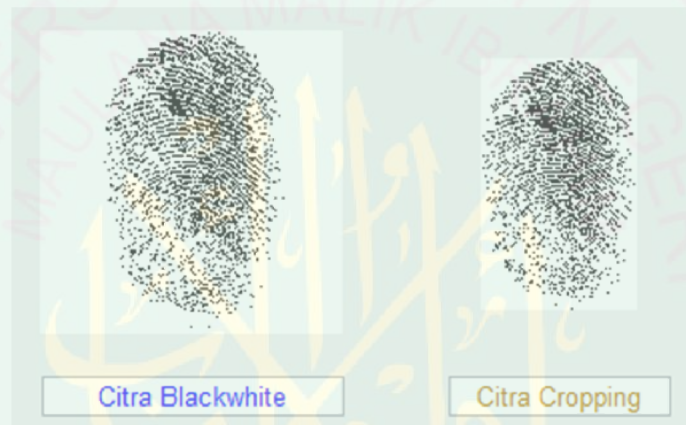
    end
  end
end

x1 = Xterendah;
y1 = Yterendah;
w = Xtertinggi-Xterendah;
h = Ytertinggi-Yterendah;

```

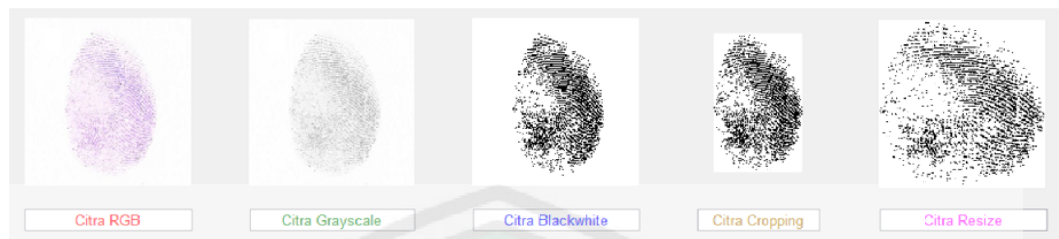
Gambar 3.9 listing source code untuk membaca citra sidik jari blackwhite

Berikut ini hasil *Region of Interest* citra blackwhite dengan nilai piksel B/W=0.



Gambar 3.10 Hasil ROI citra blackwhite dengan nilai piksel B/W = 0 terluar

Sedangkan citra resize bertujuan untuk mengembalikan hasil citra ROI blackwhite tersebut ke ukuran semula yaitu 512x512 piksel agar memudahkan proses perhitungan ketika dilakukan transformasi *wavelet daubechies* pada citra sidik jari. Gambar 3.11 menunjukkan hasil proses preprocessing yang meliputi threshold, ROI dan resize citra sidik jari.



Gambar 3.11 Hasil preprocessing yang meliputi proses thresholding, ROI dan resize citra sidik jari

Berikut ini adalah *source code* dari proses akuisisi citra sidik jari :

```
% --- Executes on button press in pushbutton1.
%%% --- Input Data Citra RGB --- %%%
function pushbutton1_Callback(hObject, eventdata, handles)
[name_file1,name_path1] = uigetfile( ...
{'*.bmp;*.jpg;*.tif','Files of type (*.bmp,*.jpg,*.tif)';
'*.bmp','File Bitmap (*.bmp)';...
'*.jpg','File jpeg (*.jpg)';
'*.tif','File Tif (*.tif)';
'*.','All Files (*.*)'},...
'Open Image');

if ~isequal(name_file1,0)
citrargb = imread(fullfile(name_path1,name_file1));
guidata(hObject,handles);
axes(handles.axes1);
imshow(citrargb);
set(handles.edit6, 'String', fullfile(name_path1,name_file1))
% size(citrargb)

%%% --- Proses Citra RGB ke Citra Grayscale --- %%%
grayscale=rgb2gray(citrargb);
guidata(hObject,handles);
axes(handles.axes2);
imshow(grayscale);
% size (grayscale)

%%% ---- Proses Citra Grayscale ke Citra Blackwhite --- %%%
blackwhite=zeros(size(grayscale));
nilaibatas=metode_threshold(grayscale);
nilaibatas;
for i=1:size(grayscale,1)
    for j=1:size(grayscale,2)
        if grayscale(i,j)<nilaibatas
            blackwhite(i,j)=0;
        else
            blackwhite(i,j)=255;
        end
    end
end
```

```

end
end

guidata(hObject,handles);
axes(handles.axes3);
imshow(blackwhite);
% Matriks_Blackwhite = size(blackwhite)

%%% --- Proses Cropping Sidik Jari dari Matriks Blackwhite ----
[x1,y1,w,h]=metode_cropping(blackwhite);
cropping = imcrop(blackwhite,[x1 y1 w h]);
guidata(hObject,handles);
axes(handles.axes4);
imshow(cropping);
% Matriks_Cropping = size(cropping)

%%% --- Proses Resize Sidik Jari ke Matriks [512 x512] --- %%%
resize = imresize(cropping, [512 512]);
guidata(hObject,handles);
axes(handles.axes5);
imshow(resize);
% Matriks_Resize = size(resize)

save data_threshold resize

else
return;
end

```

Gambar 3.12 Source Code dari Proses Akuisisi Citra Sidik Jari

3.1.3. Ekstraksi Fitur menggunakan *Wavelet Daubechies*

Daubechies adalah salah satu keluarga *Wavelet*, lebih tepatnya keluarga *Wavelet* yang bersifat orthogonal. *Wavelet* ini ditemukan oleh Ingrid Daubechies. Sedangkan dalam penelitian ini penulis menggunakan *wavelet daubechies* orde 4 (*db 4*) sebagai tetapan kernel *daubechies*-nya karena pada penelitian-penelitian sidik jari sebelumnya (tinjauan pustaka Bab II), telah terbukti lebih efektif dari *wavelet* jenis lainnya.

Ingrid Daubechies (1988) meneliti tentang Orthonormal Bases of Compactly Supported Wavelets. Sedangkan pada tahun 1992 Cohen, Daubechies

and Feauveau melanjutkan penelitian tentang tetapan *wavelet* yang berjudul Biorthogonal Bases of Compactly Supported Wavelets. Berikut ini perbedaan antara koefisien untuk kernel *orthogonal wavelet daubechies* pada tabel 3.1 dan koefisien untuk kernel *orthonormal wavelet daubechies* pada tabel 3.2 beserta skema *filter bank*-nya yang ada pada penelitian-penelitian tersebut.

Tabel 3.1 *Orthogonal Daubechies Coefficients D2-D12 (Low-Pass Filter)*

D2	D4	D6	D8	D10	D12
1	0.6830127	0.47046721	0.32580343	0.22641898	0.15774243
1	1.1830127	1.14111692	1.01094572	0.85394354	0.69950381
	0.3169873	0.650365	0.8922014	1.02432694	1.06226376
	-0.1830127	-0.19093442	-0.03957503	0.19576696	0.44583132
		-0.12083221	-0.26450717	-0.34265671	-0.31998660
		0.0498175	0.0436163	-0.04560113	-0.18351806
			0.0465036	0.10970265	0.137888809
			-0.01498699	-0.00882680	-0.03892321
				-0.01779187	-0.04466375
				4.71742793e-3	7.83251152e-4
					6.75606236e-3
					-1.52353381e-3

Bank Filter (Low-Pass) *Wavelet Daubechies* dengan 4 koefisien pada *Orthogonal Bases* adalah sebagai berikut :

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 \end{bmatrix}$$

Tabel 3.2 *Orthonormal Daubechies Coefficients db4*

Low Pass		Koefisien
N=4	H(0)	0.230377813308855
	H(1)	0.714846570552542
	H(2)	0.630880767929590
	H(3)	- 0.0279837694169839
	H(4)	- 0.187034811718881
	H(5)	0.0308413818359870
	H(6)	0.0328830116669829
	H(7)	- 0.0105974017849973

Bank Filter (Low-Pass) *Wavelet Daubechies* dengan 4 koefisien pada *Orthonormal Bases* adalah sebagai berikut :

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 & 0 & 0 \end{bmatrix}$$

Tetapan jenis koefisien yang digunakan dalam penelitian ini adalah *Orthogonal Daubechies Coefficients* karena sangat mendukung untuk diterapkan dengan metode teknik *indexing* berdasarkan diagram *NOHIS-Tree* yang pada

tahapannya terdapat *Separating Hyper-Plane Orthogonal* sehingga bisa diperoleh hasil yang optimal berdasarkan pada penelitian-penelitian sebelumnya.

Urutan koefisien *Wavelet Daubechies Orthogonal* dapat diperoleh dari tabel *Orthogonal Daubechies Coefficients D2-D12 (Low-Pass Filter)* dengan cara sebagai berikut:

- a. Balik urutan koefisien fungsi penskalaan.
- b. Balik tanda (minus atau plus) untuk setiap koefisien dengan posisi habis di modulo 2.

Sebagai contoh untuk D4, dengan koefisien fungsi penskalaan(0.6830127, 1.1830127, 0.3169873, -0.1830127) maka koefisien fungsi *wavelet*-nya dapat diperoleh dengan menggunakan langkah diatas, sebagai berikut:

Pertama, lakukan pembalikan terhadap urutan koefisien fungsi penskalaan, sehingga menjadi (-0.1830127, 0.3169873, 1.1830127, 0.6830127).

Kedua, balik tanda untuk setiap koefisien pada posisi habis dimodulo 2, sehingga koefisien fungsi *wavelet* untuk D4 adalah (-0.1830127, -0.3169873, 1.1830127, -0.6830127).

Kedua tahap diatas dapat ditulis dalam bentuk persamaan berikut.

$$[b_k = (-1)^k a_{N-1-k}]$$

Dengan k menyatakan indeks koefisien, b menyatakan urutan koefisien *Wavelet*, dan a menyatakan koefisien penskalaan. N menyatakan indeks *Wavelet*, sebagai contoh untuk D2 maka N=2 dan untuk D4 maka N=4.

$$b_0 = (-1)^0 a_3 = a_3 = -0.1830127$$

$$b_1 = (-1)^1 a_2 = -a_2 = -0.3169873$$

$$b_2 = (-1)^2 a_1 = a_1 = 1.1830127$$

$$b_3 = (-1)^3 a_0 = -a_0 = -0.6830127$$

Mengikuti skema *filter bank* pada *Orthogonal Bases* maka matriks H_0 dan H_1 untuk D4, pada citra ukuran D4 adalah sebagai berikut :

$H_0 =$

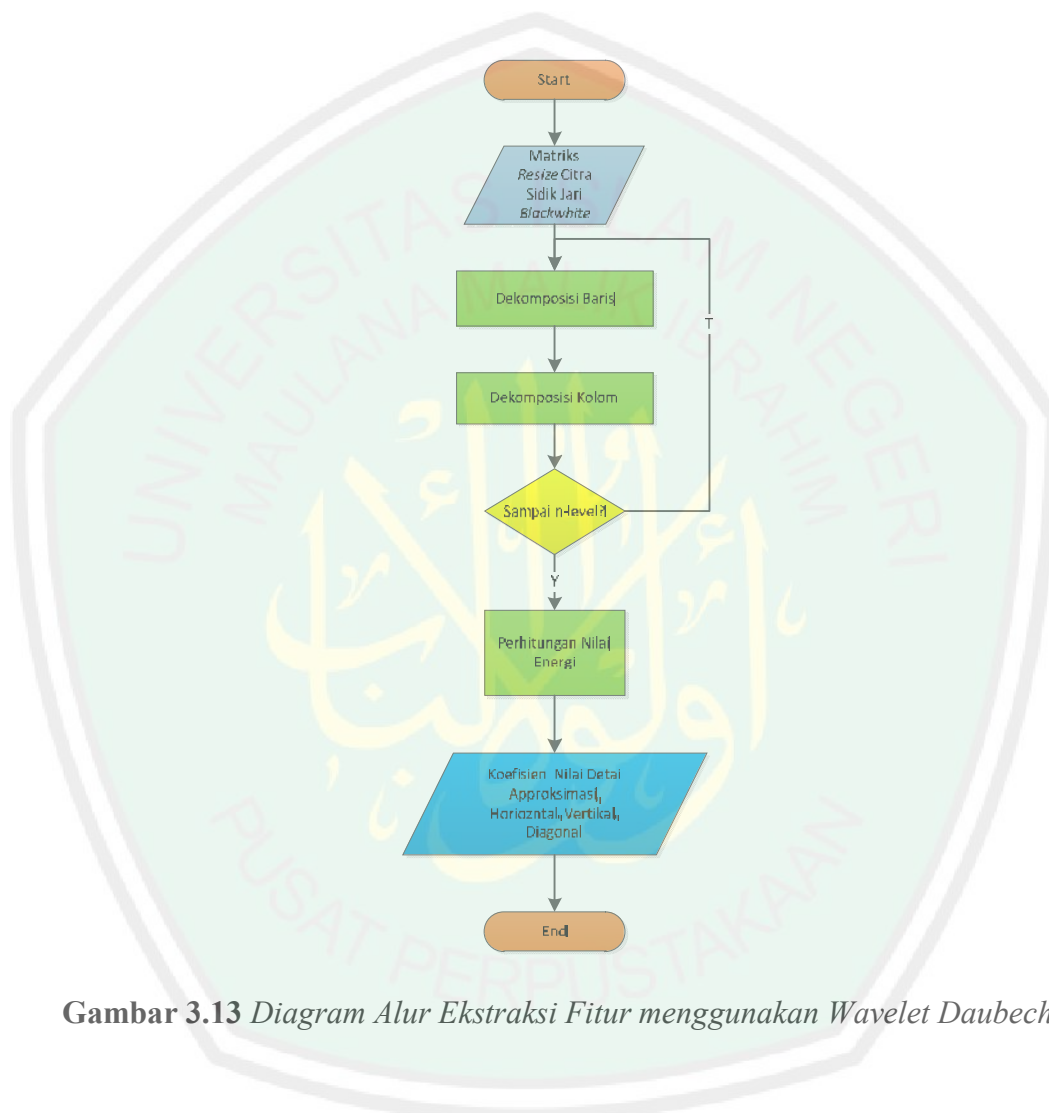
$$\begin{bmatrix} 0.6830127 & 1.1830127 & 0.3169873 & -0.1830127 & 0 & 0 & 0 & 0 \\ 0 & 0.6830127 & 1.1830127 & 0.3169873 & -0.1830127 & 0 & 0 & 0 \\ 0 & 0 & 0.6830127 & 1.1830127 & 0.3169873 & -0.1830127 & 0 & 0 \\ 0 & 0 & 0 & 0.6830127 & 1.1830127 & 0.3169873 & -0.1830127 & 0 \\ 0 & 0 & 0 & 0 & 0.6830127 & 1.1830127 & 0.3169873 & -0.1830127 \\ 0 & 0 & 0 & 0 & 0 & 0.6830127 & 1.1830127 & 0.3169873 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6830127 & 1.1830127 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6830127 \end{bmatrix}$$

$H_1 =$

$$\begin{bmatrix} -0.1830127 & -0.3169873 & 1.1830127 & -0.6830127 & 0 & 0 & 0 & 0 \\ 0 & -0.1830127 & -0.3169873 & 1.1830127 & -0.6830127 & 0 & 0 & 0 \\ 0 & 0 & -0.1830127 & -0.3169873 & 1.1830127 & -0.6830127 & 0 & 0 \\ 0 & 0 & 0 & -0.1830127 & -0.3169873 & 1.1830127 & -0.6830127 & 0 \\ 0 & 0 & 0 & 0 & -0.1830127 & -0.3169873 & 1.1830127 & -0.6830127 \\ 0 & 0 & 0 & 0 & 0 & -0.1830127 & -0.3169873 & 1.1830127 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.1830127 & -0.3169873 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1830127 \end{bmatrix}$$

Pada penelitian ini, transformasi *Wavelet Daubechies* dilakukan hingga level 2. Citra yang awalnya memiliki ukuran 512 x 512 piksel, setelah diekstraksi

feature akan menghasilkan ukuran 128 x 128 piksel. Diagram alur transformasi Wavelet 2D digambarkan pada Gambar 3.13 berikut ini.

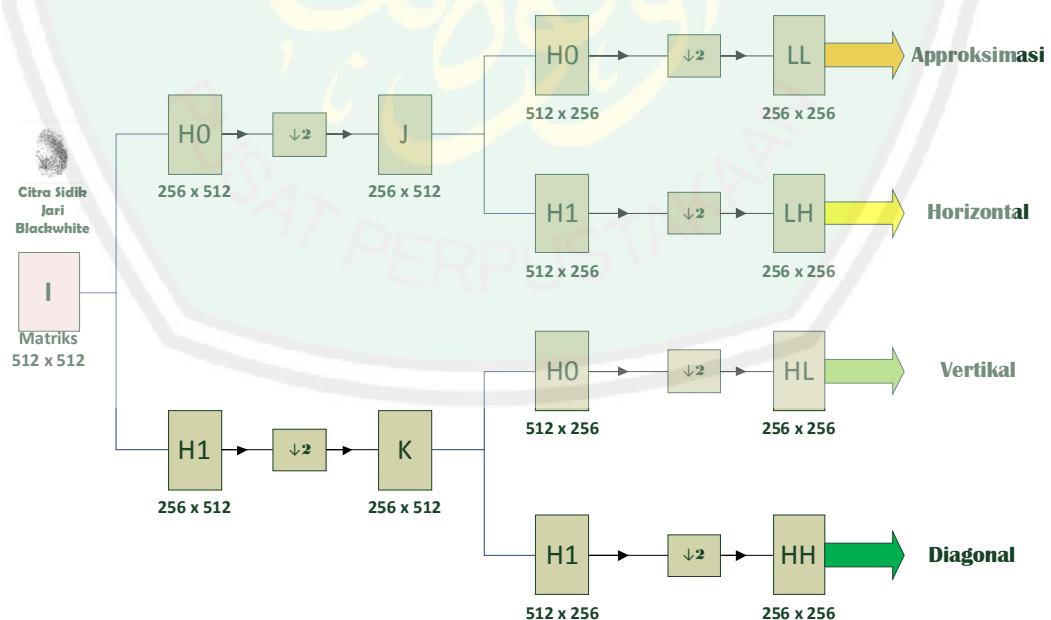


Gambar 3.13 Diagram Alur Ekstraksi Fitur menggunakan Wavelet Daubechies

Adapun penjelasan langkah pemrosesan transformasi Wavelet sebagai berikut :

- Menentukan matrik nilai citra berukuran 512 x 512 piksel.
- Menentukan matrik dekomposisi *low pass* dan matrik dekomposisi *high pass* yang berukuran 256x512 piksel.

- Mengalikan matriks *low pass* dengan matriks citra dan mengalikan matriks *high pass* dengan matriks citra, menghasilkan matriks berukuran 256x512.
 - Men-transpose matriks hasil perkalian *low pass* dan matriks hasil perkalian *high pass*, sehingga matriks menjadi berukuran 512x256.
 - Mengalikan matriks dekomposisi *low pass* dengan matriks transpose hasil perkalian *low pass* (LL) dan mengalikan matriks dekomposisi *low pass* dengan matriks transpose hasil perkalian *high pass* (LH).
 - Mengalikan matriks dekomposisi *high pass* dengan matriks transpose hasil perkalian *low pass* (HL) dan mengalikan matriks dekomposisi *high pass* dengan matriks transpose hasil perkalian *high pass* (HH).
 - Level 2 mengulangi langkah c sampai f, dengan menggunakan bagian LL.
- Berikut Contoh Dekomposisi menggunakan Daubechies Level 1.



Gambar 3.14 Dekomposisi menggunakan Wavelet Daubechies level 1

Setelah level 2 berakhir, langkah selanjutnya yaitu perhitungan energi pada masing-masing bagian yaitu bagian aproksimasi (C_A), bagian horizontal (C_V), bagian vertikal (C_H), dan bagian diagonal (C_D).

Perhitungan Energi pada *Wavelet*

Perhitungan energi berfungsi untuk menghitung energi yang dihasilkan setiap citra hasil alihragam wavelet. Dalam penelitian ini, energi dibagi dalam 4 (empat) ciri, yaitu:

- a. Prosentase energi yang berhubungan dengan nilai pendekatan (aproksimasi), E_a dihitung berdasarkan prosentase jumlahan kuadrat dari nilai koefisien aproksimasi C_a dibagi dengan jumlahan seluruh koefisien C (koefisien aproksimasi ditambah koefisien detail).

$$E_a = \frac{\sum(C_a)^2}{\sum(C)^2} \times 100\%$$

- b. Prosentase energi yang berhubungan dengan nilai detail pada arah horisontal, E_h dihitung berdasarkan prosentase jumlahan kuadrat dari nilai koefisien detail pada arah horisontal C_h dibagi dengan jumlahan seluruh koefisien C .

$$E_h = \frac{\sum(C_h)^2}{\sum(C)^2} \times 100\%$$

- c. Prosentase energi yang berhubungan dengan nilai detail pada arah vertikal, E_v dihitung berdasarkan prosentase jumlahan kuadrat dari nilai

koefisien detail pada arah horisontal C_v dibagi dengan jumlahan seluruh koefisien C .

$$E_v = \frac{\sum(C_v)^2}{\sum(C)^2} \times 100\%$$

d. Prosentase energi yang berhubungan dengan nilai detail pada arah diagonal, E_d dihitung berdasarkan prosentase jumlahan kuadrat dari nilai koefisien detail pada arah diagonal C_d dibagi dengan jumlahan seluruh koefisien C .

$$E_d = \frac{\sum(C_d)^2}{\sum(C)^2} \times 100\%$$

Hasil dari perhitungan energi dapat dilihat sebagai berikut :



Gambar 3.15 Hasil Perhitungan Energi

Berikut ini adalah *source code* untuk transformasi Wavelet Daubechies berserta perhitungan energi *wavelet daubechies* :

```
function [apr_total, hor_total, ver_total, diag_total] =
```

```

metode_wavelet (resize)

%=====
%level 1
%=====
resize = double(resize);
mdb11 = matriks_dekomposisi_baris_low(resize);
mdbh1 = matriks_dekomposisi_baris_high(resize);
% matriks_dekomposisi_baris_1 = size(mdbh1)
% matriks_resize = size(resize)

hasil_low1= mdb11 * resize;
hasil_high1= mdbh1 * resize;
% matriks_hasilkali_mdb1_n_resize = size(hasil_high1)

mdk11 = matriks_dekomposisi_kolom_low(resize);
mdkh1 = matriks_dekomposisi_kolom_high(resize);
% matriks_dekomposisi_kolom_1 = size(mdkh1)

LL1 = hasil_low1 * mdk11 ;
% matriks_hasil_level_1 = size(LL1)
LH1 = hasil_high1 * mdk11;
HL1 = hasil_low1 * mdkh1;
HH1 = hasil_high1 * mdkh1;

% =====
% level 2
% =====
mdb12 = matriks_dekomposisi_baris_low(LL1);
mdbh2 = matriks_dekomposisi_baris_high(LL1);
% matriks_dekomposisi_baris_2 = size(mdbh2)

hasil_low2= mdb12 * LL1;
hasil_high2= mdbh2 * LL1;
% matriks_hasilkali_mdb2_n_resize = size(hasil_high2)

mdk12 = matriks_dekomposisi_kolom_low(LL1);
mdkh2 = matriks_dekomposisi_kolom_high(LL1);
% matriks_dekomposisi_kolom_2 = size(mdkh2)

LL2 = hasil_low2 * mdk12 ;
% matriks_hasil_level_2 = size(LL2)
LH2 = hasil_high2 * mdk12;
HL2 = hasil_low2 * mdkh2;
HH2 = hasil_high2 * mdkh2;

%=====
%perhitungan energi wavelet
%=====
baris_lv2 = size (LL2,1);
kolom_lv2 = size (LL2,2);
sum_apr=0;
sum_hor=0;
sum_ver=0;

```

```

sum_diag=0;

for i=1:baris_lv2
    for j=1:kolom_lv2
        sum_apr = sum_apr + LL2(i,j);
        sum_hor = sum_hor + LH2(i,j);
        sum_ver = sum_ver + HL2(i,j);
        sum_diag = sum_diag + HH2(i,j);
    end
end

sum_semuanya = sum_apr+ sum_hor+sum_ver+sum_diag;

apr_total = ((sum_apr^2)/((sum_semuanya)^2))*100;
hor_total = ((sum_hor^2)/((sum_semuanya)^2))*100;
ver_total = ((sum_ver^2)/((sum_semuanya)^2))*100;
diag_total = ((sum_diag^2)/((sum_semuanya)^2))*100;

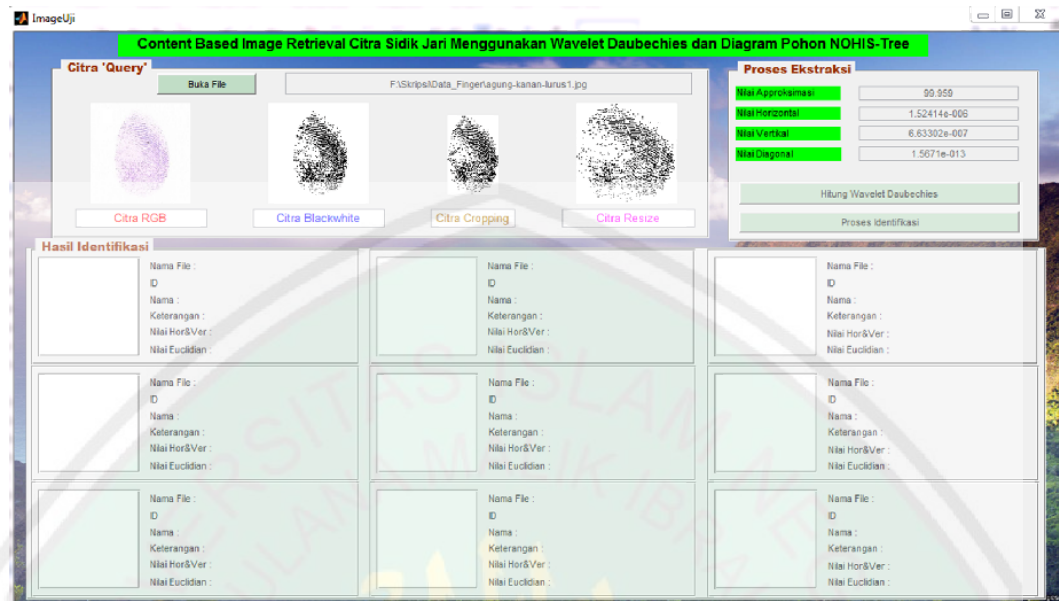
```

Gambar 3.16 Source code untuk transformasi Wavelet Daubechies beserta perhitungan energi wavelet daubechies

Berikut ini contoh hasil proses ekstraksi fitur pada citra sidik jari input dan pada citra sidik jari *query* :



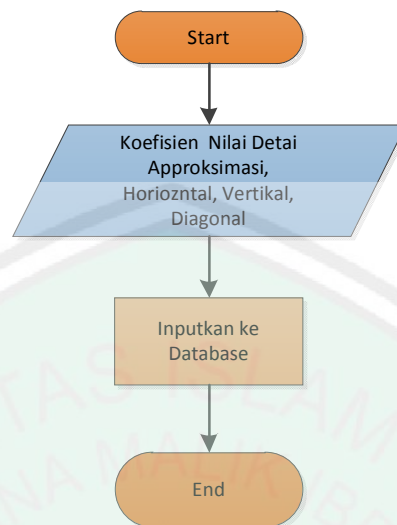
Gambar 3.17 Proses ekstaksi feature menggunakan wavelet daubechies pada akuisisi citra sidik jari input



Gambar 3.18 Ekstraksi feature menggunakan wavelet daubechies pada proses akuisisi citra sidik jari query

3.1.4. Penyimpanan Hasil Ekstraksi Fitur Citra Sidik Jari ke *Database*

Proses penyimpanan hasil ekstraksi fitur adalah bagian keempat dari proses *CBIR* yang dibangun. Berikut ini adalah diagram alur penyimpanan nilai ekstraksi fitur ke dalam database dan proses hasil ekstraksi *feature* dari citra sidik jari yang diinputkan :



Gambar 3.19 Diagram alur penyimpanan hasil ekstraksi fitur ke database



Gambar 3.20 Pengisian identitas dan ekstaksi feature citra input

Pada aplikasi identifikasi citra sidik jari ini, menggunakan XAMPP sebagai server database. Database citra sidik jari dapat di lihat di Lampiran 1. Berikut ini adalah *source code* untuk *button* proses simpan menggunakan *syntax* SQL sebagai berikut :

```

conn = database('db_cbir','root','...'
               ,'com.mysql.jdbc.Driver','jdbc:mysql://localhost/db_cbir');
  
```

```

colnames = {'id', 'nama', 'keterangan','source',...
            'aproksimasi', 'horizontal', 'vertikal','diagonal'};
exdata = { id, nama, keterangan, source ...
          apr_total, hor_total, ver_total, diag_total};
fastinsert(conn, 'data', colnames, exdata);
id

```

Gambar 3.21 Source code untuk button proses simpan menggunakan syntax SQL

Database diberi nama **db_cbir** berisikan dua tabel yaitu tabel data dan tabel temporari. Tabel data adalah tabel yang berisikan data citra sidik jari beserta identitas pemilik sidik jari. Adapun *field* dari tabel data adalah id, nama, keterangan, source, aproksimasi, horizontal, vertikal, dan diagonal. Dan tabel temporari berisikan data ekstraksi yaitu aproksimasi, horizontal, vertikal, dan diagonal. Tabel temporari berfungsi sebagai *converter* data ekstraksi citra sidik jari. Proses *converter* ini berfungsi agar data ekstraksi input dan data ekstraksi database memiliki tipe data dan ukuran (*length*) yang sama. Tabel 3.3 menunjukkan struktur dari tabel data :

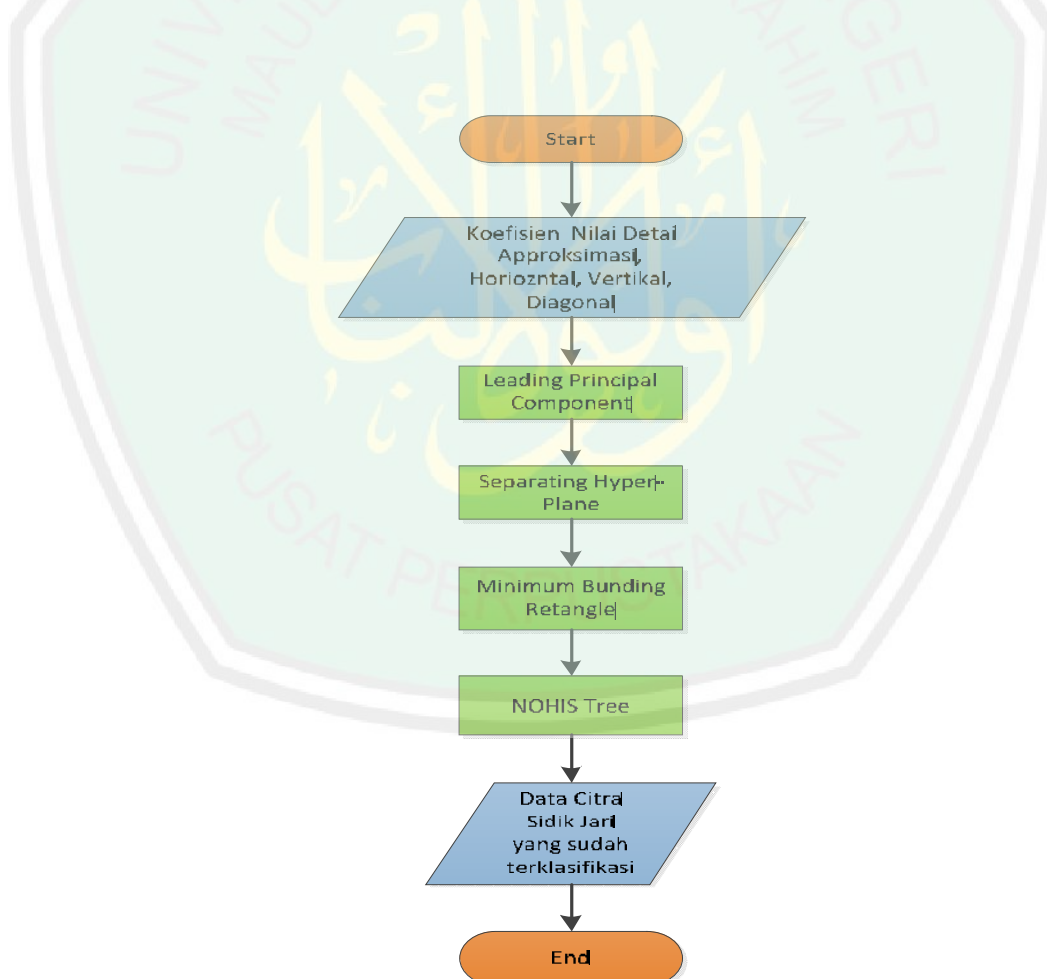
Tabel 3.3 Struktur tabel Data

Field	Tipe
ID	int (11)
Nama	int (255)
Keterangan	varchar (255)
Source	text
aproksimasi	double
horizontal	double
vertikal	double
diagonal	double

3.1.5. Teknik *Indexing* menggunakan Diagram *NOHIS-Tree*

Sistem *Content Based Image Retrieval* yang menggunakan pengindeksan yang efisien dapat melakukan dan mempercepat pencarian dalam database besar. Dalam penelitian ini disajikan indeks hierarkis berdasarkan diagram *NOHIS-tree* (*Non Overlapping Hierarchical Index Structure* atau bebas tumpang tindih indeks struktur hirarkis) ketika berskala ke database yang besar. Penelitian ini juga menghadirkan sebuah studi dari pengaruh pengelompokan pada waktu pencarian.

Berikut ini adalah diagram alur untuk Teknik Pengindeksan dengan menggunakan Diagram *NOHIS-Tree* :

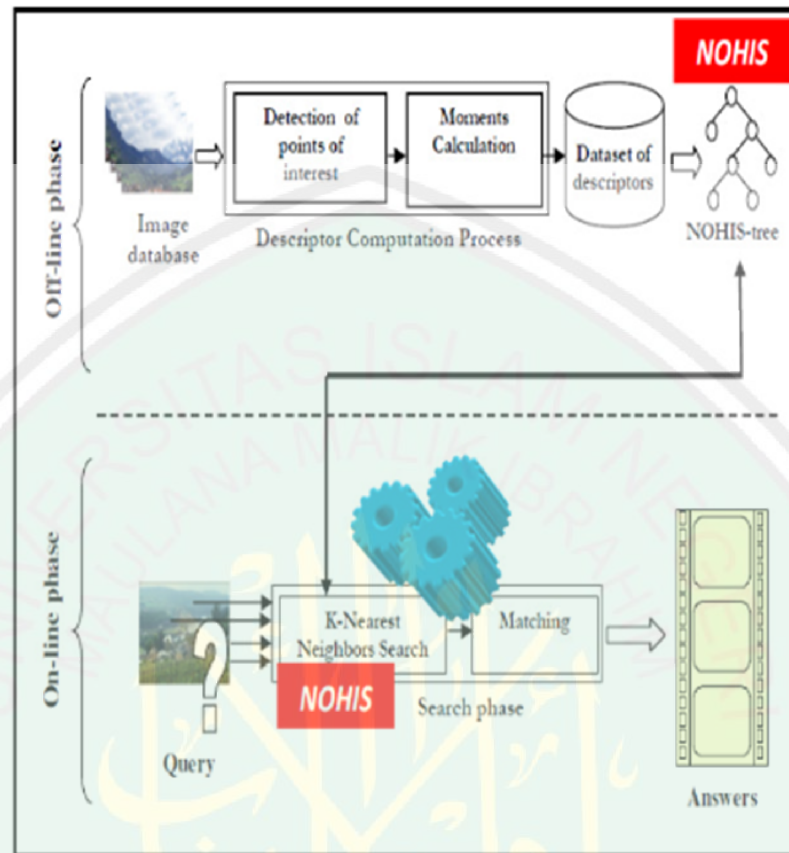


Gambar 3.22 Diagram Alur Teknik Pengindeksan dengan menggunakan Diagram *NOHIS-Tree*

Tree diagram (diagram pohon) adalah sebuah tipe diagram struktur data yang tidak linear/non linear yang digunakan terutama untuk merepresentasikan hubungan data yang bersifat hirarkis antara elemen-elemennya.

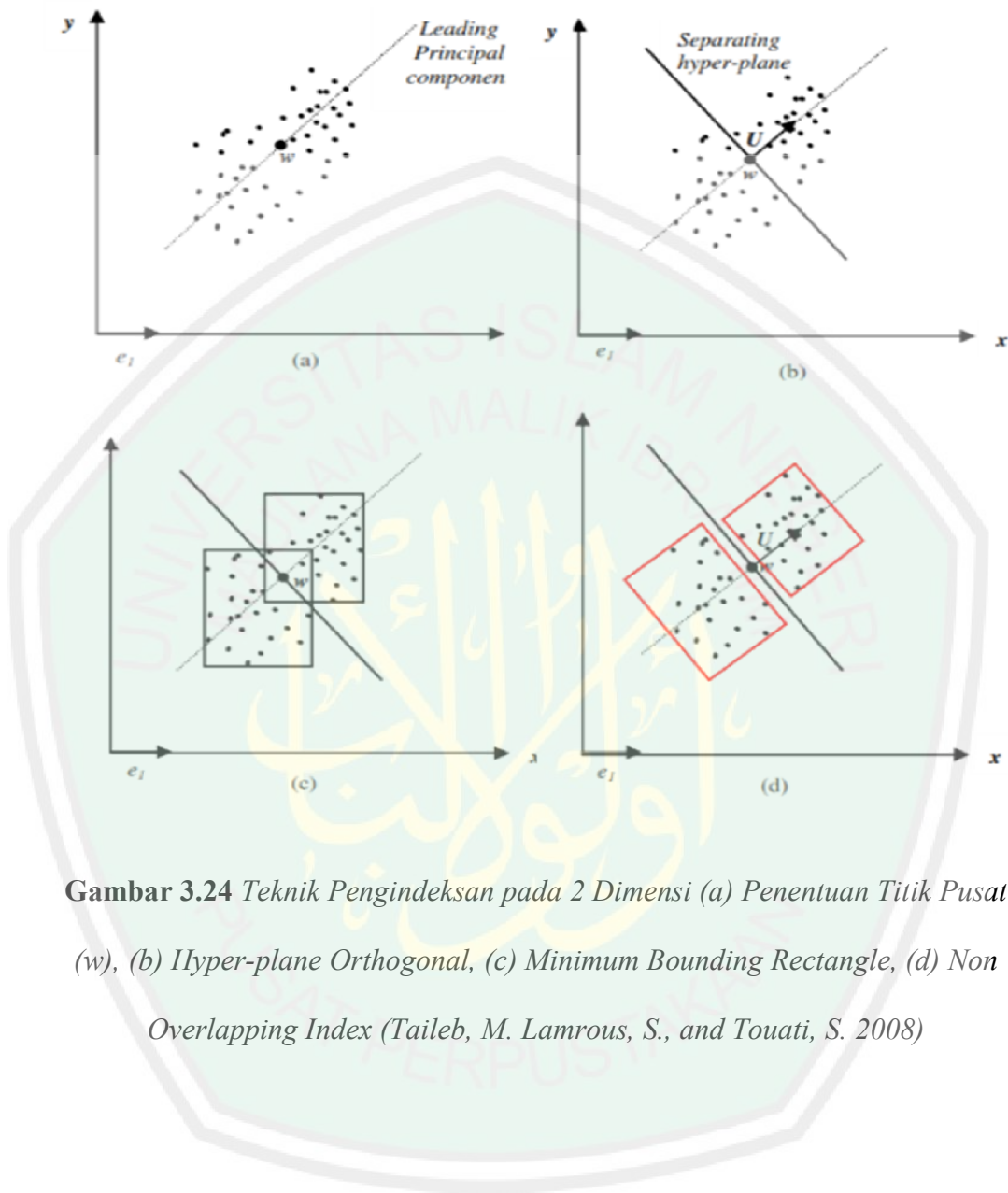
Sistem *CBIR* memulihkan banyak gambar yang serupa ke sebuah *query* dalam sebuah *database image*. Proses ini membutuhkan pendeskripsian konten visual dari gambar secara otomatis dan mewakili setiap gambar dengan satu set vektor multidimensional yang disebut *descriptor*. Dalam sebuah sistem *CBIR* yang disebut *NOHIS-Search* dideskripsikan dengan 2 fase (Taileb, 2013).

Arsitektur implementasi sistem *CBIR*, *NOHIS-Search*, ditunjukkan dalam gambar. Utamanya, ini terdiri dari 2 fase; fase offline dan fase online. pada fase off-line, deskriptor diekstrak dari masing2 gambar di database, dan kemudian deskriptor di-indeks menggunakan metode *high-dimentional indexing NOHIS-tree*. Fase online menangani proses *query* database gambar; deskriptor dari *query* diekstrak dan dikomparasi/dibandingkan dengan gambar lain di database. Gambar 3.23 yang secara visual mirip dengan *query* ditampilkan.

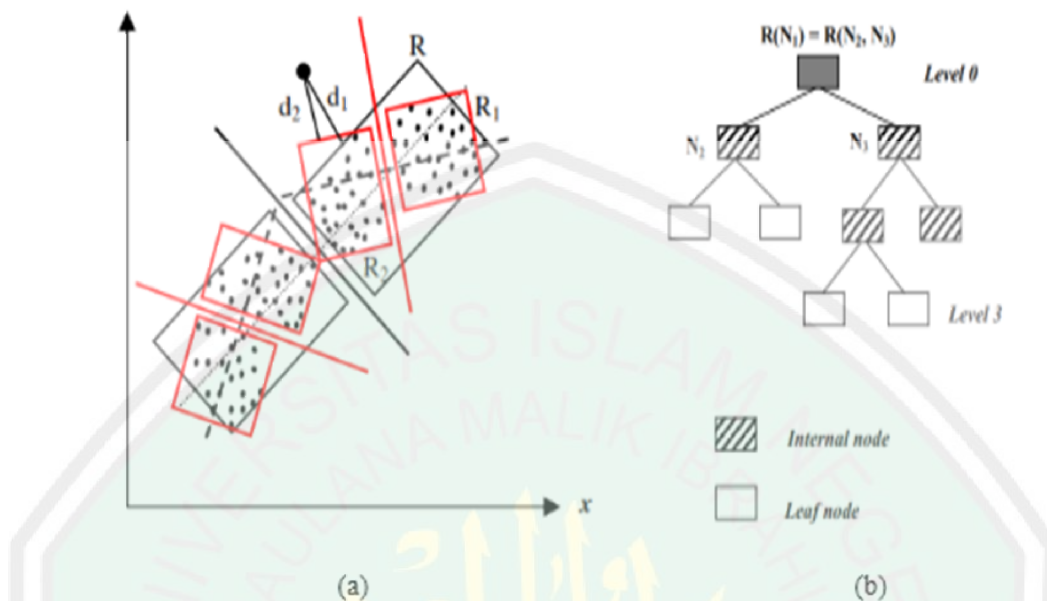


Gambar 3.23 *Arsitektur dari Nohis Search (Taileb, 2013)*

Teknik pengindeks-an multi-dimensi yang ada dapat dibagi dalam dua kelompok sesuai dengan strategi pembagian, data-partisi dan ruang-partisi berbasis struktur index. Ketika pencarian *neraest neighbors* diterapkan pada sebuah indeks data-partisi, cluster tambahan ditinjau karena tumpang tindih antara bentuk bounding (bola atau segi-4). Dalam kasus indeks ruang-partisi, konsultasi beberapa populasi atau cluster kosong adalah sangat mungkin. Dengan menggunakan NOHIS-tree, tumpang tindih dihindari dan kualitas cluster dipertahankan.



Gambar 3.24 Teknik Pengindeksan pada 2 Dimensi (a) Penentuan Titik Pusat (w), (b) Hyper-plane Orthogonal, (c) Minimum Bounding Rectangle, (d) Non Overlapping Index (Taileb, M. Lamrous, S., and Touati, S. 2008)



Gambar 3.25 Teknik Klasifikasi Ekstraksi Fitur pada NOHIS-Tree (a) Rectangle with the childrends, (b) NOHIS-tree (Taileb, M. Lamrous, S., and Touati, S. 2008)

Algoritma pengindeks-an NOHIS:

1. Seluruh set deskriptor diekstrak dari database gambar yang merupakan cluster awal; cluster ini dibagi menjadi dua sub-cluster menggunakan hirarki algoritma pengelompokan PDDP. Masing-masing dari dua sub-cluster dibagi menjadi dua partisi rekursif. Hasil pembagian rekursif adalah sebuah struktur hirarkis dari cluster yang disusun menjadi sebuah pohon biner (*binary tree*).
2. Descriptor dari masing2 sub-cluster yang diperoleh digabungkan oleh 'hyper-rectangles' diarahkan sesuai dengan komponen utama yang terpenting untuk memastikan non-overlapping antara 2 bentuk bounding.

Seluruh set descriptor direpresentasikan oleh sebuah matrix $n \times m$, $M = (d_1, d_2, \dots, d_m)$ dimana masing-masing d_i adalah sebuah descriptor, m adalah ukuran dari descriptor dan n adalah dimensinya. COV, diberikan (1), adalah matrix kovarian dan U komponen utama pertama. Partisi dibuat oleh ‘*hyper-plane orthogonal*’ menjadi komponen utama terpenting dan melewati massa w dari cluster (figure 2.b). Arah utama adalah eigenvector yang terkait dengan eigenvalue dari matrix kovarian.

Bentuk bounding yang digunakan untuk membungkus/menyelimuti descriptor cluster adalah ‘hyper-rectangles’ (dimulai dari 3D). untuk memperjelas kita akan menggunakan term MBR (minimum bounding rectangle), untuk bounding persegi minimum, merujuk pada hyper-rectangle. Figure 2.c menunjukkan kasus ketika MBRs ada di tanda referensi asal; dimana koordinat dari vector disajikan. Jelas bahwa terdapat overlap (tumpang tindih) antara MBRs dan hasil/akibatnya, di sebuah pencarian ‘nearest neighbors’, tambahan cluster akan di’kunjungi’ tanpa meningkatkan hasilnya. Untuk menghindari overlap, algoritma pengindeks-an NOHIS mengarahkan MBRs sesuai dengan komponen utama terpenting. Dalam kasus ini, sebuah perubahan tanda referensi (reference mark) adalah penting untuk menghitung koordinat descriptor dalam tanda referensi (reference mark) baru.

Tujuannya adalah untuk membangun sebuah basis ortonormal $B' = \{u_1, u_2, \dots, u_n\}$ dimana sebuah vector sama dengan U ($u_1 = U$), basis semacam itu dapat diperoleh dengan mentransformasi/mengubah B oleh sebuah ‘orthogonal isomorphism’, Kita definisikan S sebagai simetri ortogonal terhadap H .

Perhatikan bahwa nilai minimum dan maksimum dari formula ini diambil baris demi baris, sehingga $S = (s_1, \dots, s_i, \dots, s_n)$ et $T = (t_1, \dots, t_i, \dots, t_n)$ dimana s_i (resp. t_i) adalah minimum (resp. maksimum) dari komponen ke- i dari vektor yang dipertimbangkan.

Hasil akhir adalah sebuah pohon biner yang tidak seimbang yang disebut *NOHIS-tree*. Dalam sebuah internal node (bukan ‘daun’) dari *NOHIS-tree*, informasi berikut ini disimpan: S_R , T_R , S_L , T_L dan vektor V yang umum diberikan oleh (persamaan 3). Sebuah ‘leaf node’ mengandung deskriptor. ‘Daun’ mewakili cluster yang diperoleh (Taileb, Lamrous, and Touati, 2008).

Dalam penelitian ini implementasi untuk diagram *NOHIS-Tree* didasarkan hanya pada 4 langkah utama fase offline, , yaitu *Leading Principal Component* (penentuan titik pusat), *Separating Hyper-Plane Orthogonal*, *Minimum Bounding Rectangle*, dan *Non-Overlapping Index*. Sedangkan *descriptor* digunakan dari hasil ekstraksi fitur *wavelet daubechies* sehingga bisa diperoleh pengklasifikasian atau pengelompokan database citra sidik jari untuk memudahkan pencarian. *Nearest Neighbour* pada dasar teori dari penelitian sebelumnya digantikan oleh *Euclidian Distance* untuk membandingkan antara citra *query* dengan citra pada *database* yang sudah berbentuk *cluster* atau terbagi menurut diagram *NOHIS-Tree* menjadi 4 kuadran utama sebagai *root*-nya dan akan dibagi lagi menjadi sub-percabangan berikutnya sampai ke percabangan terkecil.

3.1.6. Proses Identifikasi atau *Similaraty Search* menggunakan *NOHIS-Search*

Pada penelitian ini, proses *similaraty search* atau *matching* menggunakan diagram *NOHIS-tree* dengan menyeleksi data citra sidik jari yang ada di database yang memiliki nilai *feature* horizontal dan vertikal dengan kemiripan terdekat dengan citra sidik jari *query*. Perbandingan dilanjutkan dengan perhitungan jarak terdekat (Jarak Euclidean). Nilai jarak Euclidean yang mendekati nilai nol, akan menunjuk pada data tertentu. Data sidik jari pada database yang memiliki nilai fitur termirip dengan input akan memiliki nilai jarak Euclidean yang paling mendekati nol, atau dengan kata lain paling kecil. Rumus untuk menentukan jarak euclidian adalah :

$$d(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_n - v_n)^2}$$

dengan :

$d(\vec{u}, \vec{v})$ = jarak Euclidean

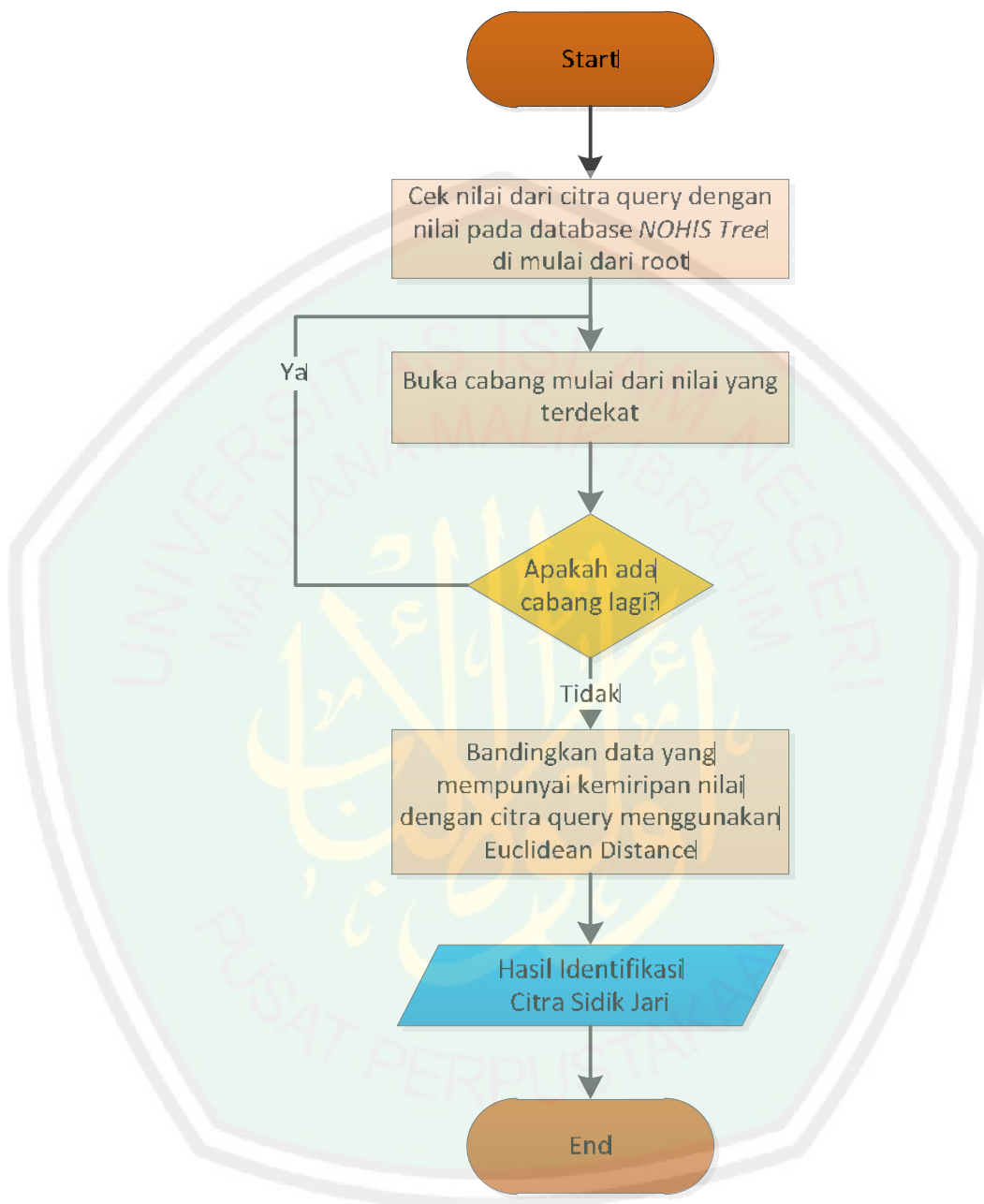
$\vec{u}$ = nilai fitur horizontal

$\vec{v}$ = nilai fitur vertikal

n = banyaknya data

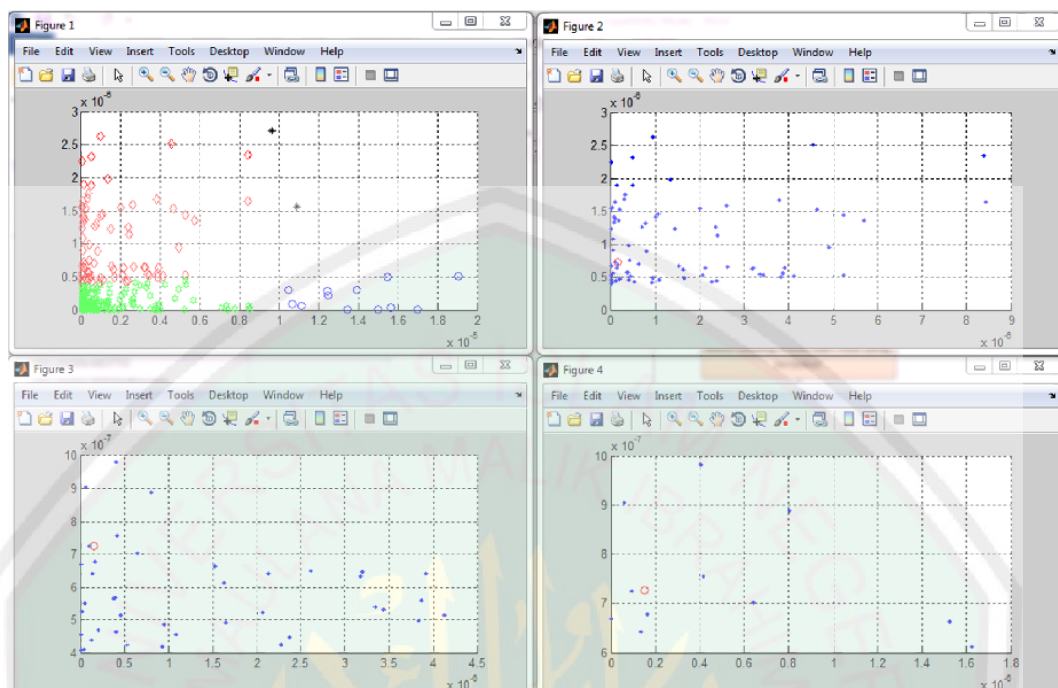
Pengenalan di peroleh dengan menghitung jarak terdekat, yaitu jarak nilai Euclidean yang paling kecil (Hendarko, 2010). Semakin kecil skor $d(\vec{u}, \vec{v})$ maka semakin mirip kedua vektor fitur yang dicocokkan. Sebaliknya semakin besar skor $d(\vec{u}, \vec{v})$ maka semakin berbeda kedua vektor fitur.

Proses pencarian pada indeks diagram *NOHIS-Tree* dimulai dari 4 Kuadran utama yaitu Kuadran I, Kuadran II, Kuadran III, dan Kuadran IV. Selanjutnya aplikasi akan langsung menuju ke salah satu kuadran tempat nilai feature dengan kemiripan terdekat tersebut berada. Apabila data bisa diklasifikasikan lagi dengan metode diagram *NOHIS-tree* maka sistem akan membaginya menjadi 4 sub-kuadran lagi dan proses ini akan berlangsung terus menerus sampai ke percabangan yang lebih spesifik atau percabangan anak yang terkecil. Berikut ini adalah diagram alur proses *Similarity Search* dengan menggunakan *NOHIS Search* :



Gambar 3.26 Diagram Alur Proses Similarity Search dengan NOHIS Search

Berikut ini menunjukkan contoh proses *similarity search* menggunakan teknik pengindeksan diagram *NOHIS-tree*.



Gambar 3.27 Contoh Proses *Similarity Search* pada diagram *NOHIS-tree* level 4

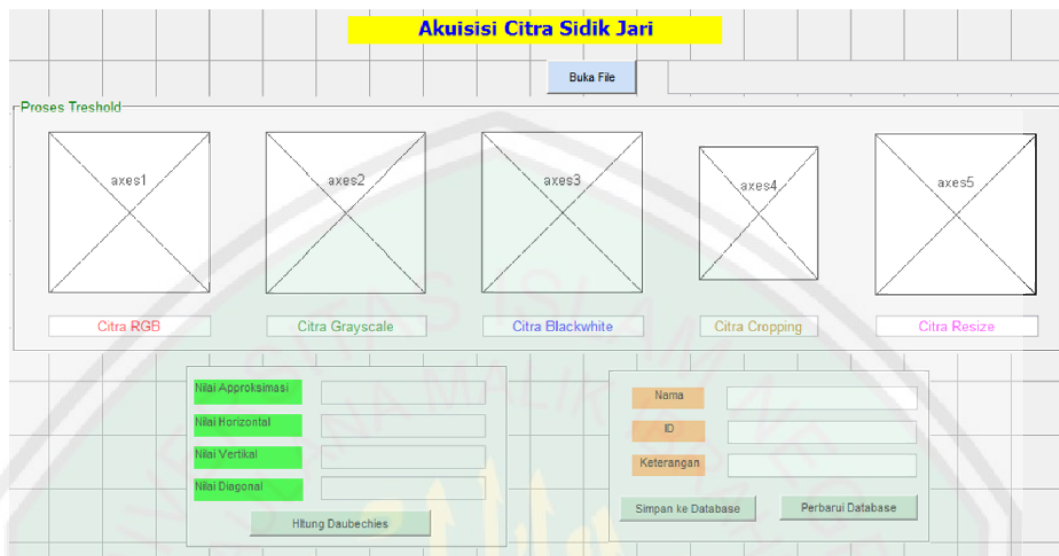
Proses *Similarity Search* dengan menggunakan diagram *NOHIS-tree* dan proses pengurutan nilai *Euclidean* dari terkecil hingga terbesar. Aplikasi kemudian mencetak sembilan nilai *Euclidean* terkecil dari diagram *NOHIS-Tree* tersebut sebagai output program dan di anggap sebagai hasil identifikasi.

Listing *Source Code* untuk proses *Similarity Search* pada aplikasi *CBIR* Citra Sidik Jari menggunakan *NOHIS-Search* terdapat pada lampiran 2.

3.2 Desain GUI (*Graphical User Interface*)

Untuk mempermudah pengguna, maka perlu dibuat tampilan *GUI* atau antarmuka pengguna grafis (*interface*). Berikut ini ditampilkan rancangan antarmuka untuk menginputkan citra sidik jari ke database dan rancangan antarmuka aplikasi *content based image retrieval* Citra Sidik Jari.

a. **Proses Input Sidik Jari**



Gambar 3.28 Antarmuka input citra sidik jari

Dalam halaman input citra sidik jari terdapat beberapa tombol antara lain :

1. **Buka File**, tombol untuk membuka file citra yang akan diinputkan kedalam Aplikasi.
2. **Hitung Daubechies**, tombol untuk memproses ekstraksi citra sidik jari yang telah diinputkan kedalam Aplikasi dengan metode Transformasi Wavelet.
3. **Simpan ke Database**, tombol untuk menyimpan nilai *feature* citra ke dalam database.
4. **Perbarui Database**, tombol untuk memperbaharui data nilai *feature* ke dalam database.

Setelah pengguna memberikan inputan pada Aplikasi dengan menekan tombol “Buka File”, citra sidik jari yang telah dipilih akan tampil pada panel ‘Citra Input’. Alamat file citra akan di cetak pada *textfield* di sebelah tombol

‘Buka File’. Selanjutnya citra input di ubah ke skala *grayscale* kemudian dikonversi menjadi citra hitam-putih. Citra *blackwhite* atau hitam putih ini kemudian akan dilakukan proses *Region of Interest* sehingga yang tampil hanya nilai piksel hitamnya saja atau Citra sidik jarinya saja dan membuang Citra putih atau noise disekitarnya. Citra hasil *ROI* ini kemudian akan di *resize* kembali menjadi Citra sidik jari yang berukuran semula yaitu 512x512 piksel. Pengguna menekan tombol “Hitung Daubechies”. Aplikasi akan mengekstraksi citra input menjadi empat nilai yaitu nilai aproksimasi, nilai horizontal, nilai vertikal, dan nilai diagonal. Terdapat tiga *field* yang harus pengguna isikan yang di tampilkan dengan *field* Nama, ID dan Keterangan. Setelah pengguna mengisikan identitas pemilik sidik jari. Langkah selanjutnya adalah menyimpan nilai *feature* yang telah dihasilkan ke dalam database. Berikut ini adalah tampilan aplikasi untuk proses input citra sidik jari :



Gambar 3.29 Tampilan proses input citra sidik jari

Dalam halaman input citra sidik jari terdapat beberapa tombol antara lain 'Buka File', 'Hitung Daubechies', Simpan ke Database dan 'Perbarui Database'. Ketika pengguna menekan tombol 'Buka File', maka akan menuju *open dialog* untuk memilih file citra yang akan diinputkan. Citra input yang diinputkan akan diletakkan ke panel 'Citra Input' kemudian akan di-*threshold* oleh program, dan kemudian hasil citra *threshold* akan diletakkan ke panel 'Citra Threshold'.



Gambar 3.30 Proses buka file dan thresholding pada proses input

Setelah itu, pengguna mengisi identitas pemilik citra sidik jari berupa Nama, ID, dan Keterangan. Nama yaitu nama lengkap identitas pemilik sidik jari. ID terdiri atas 9 digit yang teridiri dari satu digit pertama adalah kode dari jempol kanan (1) atau kiri (2), dua digit urutan citra sidik jari (01-10), dan enam digit terakhir adalah NIM dari koresponden. Setelah itu pengguna menekan tombol 'Hitung Daubechies'. Dengan menekan tombol ini, program akan mengekstraksi

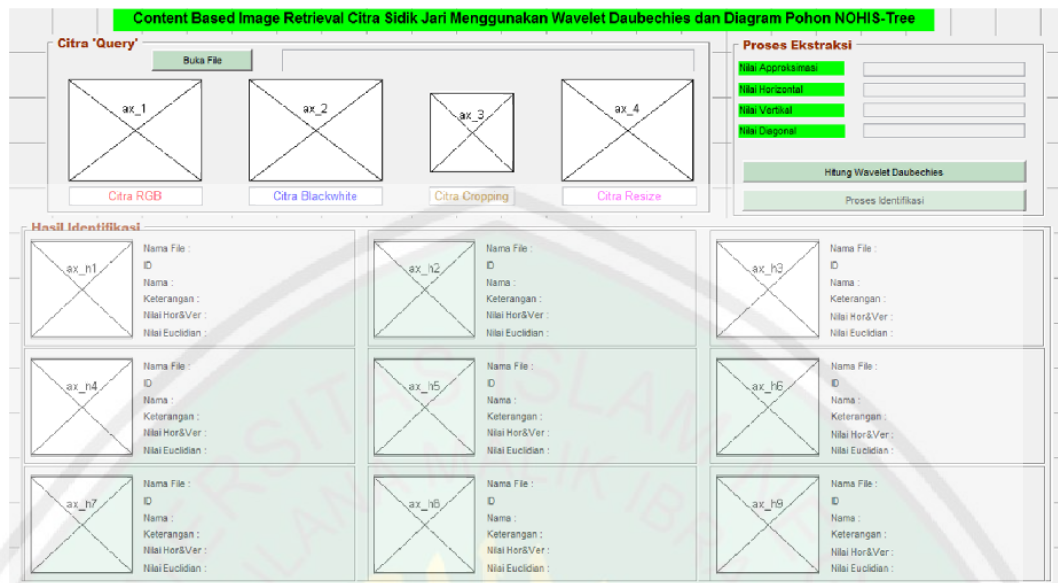
feature dari citra sidik jari yang diinputkan, yang ditunjukkan pada gambar berikut:



Gambar 3.31 Pengisian identitas dan ekstraksi *feature* citra input

Proses terakhir adalah menyimpan hasil citra sidik jari yang telah di ekstraksi ke dalam database dengan menekan tombol 'Simpan ke Database'.

b. Proses Identifikasi *Content Based Image Retrieval* Citra Sidik Jari



Gambar 3.32 Antarmuka proses identifikasi *content based image retrieval* citra sidik jari

Terdapat empat proses yang ditunjukkan pada gambar 3.19, yaitu proses akuisisi citra, *preprocessing*, *main processing* dan output identifikasi. Akuisisi citra yaitu pengambilan citra dari drive computer. Citra yang diinputkan akan diletakkan ax_1, kemudian citra input di konversi ke Citra *blackwhite* dan hasilnya akan ditampilkan pada ax_2, kemudian Citra akan di-ROI dan hasilnya akan ditampilkan pada ax_3, kemudian Citra akan di-*resize* dan hasilnya akan ditampilkan pada ax_4. Setelah citra mengalami proses *thresholding*, maka selanjutnya nilai *feature* Citra dapat dihitung menggunakan *wavelet daubechies* dan nilai energi setiap detail akan ditampilkan pada *text field* yang tersedia. Selanjutnya proses identifikasi dapat dijalankan dan Aplikasi akan mencari/mencocokkan nilai *feature* antara citra *query* dan citra pada database berdasarkan teknik pengindeksan diagram *nohis-tree* yang telah diterapkan dan gambaran proses pencariannya pada kuadran induk ke sub-sub kuadran lainnya

akan ditampilkan melalui figure yang muncul otomatis sesuai banyaknya percabangan. Hasil akhir pencarian akan menampilkan 9 citra sidik jari dengan nilai *Euclidean* terkecil dan ditampilkan pada masing-masing axes.

Dalam halaman proses identifikasi citra sidik jari, terdapat tiga tombol yaitu 'Buka File', 'Hitung Wavelet Daubechies', 'Proses Identifikasi *NOHIS*' dan 'Tes Uji Akurasi'. Ketika pengguna menekan tombol 'Buka File', maka akan keluar *open dialog* kepada pengguna untuk memilih file citra yang akan diidentifikasi. Setelah itu citra sidik jari akan di-*threshold*. Citra input dan citra yang di-*threshold* akan ditampilkan pada panel 'Citra Query'. Berikut ini adalah tampilan setelah pengguna menekan tombol 'Buka File'.



Gambar 3.33 Proses buka file dan threshold pada proses identifikasi

Proses setelah pengguna membuka file citra sidik jari, adalah menghitung nilai energi atau ekstraksi *feature* citra sidik jari dengan menekan tombol 'Hitung

Wavelet Daubechies'. Berikut ini adalah implementasi dari transformasi wavelet daubechies, Gambar 3.34 menunjukkan perhitungan ekstraksi *feature*.

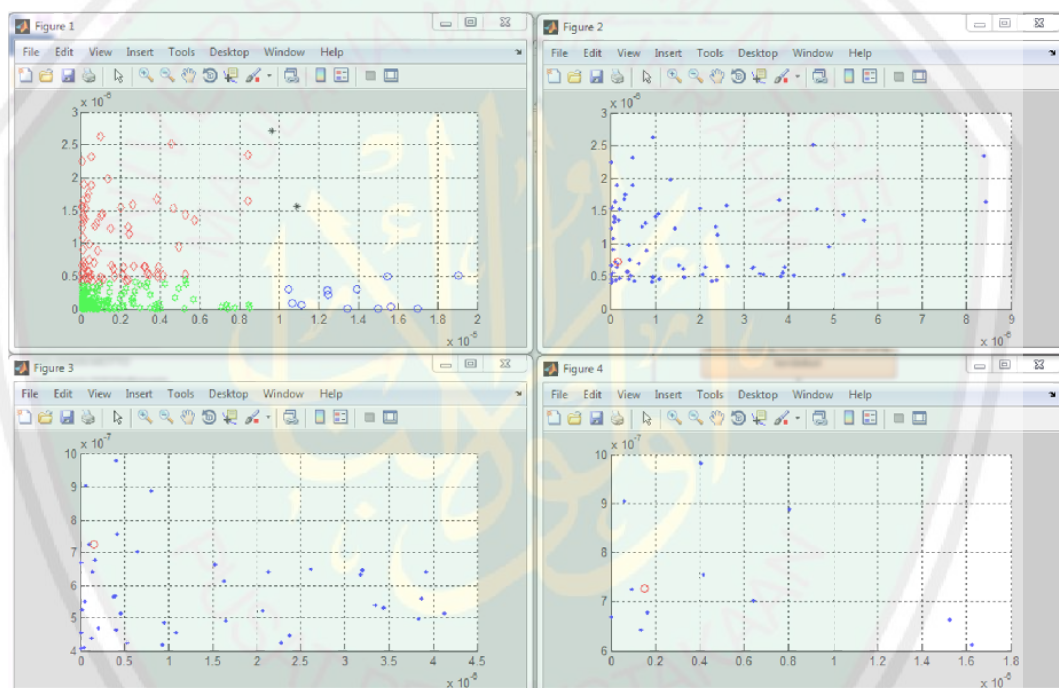


Gambar 3.34 Proses ekstraksi *feature* citra sidik jari

Proses selanjutnya adalah mengidentifikasi citra sidik jari dengan menekan tombol 'Proses Identifikasi'. Pada proses ini, citra sidik jari yang telah diinputkan akan mengalami proses pencarian menggunakan teknik pengindeksan NOHIS-tree dan proses *matching* ini menggunakan nilai ekstraksi *feature* dan di hitung nilai *Euclidean* antara citra *query* dengan citra database. Berikut ini adalah *source code* klasifikasi kuadran atau percabangan *tree* dengan menggunakan teknik pengindeksan diagram *NOHIS-tree*:

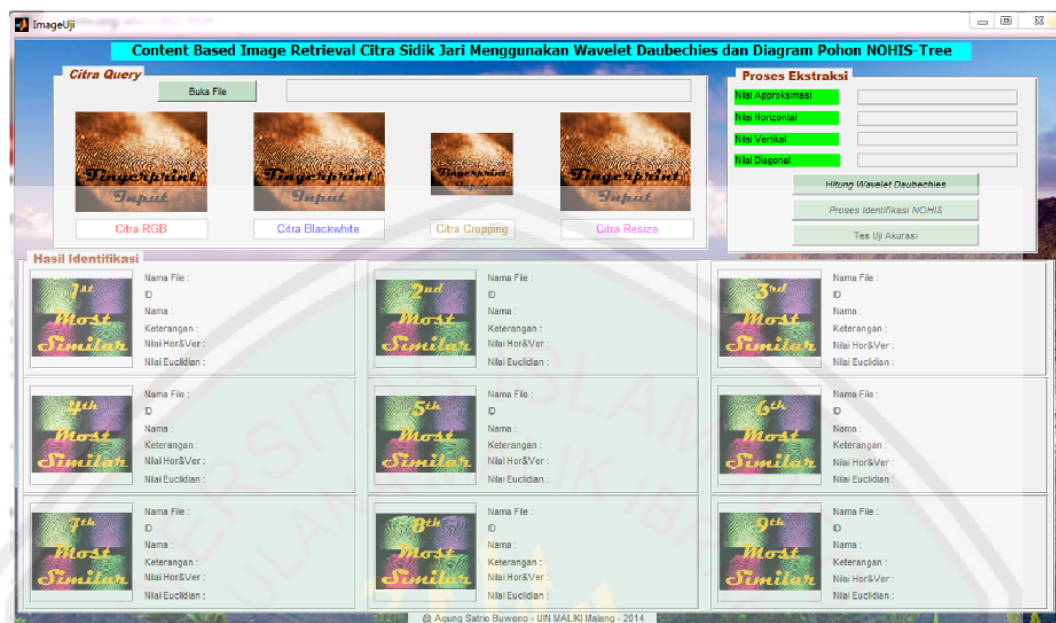
Pada penelitian ini, proses *Similarity Search* menggunakan diagram *NOHIS-tree* akan menyeleksi data yang ada di database yang memiliki nilai *feature* dengan kemiripan terdekat dengan Citra *query*. Proses pencarian dimulai dari 4 Kuadran utama yaitu Kuadran I, Kuadran II, Kuadran III, dan Kuadran IV. Selanjutnya Aplikasi akan langsung menuju ke salah satu kuadran tempat nilai

feature dengan kemiripan terdekat tersebut berada. Apabila data bisa diklasifikasikan lagi dengan metode diagram *NOHIS-tree* maka Aplikasi akan membaginya menjadi 4 sub-kuadran lagi dan proses ini akan berlangsung terus menerus sampai ke percabangan yang lebih spesifik atau percabangan anak yang terkecil. Gambar 3.35 menunjukkan proses *Similarity Search* menggunakan teknik pengindeksan diagram *NOHIS-tree*.



Gambar 3.35 Contoh Proses *Similarity Search* pada diagram *NOHIS-tree* level 4

Berikut ini adalah *Similarity Search* dengan menggunakan diagram *NOHIS-tree* dan pengurutan nilai *Euclidean* dari terkecil hingga terbesar dan mencetak sembilan nilai *Euclidean* terkecil dari diagram *NOHIS-Tree* tersebut sebagai output program dan di anggap sebagai hasil identifikasi. Berikut ini adalah tampilan *GUI* dari proses *ranking* identifikasi citra sidik jari :



Gambar 3.36 Tampilan Ranking hasil identifikasi Content Based Image Retrieval citra sidik jari

Listing *Soure Code* untuk menampilkan urutan gambar hasil identifikasi sidik jari teratas dengan menggunakan perbandingan *Euclidian Distance* pada diagram *NOHIS-Tree* terdapat pada lampiran 3.

3.3 Implementasi Aplikasi

Terdapat dua kebutuhan dalam implementasi aplikasi *Content Based Image Retrieval* Citra Sidik Jari menggunakan metode *Wavelet Daubechies* dan diagram *NOHIS-Tree* yaitu kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut penjelasannya :

1. Perangkat Keras (*Hardware*)

Untuk merancang dan membuat aplikasi *Content Based Image Retrieval* Citra Sidik Jari Menggunakan Metode *Wavelet Daubechies* dan Diagram

NOHIS-Tree, penulis menggunakan perangkat komputer dengan spesifikasi: Processor Intel® Core™ i3 CPU 2330M @2.20GHz (4 CPUs) dan RAM 2048MB.

2. Perangkat Lunak (*Software*)

Dalam perancangan dan pembuatan aplikasi, penulis menggunakan beberapa perangkat lunak yaitu :

a. Aplikasi Operasi 7 Ultimate

Sistem operasi windows 7 Ultimate digunakan sebagai susunan arahan yang dapat difahami oleh komputer. Dibuat untuk mengarahkan komputer melaksanakan, mengawal, menjadwalkan, dan menyelaraskan sesuatu operasi komputer.

b. Matlab 7.13.0.564 (R2011b)

Matlab merupakan sebuah lingkungan komputasi numerical dan bahasa pemrograman komputer yang memungkinkan manipulasi matriks, implementasi algoritma, pembuatan antarmuka pengguna dan pengantarmukaan program dengan bahasa lainnya. Matlab digunakan sebagai tool dalam melakukan pemrograman dan pembangunan Aplikasi ini.

c. Microsoft Office 2013

Microsoft office adalah sebuah paket aplikasi yang digunakan untuk pembuatan dan penyimpanan dokumen yang berjalan di bawah system operasi windows. Microsoft office dalam perancangan

Aplikasi digunakan untuk melakukan perancangan dan pembuatan laporan dari penelitian ini.

d. Apache Friends XAMPP (Basis Package) version 1.8.3

XAMPP adalah perangkat lunak yang mendukung banyak sistem operasi. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. XAMPP digunakan sebagai server database citra sidik jari.

e. Adobe Photoshop CS6

Adobe Photoshop CS6 adalah sebuah perangkat lunak yang digunakan untuk proses akuisisi data yaitu untuk *cropping* citra sidik jari yang telah di-*scan*. Citra sidik jari tersebut di potong sesuai dengan kebutuhan penelitian yaitu 512x512 piksel. Serta digunakan untuk menghilangkan noise berupa titik-titik hitam yang ada disekitar sidik jari untuk proses *cropping* Citra *blackwhite* pada matlab.

BAB IV

EKSPERIMEN DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai rangkaian uji coba dan evaluasi terhadap penelitian yang telah dilakukan. Uji coba ditujukan untuk melihat sejauh mana keberhasilan dari implementasi perangkat lunak ini dan evaluasi dilakukan dengan melakukan analisa terhadap hasil dari uji coba dan juga untuk mendapatkan kesimpulan dan saran untuk pengembangan kedepan bagi implementasi aplikasi perangkat lunak ini.

4.1 Langkah-Langkah Eksperimen

Langkah-langkah uji coba pada penelitian ini dapat dijabarkan sebagai berikut :

- a. Akuisisi citra, meliputi pengambilan data citra sidik jari. Data sidik jari yang di ambil adalah sepuluh citra ibu jari kanan dan sepuluh ibu jari kiri. Pengambilan data yaitu dengan cara menempelkan ibu jari yang telah diberikan *stam pad ink* berwarna biru ke kertas template sidik jari
- b. *Scanning*. Kertas template kemudian di-*scanning* menggunakan alat *scanning* yang menghasilkan citra dengan ukuran 4677 x 3400 piksel.
- c. *Cropping*. Citra hasil *scanning* kemudian di-*crop* dengan ukuran 512 x512 piksel untuk masing-masing citra sidik jari dengan menggunakan *software* Adobe Photoshop CS6.

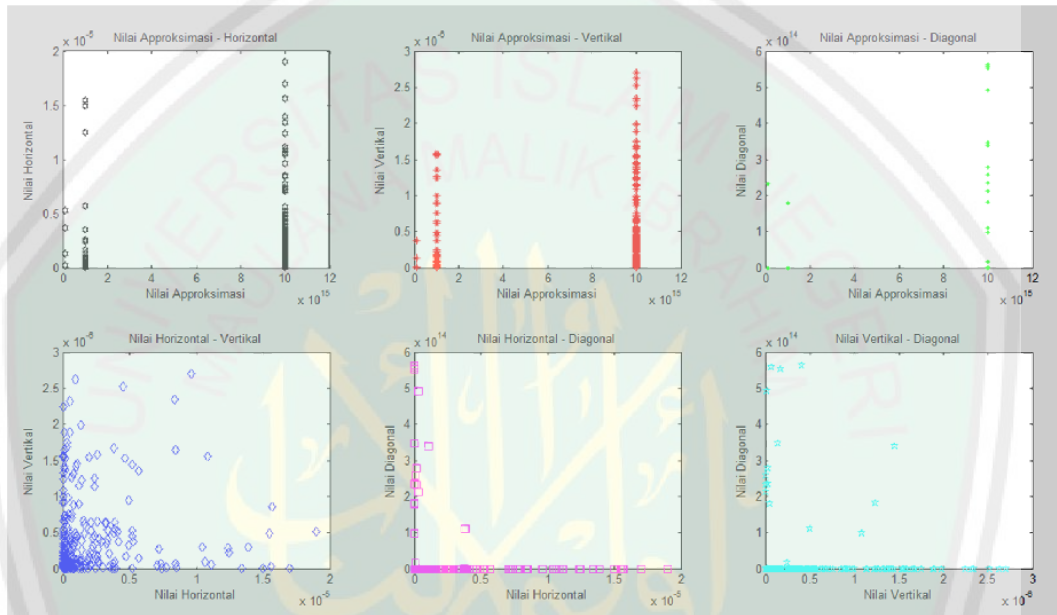
- d. Akuisisi Citra Sidik Jari dan proses *Preprocessing* meliputi proses *threshold*, *cropping* citra *B/W* dan *resize* ke ukuran semula menggunakan MATLAB 2011b.
- e. Simpan dalam database, ekstraksi *feature* menggunakan *wavelet daubechies* kemudian di simpan dalam database **db_cbir** di XAMPP.
- f. Uji Coba. Setelah seluruh hasil ekstraksi fitur citra sidik jari dimasukkan ke database, citra sidik jari di uji coba ke dalam aplikasi yang telah dilakukan penerapan teknik pengindeksan dengan menggunakan diagram *NOHIS-Tree*.
- g. Mendeteksi masuknya data yang sesuai pada hasil uji coba yaitu pada empat macam peringkat : 9 prioritas tertinggi, 6 prioritas tertinggi, 3 prioritas tertinggi, dan 1 prioritas tertinggi.

4.2 Eksperimen atau Uji Coba

Eksperimen mulai dilakukan setelah citra input sidik jari tersimpan seluruhnya dalam database **db_cbir** pada database XAMPP, selanjutnya dilakukan analisa untuk menerapkan teknik *indexing* berdasarkan diagram *NOHIS-Tree* pada hasil ekstraksi fitur citra sidik jari.

Data hasil perhitungan nilai energi yang berupa nilai aproksimasi dan nilai koefisien detail (horizontal, vertikal, diagonal) dari citra sidik jari yang sudah tersimpan pada database akan diteliti persebaran datanya. Persebaran data yang cocok akan digunakan untuk penerapan teknik pengindeksan berdasarkan diagram *NOHIS-Tree* sehingga data dapat dikelompokkan atau diklasifikasi untuk memudahkan pencarian.

Analisa dilakukan dengan menggabungkan 2 nilai koefisien diantara 4 nilai yang terdapat dalam database citra sidik jari yaitu nilai koefisien aproksimasi, koefisien horizontal, koefisien vertikal dan koefisien diagonal. Berikut ini hasil analisisnya :



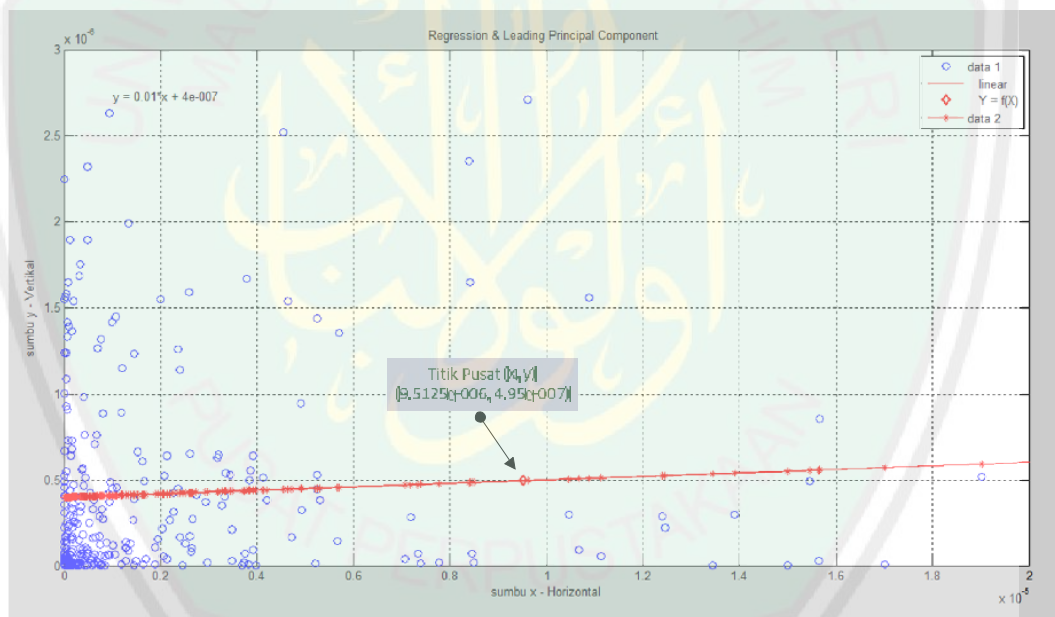
Gambar 4.1 Persebaran Data 2 Nilai Koefisien Aproksimasi dan Koefisien
Detail

Berdasarkan hasil analisa tersebut maka persebaran data yang paling baik untuk penerapan teknik pengindeksan Citra sidik jari 2 dimensi menggunakan *NOHIS-Tree* adalah kombinasi nilai koefisien horizontal dan nilai koefisien vertikal.

Penerapan Metode menggunakan Diagram *NOHIS-Tree* atau Teknik Pengindeksan pada Citra Sidik Jari 2 Dimensi ini meliputi 4 tahap utama, yaitu :

- i. Penentuan Titik Pusat (*Leading Principal Component*)

Penentuan titik pusat dari persebaran data dua nilai yaitu koefisien horizontal dan koefisien vertikal ini pertama-tama menggunakan persamaan regresi linear untuk menemukan pola atau fungsi korelasi dari dua variabel tersebut. Setelah ditemukan persamaan fungsi linearnya maka nilai tengah (median) dari data koefisien horizontal (x) dimasukkan ke dalam fungsi linear tersebut sehingga diperoleh nilai koefisien vertikalnya (y). Titik pusat dari fungsi regresi linear inilah yang kemudian dijadikan sebagai titik pusat (x,y) untuk keseluruhan data. Berikut gambar regresi linearnya :

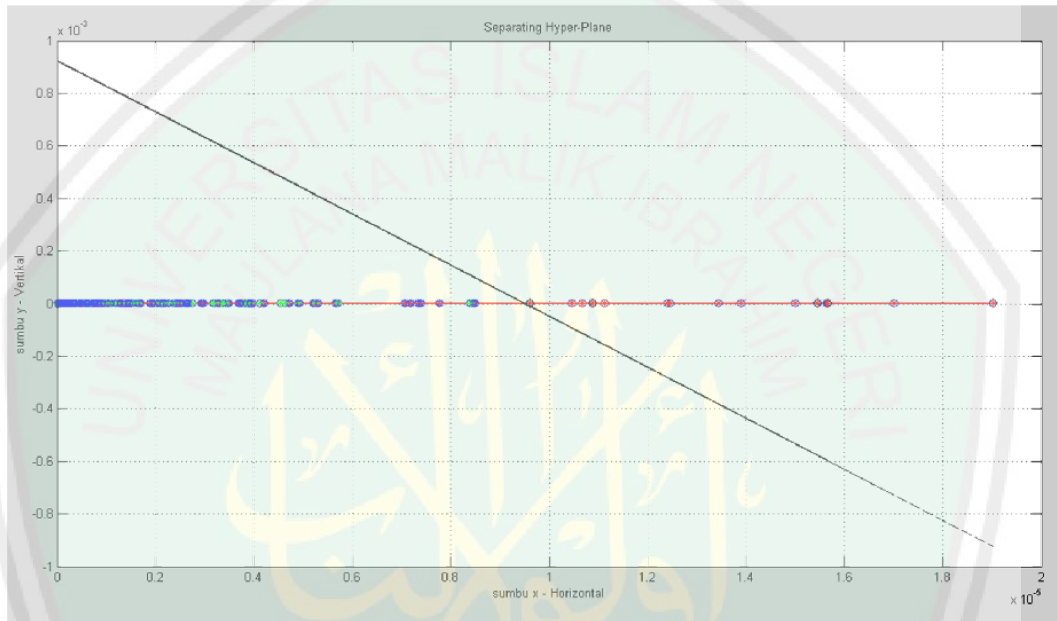


Gambar 4.2 Titik Pusat menggunakan Persamaan Regresi Linear

ii. *Separating Hyper-plane Orthogonal*

Separating Hyper-plane Orthogonal yaitu memisahkan atau membagi dua persebaran data secara tegak lurus (orthogonal) pada garis fungsi regresi linear dan melalui titik pusat. Dalam hal ini kami menggunakan persamaan yang melalui titik (x, y) dan tegak lurus terhadap garis $y = mx + c$. Dimana syarat dua garis

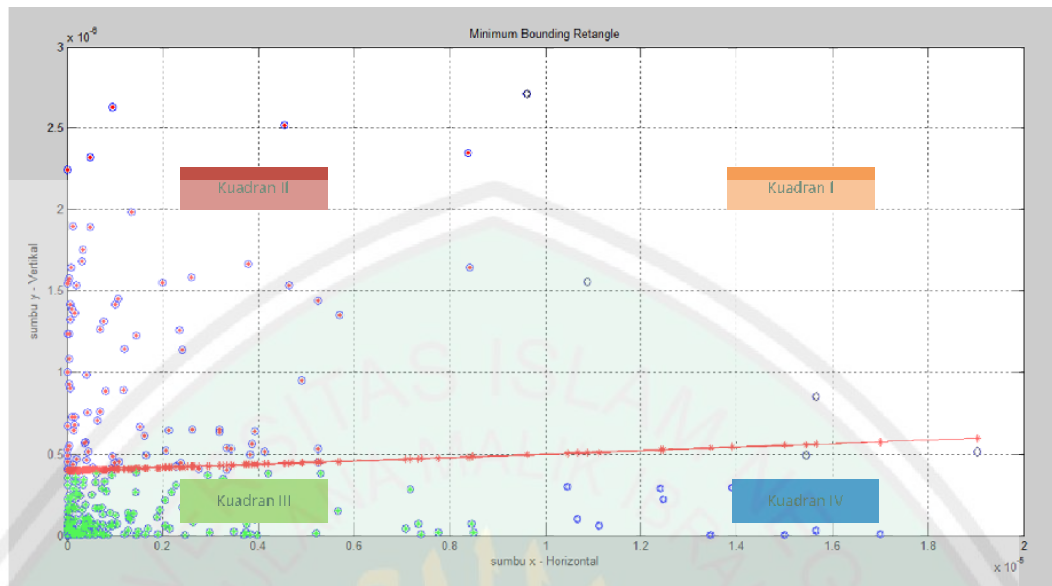
yang tegak lurus adalah hasil kali dua gradiennya adalah -1 ($m_1 \times m_2 = -1$), kemudian m_2 yang telah ditemukan dan melalui satu titik (x_1, y_1) akan didapat persamaannya dengan dimasukkan ke persamaan berikut : $y - y_1 = m_2 (x - x_2)$. Berikut hasil perpotongan dua garis tersebut :



Gambar 4.3 *Separating Hyper-plane Orthogonal*

iii. *Minimum Bounding Rectangle*

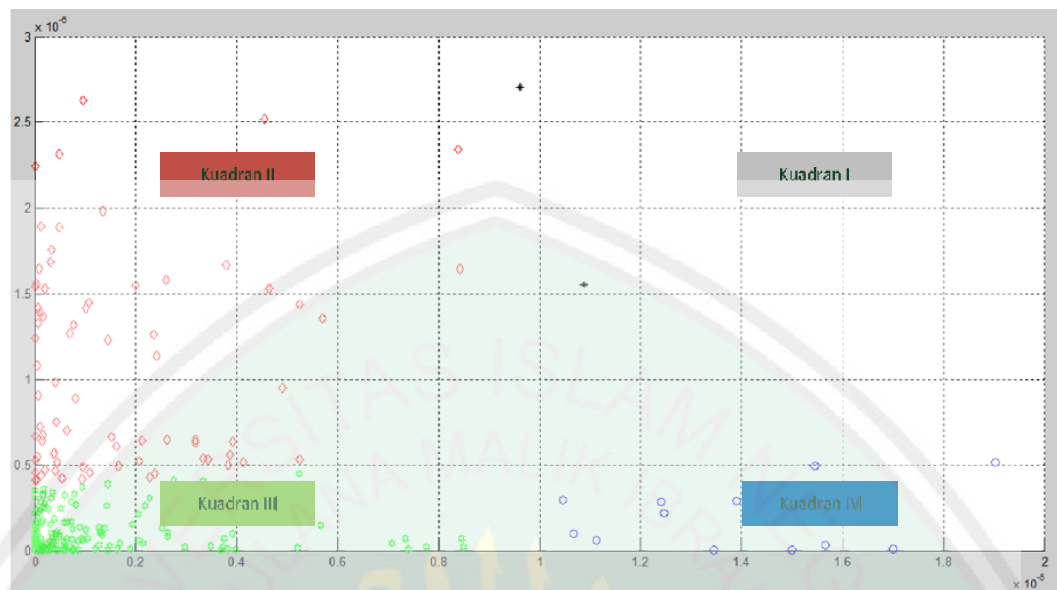
Minimum Bounding Rectangle adalah Sebuah batas minimum yang berbentuk kotak yang berfungsi untuk mengelompokkan data-data sehingga terklasifikasi menjadi bagian-bagian yang lebih spesifik. Dalam penelitian ini kami menerapkan MBR-nya menjadi 4 bagian utama yang dibatasi oleh garis 1 (regresi linear) dan garis ke 2 (*separating hyper-plane orthogonal*) yang kemudian dinamai dengan Kuadran I, Kuadran II, Kuadran III, dan Kuadran IV. Berikut gambaran pembagian kuadran :



Gambar 4.4 *Pembagian Kuadran (Minimum Bounding Rectangle)*

iv. *Non-Overlapping Index*

Pada tahap ini setelah persebaran data dibagi menjadi 4 bagian yaitu kuadran 1,2,3, dan 4 namun ada beberapa data yang menyimpang/tumpang tindih. Sebagai contoh pada gambar 4.4, ada data kuadran III yang masuk ke dalam wilayah kuadran IV sehingga perlu dilakukan pemisahan lagi sehingga data benar-benar berada tepat pada wilayah atau daerahnya masing-masing. Berikut hasil pembagiannya yang benar-benar terpisah (*non-overlapping*) :



Gambar 4.5 *Non-Overlapping Index*

Berikut ini adalah listing kode sumber untuk menampilkan teknik pengindeksan berdasarkan diagram *NOHIS-Tree*:

```
function
[NewK,Kuadran]=metode_nohis_2(Matriks_data,diplot,h_input,v_input)

h_input;
v_input;

baris=size(Matriks_data,1);
kolom=size(Matriks_data,2);

id = Matriks_data(:,1);
horizontal = Matriks_data(:,2);
vertikal = Matriks_data(:,3);

P=polyfit(horizontal,vertikal,1);
%Titik Pusat
xMin=min(horizontal);
xMax=max(horizontal);
xPusat=(xMin+xMax)/2;
yPusat=(P(1))*xPusat+ (P(2));

indexK = 1;
NewK=zeros(1,3);

y_input=P(1)*h_input+P(2);
```

```

m=-1/P(1);
y1_input=m*(h_input-xPusat)+yPusat;

if v_input > y_input && v_input> y1_input
    Kuadran = 1;
end
if v_input > y_input && v_input < y1_input
    Kuadran = 2;
end
if v_input < y_input && v_input < y1_input
    Kuadran = 3;
end
if v_input < y_input && v_input > y1_input
    Kuadran = 4;
end

for i=1:baris
    y=P(1)*horizontal(i)+P(2);
    m=-1/P(1);
    y1=m*(horizontal(i)-xPusat)+yPusat;

    if Kuadran == 1
        if vertikal(i) > y && vertikal(i) > y1
            NewK(indexK,1)=id(i);
            NewK(indexK,2)=horizontal(i);
            NewK(indexK,3)=vertikal(i);
            indexK=indexK+1;
        end
    end

    if Kuadran == 2
        if vertikal(i) > y && vertikal(i) < y1
            NewK(indexK,1)=id(i);
            NewK(indexK,2)=horizontal(i);
            NewK(indexK,3)=vertikal(i);
            indexK=indexK+1;
        end
    end

    if Kuadran == 3
        if vertikal(i) < y && vertikal(i) < y1
            NewK(indexK,1)=id(i);
            NewK(indexK,2)=horizontal(i);
            NewK(indexK,3)=vertikal(i);

            indexK=indexK+1;
        end
    end

    if Kuadran == 4
        if vertikal(i) < y && vertikal(i) > y1
            NewK(indexK,1)=id(i);
            NewK(indexK,2)=horizontal(i);
            NewK(indexK,3)=vertikal(i);
        end
    end
end

```

```

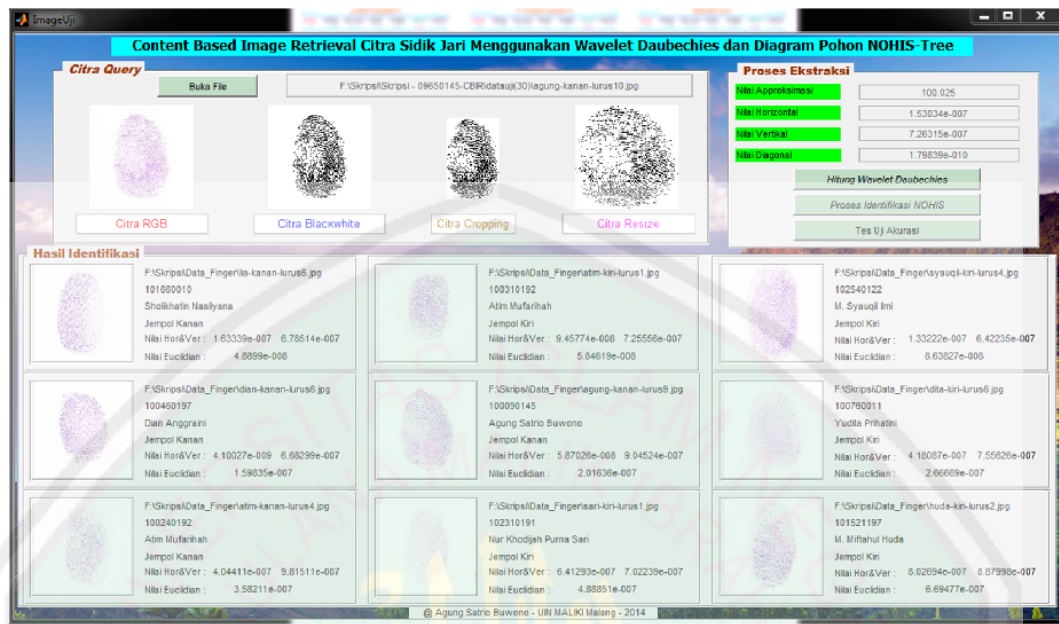
        indexK=indexK+1;
    end
end
end

```

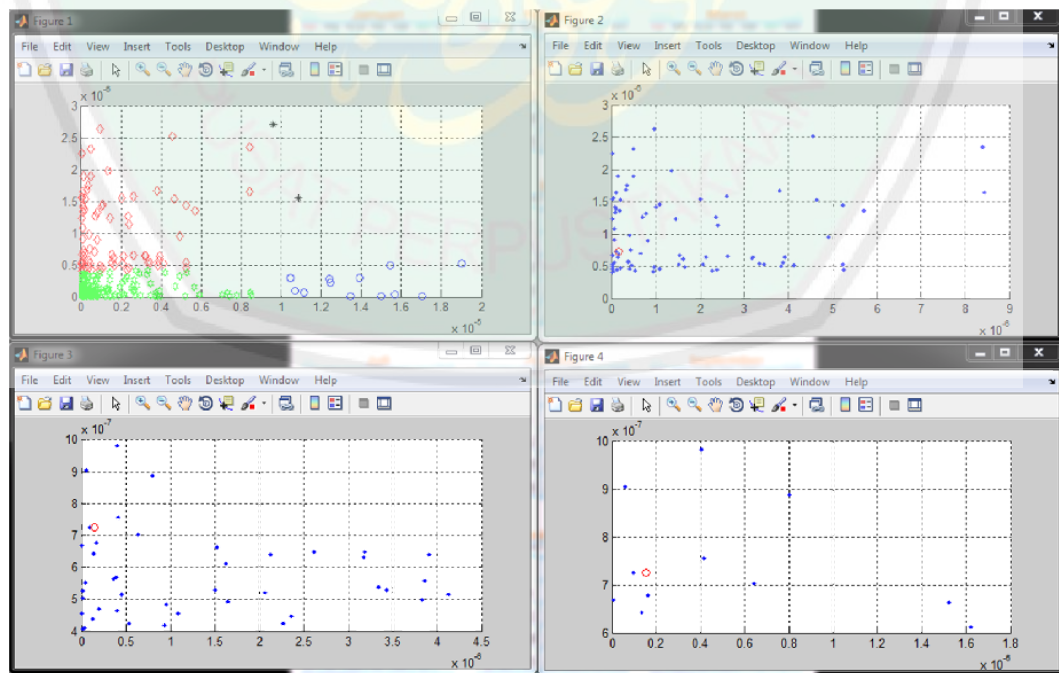
Gambar 4.6 Kode sumber untuk menampilkan teknik pengindeksan berdasarkan diagram *NOHIS-Tree*

Proses pengujian aplikasi dilakukan dengan cara membandingkan hasil identifikasi citra sidik jari yang diperoleh dari proses uji coba. Setelah itu hasil program juga akan dibandingkan dengan identifikasi secara manual.

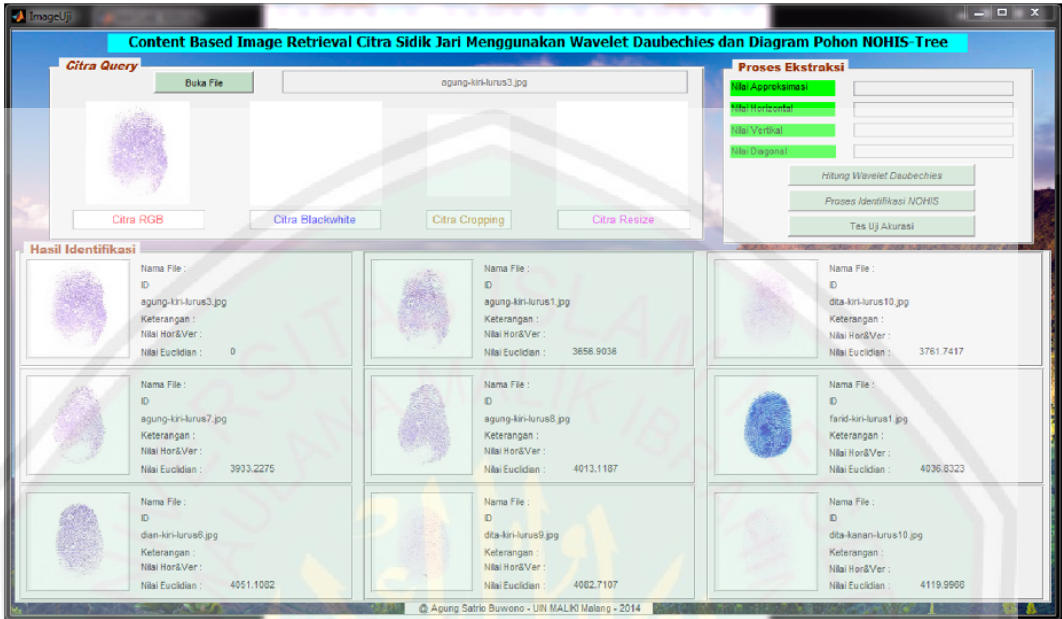
Aplikasi akan mendeteksi identitas dari pemilik citra sidik jari inputan. Proses uji coba akan melakukan pengujian satu persatu terhadap citra sidik jari. Dari hasil identifikasi akan diperoleh jarak *Euclidean* terkecil beserta identitas pemilik sidik jari. Pada uji coba ini digunakan 10 data sampel citra sidik jari kanan dan 10 data sampel citra sidik jari kiri. Gambar 4.7 menunjukkan hasil proses identifikasi *CBIR* citra sidik jari, Gambar 4.8 menunjukkan identifikasi kuadran pada diagram *NOHIS-Tree* dan Gambar 4.9 menunjukkan identifikasi menggunakan *daubechies db4* Matlab dan metode *Brute Force* serta Tabel 4.1 menunjukkan hasil uji coba identifikasi citra sidik jari.



Gambar 4.7 Hasil proses identifikasi CBIR citra sidik jari menggunakan wavelet daubechies dan diagram NOHIS-Tree



Gambar 4.8 Hasil proses identifikasi pada diagram NOHIS-tree citra sidik jari



Gambar 4.9 Hasil proses identifikasi menggunakan daubechies db4 matlab dan metode brute force

Tabel 4.1 Hasil uji citra sidik jari

Data Uji	Hasil 1	Hasil 2	Hasil 3	Hasil 4	Hasil 5	Hasil 6	Hasil 7	Hasil 8	Hasil 9
<i>Agung Kanan</i>	Lia Kanan	Atim Kiri	Syauqil Kiri	Dian Kanan	Agung Kanan	Dita Kiri	Atim Kanan	Sari Kiri	Huda Kiri
<i>Agung Kiri</i>	Sari Kiri	Tino Kiri	Lia Kiri	Agung Kiri	Farid Kanan	Atim Kiri	Tareka Kanan	Sanata Kiri	Syauqil Kiri
<i>Atim Kanan</i>	Dita Kanan	Atim Kanan	Tareka Kanan						
<i>Atim Kiri</i>	Habib Kanan	Huda Kanan	Dita Kiri	Huda Kanan	Tareka Kiri	Sanata Kanan	Atim Kiri	Sanata Kiri	Faizar Kiri
<i>Dian Kanan</i>	Agung Kanan	Atim Kiri	Dian Kanan	Lia Kanan	Syauqil Kiri	Atim Kanan	Dita Kiri	Sari Kiri	Huda Kiri
<i>Dian Kiri</i>	Prima Kanan	Syauqil Kiri	Syauqil Kanan	Sanata Kanan					
<i>Dita Kanan</i>	Huda Kiri	Agung Kanan	Agung Kanan	Sari Kiri	Dita Kiri	Atim Kanan	Lia Kanan	Syauqil Kiri	Atim Kiri
<i>Dita Kiri</i>	Fitriana Kanan	Sari Kanan	Dita Kiri	Dita Kanan	Syauqil Kiri	Sanata Kiri	Lia Kanan	Habib Kiri	Huda Kiri
<i>Faizar Kanan</i>	Faizar Kanan	Agung Kiri	Tino kanan	Tareka Kanan	Prima Kanan	Farid Kanan			
<i>Faizar Kiri</i>	Faizar Kanan	Fitriana Kiri	Dita Kanan	Sanata Kiri	Fitriana Kiri	Fitriana Kiri	Atim Kanan	Huda Kanan	Dian Kanan
<i>Farid Kanan</i>	Tino Kiri	Prima Kanan	Tino Kanan	Dian Kanan	Lia Kiri	Lia Kanan	Farid Kanan	Faizar Kiri	Sari Kanan

Data Uji	Hasil 1	Hasil 2	Hasil 3	Hasil 4	Hasil 5	Hasil 6	Hasil 7	Hasil 8	Hasil 9
<i>Fitriana Kanan</i>	Tareka Kiri	Fitriana Kanan	Habib Kanan						
<i>Fitriana Kiri</i>	Faizar Kanan	Fitriana Kiri	Sari Kiri	Atim Kanan	Dita Kanan	Tino Kanan	Tareka Kiri	Sari Kiri	Fitriana Kanan
<i>Habib Kanan</i>	Sanata Kanan	Tareka Kiri	Dita Kiri	Huda Kanan	Huda Kanan	Habib Kanan	Atim Kiri	Sanata Kiri	Faizar Kiri
<i>Habib Kiri</i>	Farid Kanan	Faizar Kanan	Dita Kiri	Sari Kanan	Lia Kanan	Lia Kiri	Atim Kanan	Lia Kanan	Syauqil Kiri
<i>Huda Kanan</i>	Huda Kanan	Sanata Kanan	Lia Kiri	Dian Kiri	Farid Kanan	Tareka Kanan			
<i>Huda Kiri</i>	Huda Kanan	Habib Kanan	Dita Kiri	Huda Kanan	Tareka Kiri	Sanata Kanan	Atim Kiri	Sanata Kiri	Faizar Kiri
<i>Lia Kanan</i>	Lia Kanan	Sari Kanan	Fitriana Kanan	Sari Kiri	Tino Kanan	Huda Kiri	Dita Kiri	Fitriana Kanan	Tareka Kiri
<i>Lia Kiri</i>	Dita Kanan	Faizar Kanan	Syauqil Kiri	Fitriana Kiri	Syauqil Kanan	Sari Kiri	Tareka Kiri	Huda Kiri	Dian Kanan
<i>Prima Kanan</i>	Sari Kiri	Huda Kiri	Dita Kiri	Atim Kanan	Lia Kanan	Syauqil Kiri	Atim Kiri	Agung Kanan	Dian Kanan
<i>Prima Kiri</i>	Farid Kanan	Faizar Kiri	Sari Kanan	Lia Kiri	Lia Kanan	Habib Kanan	Lia Kiri	Dian Kanan	Prima Kanan
<i>Sanata Kanan</i>	Farid Kanan	Tareka Kanan	Sanata Kanan	Prima Kiri	Syauqil Kanan	Habib Kiri	Syauqil Kiri		
<i>Sanata Kiri</i>	Fitriana Kanan	Lia Kanan	Sari Kanan	Tino Kanan	Sari Kiri	Huda Kiri	Dita Kiri	Fitriana Kanan	Tareka Kiri
<i>Sari Kanan</i>	Huda Kanan	Prima Kanan	Sari Kanan	Agung Kiri	Faizar Kiri	Tino Kanan	Sanata Kiri	Syauqil Kanan	Atim Kiri

Data Uji	Hasil 1	Hasil 2	Hasil 3	Hasil 4	Hasil 5	Hasil 6	Hasil 7	Hasil 8	Hasil 9
<i>Sari Kiri</i>	Fitriana Kanan	Dita Kiri	Tareka Kiri	Huda Kiri	Sari Kiri	Tino Kanan	Sari Kanan	Lia Kanan	Fitriana Kanan
<i>Syauqil Kanan</i>	Sari Kanan								
<i>Syauqil Kiri</i>	Lia Kanan	Atim Kanan	Tareka Kanan	Syauqil Kiri	Sanata Kiri	Lia Kiri	Atim Kiri	Lia Kanan	Sari Kanan
<i>Tareka Kanan</i>	Agung Kiri	Faizar Kanan	Tino Kanan	Tareka Kanan	Prima Kanan	Farid Kanan			
<i>Tareka Kiri</i>	Farid Kanan	Faizar Kiri	Sari Kanan	Habib Kanan	Lia Kiri	Lia Kiri	Lia Kanan	Dian Kanan	Prima Kanan
<i>Tino Kanan</i>	Farid Kanan	Prima Kanan	Tareka Kanan	Tino Kanan	Faizar Kanan	Agung Kiri			
<i>Tino Kiri</i>	Tino Kiri	Atim Kiri	Habib Kiri	Syauqil Kiri	Atim Kiri	Prima Kanan	Syauqil Kiri	Dian Kanan	Tareka Kanan
<i>Agung Kanan</i>	Agung Kanan	Huda Kiri	Sari Kiri	Dita Kiri	Atim Kanan	Lia Kanan	Syauqil Kiri	Atim Kiri	Agung Kanan

4.3 Pembahasan

Uji coba dilakukan sebanyak data uji yang tersedia yaitu sebanyak 32 kali yang ditunjukkan pada Tabel 4.1. Setiap koresponden diuji dengan dua data uji sidik jari, yaitu data uji citra sidik jari kanan, data uji citra sidik jari kiri, dan data uji citra sidik jari acak kanan dan kiri. Berdasarkan hasil uji yang ditunjukkan pada Tabel 4.1. Kesesuaian antara data uji untuk 9 teratas ditunjukkan pada Tabel 4.2 berikut.

Tabel 4.2 Kesesuaian Data Uji 9 Hasil Teratas

No	Data Uji	Kesesuaian	Keterangan
1	Agung Kanan	Cocok	Cocok pada hasil ke 5
2	Agung Kiri	Cocok	Cocok pada hasil ke 4
3	Atim Kanan	Cocok	Cocok pada hasil ke 2
4	Atim Kiri	Cocok	Cocok pada hasil ke 7
5	Dian Kanan	Cocok	Cocok pada hasil ke 3
6	Dian Kiri	Tidak Cocok	-
7	Dita Kanan	Tidak Cocok	-
8	Dita Kiri	Cocok	Cocok pada hasil ke 3
9	Faizar Kanan	Cocok	Cocok pada hasil ke 1
10	Faizar Kiri	Tidak Cocok	-
11	Farid Kanan	Cocok	Cocok pada hasil ke 7
12	Fitriana Kanan	Cocok	Cocok pada hasil ke 2
13	Fitriana Kiri	Cocok	Cocok pada hasil ke 2
14	Habib Kanan	Cocok	Cocok pada hasil ke 6

No	Data Uji	Kesesuaian	Keterangan
15	Habib Kiri	Tidak Cocok	-
16	Huda Kanan	Cocok	Cocok pada hasil ke 1
17	Huda Kiri	Tidak Cocok	-
18	Lia Kanan	Cocok	Cocok pada hasil ke 1
19	Lia Kiri	Tidak Cocok	-
20	Prima Kanan	Tidak Cocok	-
21	Prima Kiri	Tidak Cocok	-
22	Sanata Kanan	Cocok	Cocok pada hasil ke 3
23	Sanata Kiri	Tidak Cocok	-
24	Sari Kanan	Cocok	Cocok pada hasil ke 3
25	Sari Kiri	Tidak Cocok	-
26	Syauqil Kanan	Tidak Cocok	-
27	Syauqil Kiri	Cocok	Cocok pada hasil ke 4
28	Tareka Kanan	Cocok	Cocok pada hasil ke 4
29	Tareka Kiri	Tidak Cocok	-
30	Tino Kanan	Cocok	Cocok pada hasil ke 4
31	Tino Kiri	Cocok	Cocok pada hasil ke 1
32	Agung Kanan	Cocok	Cocok pada hasil ke 1 dan 9

Sedangkan tabel kesesuaian untuk 6 nilai teratas adalah :

Tabel 4.3 Kesesuaian Data Uji 6 Hasil Teratas

No	Data Uji	Kesesuaian	Keterangan
1	Agung Kanan	Cocok	Cocok pada hasil ke 5
2	Agung Kiri	Cocok	Cocok pada hasil ke 4
3	Atim Kanan	Cocok	Cocok pada hasil ke 2
4	Atim Kiri	Tidak Cocok	-
5	Dian Kanan	Cocok	Cocok pada hasil ke 3
6	Dian Kiri	Tidak Cocok	-
7	Dita Kanan	Tidak Cocok	-
8	Dita Kiri	Cocok	Cocok pada hasil ke 3
9	Faizar Kanan	Cocok	Cocok pada hasil ke 1
10	Faizar Kiri	Tidak Cocok	-
11	Farid Kanan	Tidak Cocok	-
12	Fitriana Kanan	Cocok	Cocok pada hasil ke 2
13	Fitriana Kiri	Cocok	Cocok pada hasil ke 2
14	Habib Kanan	Cocok	Cocok pada hasil ke 6
15	Habib Kiri	Tidak Cocok	-
16	Huda Kanan	Cocok	Cocok pada hasil ke 1
17	Huda Kiri	Tidak Cocok	-
18	Lia Kanan	Cocok	Cocok pada hasil ke 1
19	Lia Kiri	Tidak Cocok	-
20	Prima Kanan	Tidak Cocok	-
21	Prima Kiri	Tidak Cocok	-
22	Sanata Kanan	Cocok	Cocok pada hasil ke 3

No	Data Uji	Kesesuaian	Keterangan
23	Sanata Kiri	Tidak Cocok	-
24	Sari Kanan	Cocok	Cocok pada hasil ke 3
25	Sari Kiri	Tidak Cocok	-
26	Syauqil Kanan	Tidak Cocok	-
27	Syauqil Kiri	Cocok	Cocok pada hasil ke 4
28	Tareka Kanan	Cocok	Cocok pada hasil ke 4
29	Tareka Kiri	Tidak Cocok	-
30	Tino Kanan	Cocok	Cocok pada hasil ke 4
31	Tino Kiri	Cocok	Cocok pada hasil ke 1
32	Agung Kanan	Cocok	Cocok pada hasil ke 1

Kesesuaian data uji berdasarkan hasil 3 teratas ditunjukkan pada Tabel 4.4 sebagai berikut :

Tabel 4.4 Kesesuaian Data Uji 3 Hasil Teratas

No	Data Uji	Kesesuaian	Keterangan
1	Agung Kanan	Tidak Cocok	-
2	Agung Kiri	Tidak Cocok	-
3	Atim Kanan	Cocok	Cocok pada hasil ke 2
4	Atim Kiri	Tidak Cocok	-
5	Dian Kanan	Cocok	Cocok pada hasil ke 3
6	Dian Kiri	Tidak Cocok	-
7	Dita Kanan	Tidak Cocok	-

No	Data Uji	Kesesuaian	Keterangan
8	Dita Kiri	Cocok	Cocok pada hasil ke 3
9	Faizar Kanan	Cocok	Cocok pada hasil ke 1
10	Faizar Kiri	Tidak Cocok	-
11	Farid Kanan	Tidak Cocok	-
12	Fitriana Kanan	Cocok	Cocok pada hasil ke 2
13	Fitriana Kiri	Cocok	Cocok pada hasil ke 2
14	Habib Kanan	Tidak Cocok	-
15	Habib Kiri	Tidak Cocok	-
16	Huda Kanan	Cocok	Cocok pada hasil ke 1
17	Huda Kiri	Tidak Cocok	-
18	Lia Kanan	Cocok	Cocok pada hasil ke 1
19	Lia Kiri	Tidak Cocok	-
20	Prima Kanan	Tidak Cocok	-
21	Prima Kiri	Tidak Cocok	-
22	Sanata Kanan	Cocok	Cocok pada hasil ke 3
23	Sanata Kiri	Tidak Cocok	-
24	Sari Kanan	Cocok	Cocok pada hasil ke 3
25	Sari Kiri	Tidak Cocok	-
26	Syauqil Kanan	Tidak Cocok	-
27	Syauqil Kiri	Tidak Cocok	-
28	Tareka Kanan	Tidak Cocok	-
29	Tareka Kiri	Tidak Cocok	-

No	Data Uji	Kesesuaian	Keterangan
30	Tino Kanan	Tidak Cocok	-
31	Tino Kiri	Cocok	Cocok pada hasil ke 1
32	Agung Kanan	Cocok	Cocok pada hasil ke 1

Kesesuaian data uji berdasarkan hasil 1 teratas ditunjukkan pada Tabel 4.5 sebagai berikut :

Tabel 4.5 Kesesuaian Data Uji 1 Hasil Teratas

No	Data Uji	Kesesuaian	Keterangan
1	Agung Kanan	Tidak Cocok	-
2	Agung Kiri	Tidak Cocok	-
3	Atim Kanan	Tidak Cocok	-
4	Atim Kiri	Tidak Cocok	-
5	Dian Kanan	Tidak Cocok	-
6	Dian Kiri	Tidak Cocok	-
7	Dita Kanan	Tidak Cocok	-
8	Dita Kiri	Tidak Cocok	-
9	Faizar Kanan	Cocok	Cocok pada hasil ke 1
10	Faizar Kiri	Tidak Cocok	-
11	Farid Kanan	Tidak Cocok	-
12	Fitriana Kanan	Tidak Cocok	-
13	Fitriana Kiri	Tidak Cocok	-
14	Habib Kanan	Tidak Cocok	-

No	Data Uji	Kesesuaian	Keterangan
15	Habib Kiri	Tidak Cocok	-
16	Huda Kanan	Cocok	Cocok pada hasil ke 1
17	Huda Kiri	Tidak Cocok	-
18	Lia Kanan	Cocok	Cocok pada hasil ke 1
19	Lia Kiri	Tidak Cocok	-
20	Prima Kanan	Tidak Cocok	-
21	Prima Kiri	Tidak Cocok	-
22	Sanata Kanan	Tidak Cocok	-
23	Sanata Kiri	Tidak Cocok	-
24	Sari Kanan	Tidak Cocok	-
25	Sari Kiri	Tidak Cocok	-
26	Syauqil Kanan	Tidak Cocok	-
27	Syauqil Kiri	Tidak Cocok	-
28	Tareka Kanan	Tidak Cocok	-
29	Tareka Kiri	Tidak Cocok	-
30	Tino Kanan	Tidak Cocok	-
31	Tino Kiri	Cocok	Cocok pada hasil ke 1
32	Agung Kanan	Cocok	Cocok pada hasil ke 1

Dari hasil 32 data uji yang diujikan, terdapat beberapa hasil akurasi yang dikelompokkan menjadi 4 yaitu hasil uji 9 teratas, hasil uji 6 teratas, hasil uji 3 teratas, dan hasil 1 teratas. Pada hasil uji 9 teratas didapati hasil uji cocok

berjumlah 20 data uji dan tidak cocok berjumlah 12 data uji. Perhitungan akurasi hasil uji 9 teratas adalah sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\ &= \frac{20}{32} \times 100 \\ &= 62.50 \% \end{aligned}$$

Tabel 4.3 menunjukkan kesesuaian data uji pada urutan 6 teratas. Terdapat 18 data uji yang cocok dan sesuai dengan identifikasi dan 14 data uji yang tidak cocok. Hasil ini lebih rendah daripada jika hasil uji 8 teratas. Akurasi dapat di hitung sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\ &= \frac{18}{32} \times 100 \\ &= 56.25 \% \end{aligned}$$

Sedangkan Tabel 4.4 menunjukkan kesesuaian hasil data uji dengan hasil 3 teratas. Pada hasil uji tersebut, didapati 12 data uji yang cocok dan 20 data uji yang tidak cocok. Dibandingkan dengan hasil sebelumnya, hasil ini memberikan tingkat akurasi lebih rendah yaitu 37.50%. Perhitungan akurasi adalah sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\ &= \frac{12}{32} \times 100 \\ &= 37.50 \% \end{aligned}$$

Hasil uji 1 teratas ditunjukkan pada Tabel 4.5, pada tabel tersebut hasil data uji cocok berjumlah 5 dan hasil data uji tidak cocok berjumlah 27. Hasil ini jauh lebih rendah tingkat akurasi daripada hasil-hasil sebelumnya. Jika dihitung ke dalam persentase, maka akurasi dari hasil uji coba dengan 1 urutan teratas adalah sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\ &= \frac{5}{32} \times 100 \\ &= 15.625 \% \end{aligned}$$

Penelitian ini juga menghadirkan studi perbandingan tentang pengaruh teknik pengindeksan atau pengelompokan terhadap waktu pencarian. Studi perbandingan yang pertama adalah membandingkan proses identifikasi menggunakan metode *wavelet daubechies* dan diagram *NOHIS-Tree* dibandingkan dengan proses identifikasi menggunakan *wavelet daubechies db4* matlab dan metode pencarian *brute Force*. Tabel 4.6 menunjukkan studi hasil perbandingan tingkat akurasi dan efisiensi waktu pencarian.

Tabel 4.6 Perbandingan Efisiensi dan Akurasi Proses Identifikasi.

No	Data Uji	Akurasi (Kesesuaian Pencocokan)		Efisiensi (Waktu Pencarian)	
		Daubechies	Db4 Matlab	<i>NOHIS-Tree</i> (menit)	<i>Brute Force</i> (menit)
1.	Agung Kanan	Cocok pada hasil ke 5	Cocok pada hasil ke 4	00:07.71	02.59.23

No	Data Uji	Akurasi (Kesesuaian Pencocokan)		Efisiensi (Waktu Pencarian)	
		Daubechies	Db4 Matlab	<i>NOHIS-Tree</i> (menit)	<i>Brute Force</i> (menit)
2.	Agung Kiri	Cocok pada hasil ke 4	Cocok pada hasil ke 1,3,4	00:09.35	02:49.21
3.	Atim Kanan	Cocok pada hasil ke 2	Cocok pada hasil ke 1,2,3,4,5	00:11.20	03:51.74
4.	Atim Kiri	Cocok pada hasil ke 7	Cocok pada hasil ke 2,6,7	00:06.03	03:41.50
5.	Dian Kanan	Cocok pada hasil ke 3	Cocok pada hasil ke 1,4	00:09.37	03:35.59
6.	Dian Kiri	-	Cocok pada hasil ke 1,2,3,4,5,6	00:06.54	03:18.95
7.	Dita Kanan	-	-	00:15.66	03:11.95
8.	Dita Kiri	Cocok pada hasil ke 3	-	00:17.56	02:49.05
9.	Faizar Kanan	Cocok pada hasil ke 1	-	00:08.04	02:46.83
10.	Faizar Kiri	-	Cocok pada hasil ke 1,3,4,6	00:09.46	02:45.52
11.	Farid Kanan	Cocok pada hasil ke 7	Cocok pada hasil ke 1,2,3,4,5,6	00:11.25	03:00.23
12.	Fitriana Kanan	Cocok pada hasil ke 2	Cocok pada hasil ke 1,2,3,6,8,9	00:06.45	02:44.89
13.	Fitriana Kiri	Cocok pada hasil ke 2	Cocok pada hasil ke 7	00:16.16	02:43.42
14.	Habib Kanan	Cocok pada hasil ke 6	Cocok pada hasil ke 4	00:09.82	03:36.69
15.	Habib Kiri	-	Cocok pada hasil ke 6	00:09.95	03:24.90
16.	Huda Kanan	Cocok pada hasil ke 1	Cocok pada hasil ke 7,8	00:08.14	02:45.25

No	Data Uji	Akurasi (Kesesuaian Pencocokan)		Efisiensi (Waktu Pencarian)	
		Daubechies	Db4 Matlab	<i>NOHIS-Tree</i> (menit)	<i>Brute Force</i> (menit)
17.	Huda Kiri	-	Cocok pada hasil ke 2,4,5	00:17.60	02:47.91
18.	Lia Kanan	Cocok pada hasil ke 1	Cocok pada hasil ke 1,2	00:09.57	03:14.74
19.	Lia Kiri	-	Cocok pada hasil ke 2	00:09.75	02:39.93
20.	Prima Kanan	-	Cocok pada hasil ke 1,8	00:09.72	02:49.83
21.	Prima Kiri	-	Cocok pada hasil ke 1,4	00:09.63	02:39.05
22.	Sanata Kanan	Cocok pada hasil ke 3	Cocok pada hasil ke 2,9	00:09.43	03:01.97
23.	Sanata Kiri	-	Cocok pada hasil ke 1,2,7	00:09.68	02:28.01
24.	Sari Kanan	Cocok pada hasil ke 3	-	00:10.62	02:37.41
25.	Sari Kiri	-	Cocok pada hasil ke 1,2	00:10.39	03:04.29
26.	Syauqil Kanan	-	Cocok pada hasil ke 1,3,7	00:05.49	04:05.72
27.	Syauqil Kiri	Cocok pada hasil ke 4	Cocok pada hasil ke 1,2,5	00:11.08	03:56.18
28.	Tareka Kanan	Cocok pada hasil ke 4	Cocok pada hasil ke 2,3,7	00:09.30	03:38.21
29.	Tareka Kiri	-	Cocok pada hasil ke 8	00:09.69	03:59.91

No	Data Uji	Akurasi (Kesesuaian Pencocokan)		Efisiensi (Waktu Pencarian)	
		Daubechies	Db4 Matlab	<i>NOHIS-Tree</i> (menit)	<i>Brute Force</i> (menit)
30.	Tino Kanan	Cocok pada hasil ke 4	Cocok pada hasil ke 1,2,3,4	00:08.83	03:05.41
31.	Tino8 Kiri	Cocok pada hasil ke 1	-	00:10.03	03:18.67
32.	Agung Kanan	Cocok pada hasil ke 1 dan 9	Cocok pada hasil ke 1,2,6,8	00:08:56	02:57:23

Terdapat beberapa hasil akurasi dengan menggunakan *db4* matlab yang dikelompokkan menjadi 4 dari hasil 32 data uji yang diujikan yaitu hasil uji 9 teratas, hasil uji 6 teratas, hasil uji 3 teratas, dan hasil 1 teratas. Perhitungan akurasinya adalah sebagai berikut :

❖ 9 hasil teratas

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\
 &= \frac{27}{32} \times 100 \\
 &= 84.375 \%
 \end{aligned}$$

❖ 6 hasil teratas

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\
 &= \frac{24}{32} \times 100 \\
 &= 75 \%
 \end{aligned}$$

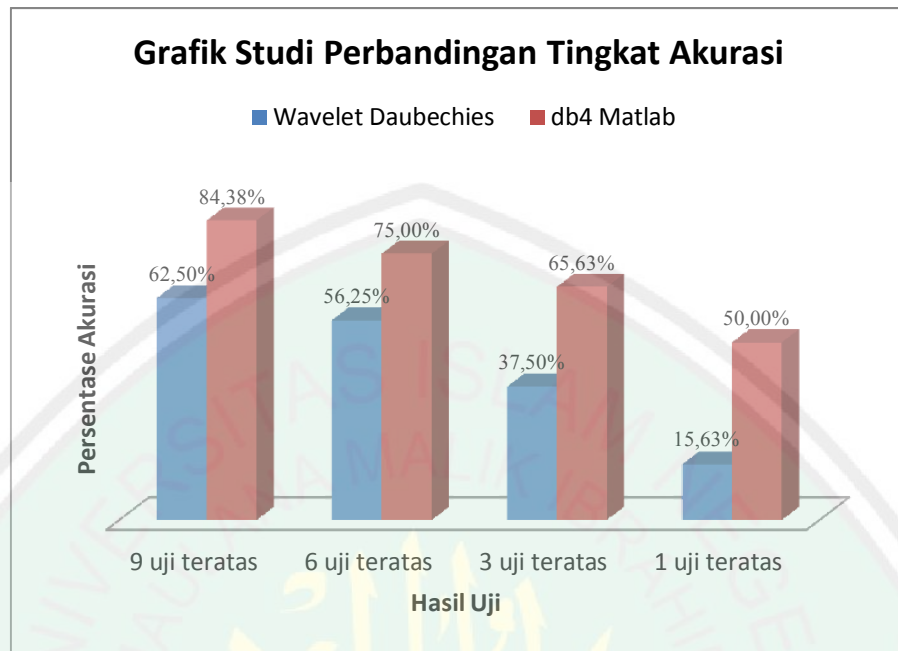
❖ 3 hasil teratas

$$\begin{aligned}
 Akurasi &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\
 &= \frac{21}{32} \times 100 \\
 &= 65.625 \%
 \end{aligned}$$

❖ 1 hasil teratas :

$$\begin{aligned}
 Akurasi &= \frac{\text{data cocok}}{\sum \text{data}} \times 100\% \\
 &= \frac{16}{32} \times 100 \\
 &= 50 \%
 \end{aligned}$$

Berikut grafik studi perbandingan tingkat akurasi antara metode wavelet daubechies dengan menggunakan kernel yang dibangun pada aplikasi dengan metode *daubechies 4 (db4)* yang merupakan fungsi yang terdapat pada *library* Matlab:



Gambar 4.10 Studi Perbandingan Tingkat Akurasi Wavelet Daubechies dengan Fungsi db4 Matlab

Berdasarkan hasil studi perbandingan pengaruh pengelompokan atau teknik *indexing* pada *CBIR* citra sidik jari dilihat dari catatan waktu ke empat data uji yang diperbandingkan dapat diperoleh hasil sebagai berikut :

$$\begin{aligned}
 \text{Efisiensi Brute Force} &= \frac{\sum \text{waktu pencarian}}{\sum \text{data}} \\
 &= \frac{5832}{32} \\
 &= 182,25 \text{ detik} \\
 &= 3.0375 \text{ menit}
 \end{aligned}$$

$$\begin{aligned}
 \text{Efisiensi NOHIS - Tree} &= \frac{\sum \text{waktu pencarian}}{\sum \text{data}} \\
 &= \frac{307}{32} \\
 &= 9.59375 \text{ detik}
 \end{aligned}$$

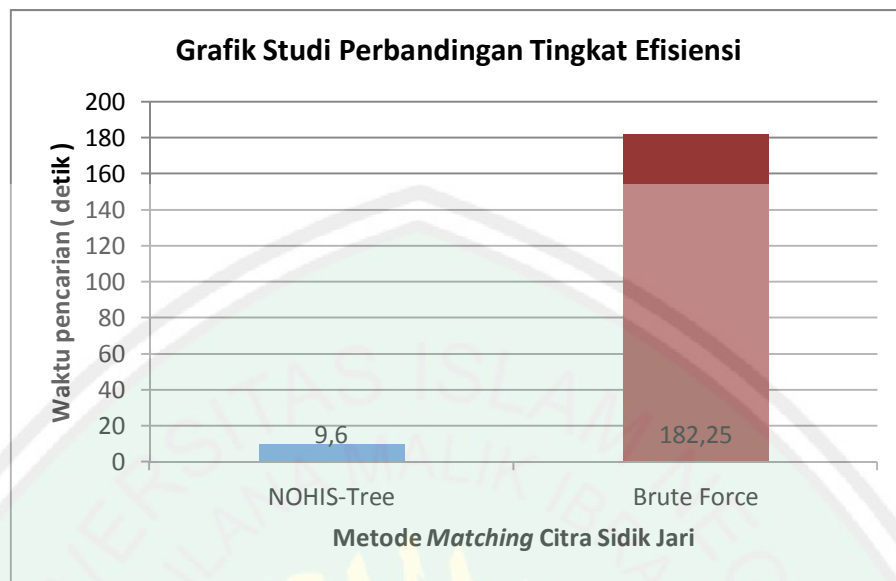
- Rata – Rata Waktu Pencarian dengan menggunakan *Brute Force*
 $= 182.25 \text{ detik} = 3.0375 \text{ menit}$
- Rata – Rata Waktu Pencarian dengan menggunakan *NOHIS – Tree*
 $= 9.59375 \text{ detik}$

Sehingga diketahui perbandingan waktu pencarian :

$$\text{NOHIS-Tree : Brute Force} = 1 : 18$$

Artinya bahwa metode *NOHIS-Tree* mempunyai efisiensi waktu pencarian 18 kali lebih cepat daripada metode *Brute Force*. Metode *NOHIS-Tree* hanya membutuhkan waktu rata-rata 9,6 detik pada proses pencarian dan identifikasinya. sedangkan *CBIR* dengan menggunakan metode *brute force* membutuhkan waktu lebih lama yakni 182,25 detik atau 3,03 menit.

Berikut grafik dari hasil studi perbandingan waktu pencarian dengan menggunakan metode Diagram *NOHIS-Tree* dengan metode *Similarity Search* menggunakan metode *Brute Force*.



Gambar 4.11 Grafik Studi Perbandingan Efisiensi waktu pencarian antara NOHIS-Tree dengan Brute Force

4.4 Integrasi *Content Based Image Retrieval* Citra Sidik Jari menggunakan metode *Wavelet Daubechies* dan diagram *NOHIS-Tree* dengan Al-Qur'an

Sidik jari adalah hasil reproduksi tapak jari baik yang sengaja diambil, dicapkan dengan tinta, maupun bekas yang ditinggalkan benda karena pernah tersentuh kulit telapak tangan atau kaki. Sebelumnya, orang menganggap sidik jari sebagai lengkungan-lengkungan biasa tanpa makna khusus. Namun dalam Al-qur'an, Allah merujuk kepada sidik jari yang kurang menarik perhatian orang waktu itu dan mengarahkan perhatian kita pada arti penting sidik jari, yang baru mampu dipahami di zaman sekarang.

Pada abad ke-7 M, Al-qur'an telah menyebutkan bahwa sidik jari menjadi tanda pengenalan manusia. Dalam Al-qur'an disebutkan mudah bagi Alla untuk

menghidupkan manusia setelah kematiannya, pernyataan tentang sidik jari manusia secara khusus diterangkan dalam surat Al-Qiyamah ayat 3 dan 4.

أَتَحْسَبُ الْإِنْسَانُ أَنْ يَجْمَعَ عِظَامَهُ ﴿٣﴾ بَلَىٰ قَدَرِينَ عَلَىٰ أَنْ نُسَوِّيَ بَنَانَهُ ﴿٤﴾

Artinya : “Apakah manusia mengira, bahwa Kami tidak akan mengumpulkan (kembali) tulang belulangnyanya?. bukan demikian, sebenarnya Kami Kuasa menyusun (kembali) jari jemarinya dengan sempurna.”

Dalam Al-Qur'anul Karim Allah SWT berfirman :

وَقَالُوا لَجُلُودِهِمْ لِمَ شَهِدْتُمْ عَلَيْنَا ۖ قَالُوا أَنْطَقَنَا اللَّهُ الَّذِي أَنْطَقَ كُلَّ شَيْءٍ وَهُوَ خَلَقَكُمْ أَوَّلَ مَرَّةٍ وَإِلَيْهِ تُرْجَعُونَ ﴿٢١﴾ وَمَا كُنْتُمْ تَسْتَتِرُونَ أَنْ يَشْهَدَ عَلَيْكُمْ سَمْعُكُمْ وَلَا أَبْصَارُكُمْ وَلَا جُلُودُكُمْ وَلَكِنْ ظَنَنْتُمْ أَنَّ اللَّهَ لَا يَعْلَمُ كَثِيرًا مِمَّا تَعْمَلُونَ ﴿٢٢﴾

Artinya : “Dan mereka berkata kepada kulit mereka: "Mengapa kamu menjadi saksi terhadap kami?" kulit mereka menjawab: "Allah yang menjadikan segala sesuatu pandai berkata telah menjadikan Kami pandai (pula) berkata, dan Dia-lah yang menciptakan kamu pada kali pertama dan hanya kepada-Nya lah kamu dikembalikan". Kamu sekali-sekali tidak dapat bersembunyi dari kesaksian pendengaran, penglihatan dan kulitmu kepadamu bahkan kamu mengira bahwa Allah tidak mengetahui kebanyakan dari apa yang kamu kerjakan (QS:Al-Fushshilat ayat 21 dan 22).

Kandungan Surat Al-Fushshilat ayat 21 dan 22 memberikan kita informasi bahwa kelak di hari akhir mata, telinga, dan kulit manusia akan menjadi saksi atas apa yang dia kerjakan selama di dunia. Kulit manusia pada jari manusia (sidik jari) yang bersifat permanent diciptakan oleh Allah SWT. Dia berkuasa untuk

membuatnya berfungsi kembali di hari kebangkitan sebagai saksi atas amaliyah manusia.

Dalam penelitian ini digunakan metode *Wavelet Daubechies* dan Diagram *NOHIS-Tree*. Dengan diterapkannya teknik pengindeksan diagram *NOHIS-Tree* maka proses identifikasi citra sidik jari menghasilkan waktu pencarian yang lebih optimal karena nilai-nilai fitur tertentu telah terklasifikasi sehingga memudahkan pencarian atau pengenalan. Hal ini telah dijelaskan dalam Al-Qur'an surat Al-Hujurat ayat 13 bahwa manusia diciptakan berbangsa-bangsa dan bersuku-suku agar supaya mudah dalam proses saling kenal-mengenal. Allah SWT berfirman :

يٰٓأَيُّهَا النَّاسُ إِنَّا خَلَقْنٰكُمْ مِنْ ذَكَرٍ وَأُنْثٰى وَجَعَلْنٰكُمْ شُعُوبًا وَقَبَآئِلَ لِتَعَارَفُوْٓا ۚ إِنَّ أَكْرَمَكُمْ عِنْدَ
 ٱللّٰهِ أَتْقٰىكُمْ ۚ إِنَّ ٱللّٰهَ عَلِيْمٌ خَبِيْرٌ ﴿١٣﴾

Artinya : “Hai manusia, Sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa - bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. Sesungguhnya orang yang paling mulia diantara kamu disisi Allah ialah orang yang paling taqwa diantara kamu. Sesungguhnya Allah Maha mengetahui lagi Maha Mengenal.” (QS Al-Hujurat : 13)

Content Based Image Retrieval citra sidik jari menggunakan *Wavelet Daubechies* dan Diagram *NOHIS-Tree* ini adalah merupakan salah satu alat teknologi informasi yang dapat digunakan oleh manusia.

Perangkat akses dan perangkat keamanan menggunakan sidik jari pada dasarnya adalah sebuah alat. Sebuah alat yang digunakan manusia mempunyai potensi untuk digunakan kepada kebaikan dan keburukan. Aplikasi sidik jari yang

akurat dan proses pencariannya yang cepat, penggunaannya bisa membantu pihak berwajib atau pihak-pihak yang berkepentingan untuk menindaklanjuti, mengurangi atau mencegah kriminalitas yang dilakukan oleh manusia. Hal ini sesuai dengan yang diperintahkan Allah SWT kepada manusia, yaitu :

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ ۖ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۚ وَاتَّقُوا اللَّهَ ۚ إِنَّ اللَّهَ شَدِيدُ
الْعِقَابِ

Artinya : “Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. dan bertakwalah kamu kepada Allah, Sesungguhnya Allah Amat berat siksa-Nya. (QS : Al-Maidah ayat 2)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil implementasi dan uji coba yang telah peneliti lakukan dapat disimpulkan bahwa :

- a. *Wavelet Daubechies* dan Diagram *NOHIS-Tree* dapat diterapkan pada aplikasi *Content Based Image Retrieval* citra sidik jari. *Wavelet Daubechies* digunakan sebagai metode untuk mengekstraksi *feature* citra sidik jari input dan citra sidik jari *query*. Citra sidik jari di transformasi hingga level 2 yang menghasilkan 4 nilai yaitu aproksimasi, horizontal, vertikal dan diagonal yang kemudian dihitung nilai energinya dan hasil perhitungan nilai energi dari citra input tersebut disimpan dalam database. Data hasil perhitungan nilai energi citra sidik jari yang sudah tersimpan pada database akan digunakan untuk proses *indexing*. Proses *indexing* ini diperlukan untuk melakukan proses pencarian yang cepat dan efisien. Teknik pengindeksan berdasarkan diagram *NOHIS-Tree* diterapkan pada nilai energi horizontal dan vertikal. Pada proses identifikasi, data uji akan mengalami proses pencarian dan pengklasifikasian dari nilai *feature*-nya menggunakan teknik pengindeksan *NOHIS-tree*. Selanjutnya, proses *Similarity Search* menggunakan perbandingan nilai ekstraksi *feature* antara citra *query* dengan citra database dengan menggunakan nilai

Euclidean Distance. 9 hasil teratas dianggap sebagai hasil identifikasi *Content Based Image Retrieval* citra sidik jari .

- b. Performa *Wavelet Daubechies* sebagai metode ekstraksi fitur pada aplikasi *Content Based Image Retrieval* citra sidik jari terbagi menjadi 4 hasil yaitu hasil 9 teratas sebesar 62,50%, hasil uji 6 teratas memiliki akurasi 56,25%, hasil uji 3 teratas memiliki akurasi 37,50%, dan hasil uji 1 teratas memiliki akurasi sebesar 15,625%.
- c. Performa *CBIR* dengan menggunakan metode *NOHIS-Tree* mempunyai efisiensi waktu pencarian 18 kali lebih cepat daripada metode *brute force*. Metode *NOHIS-Tree* membutuhkan waktu pada setiap proses pencarian dan identifikasi rata-rata 9,6 detik. Sedangkan *CBIR* dengan menggunakan metode *brute force* membutuhkan waktu lebih lama yakni 182,25 detik atau 3,04 menit.

5.2 Saran

Terdapat banyak kekurangan dalam penelitian aplikasi *Content Based Image Retrieval* citra sidik jari ini. Oleh karena itu penulis menyarankan beberapa hal sebagai bahan pengembangan selanjutnya, diantaranya :

- a. Pengambilan data tidak lagi menggunakan *stamp pad* dan *stamp pad ink* dikarenakan hasil yang didapat tidak dapat rata antara sidik jari sampel satu dan lainnya.
- b. Alat *scan* yang digunakan diusahakan tidak membuat *noise* file citra sidik jari yang dihasilkan.

- c. Perlu dilakukan studi lebih mendalam tentang pembuatan kernel *wavelet daubechies* secara manual dan jenis *wavelet* lainnya untuk diterapkan sebagai metode pengekstraksi fitur. Serta tetapan rumus atau persamaan yang valid untuk mencari nilai-nilai energi dari transformasi *wavelet* dua dimensi untuk memperoleh tingkat akurasi yang lebih optimal.
- d. Perlu diadakan penelitian lanjutan untuk mendapatkan tingkat akurasi yang lebih baik dengan tingkat efisiensi optimal, yaitu dengan menggabungkan *Othonormal Wavelet Daubechies* dengan Teknik *Indexing Diagram NOHIS-Tree*.

DAFTAR PUSTAKA

- Cohen, A., Daubechies, I. dan Feauveau, J. -C. (1992). *Biorthogonal bases of compactly supported wavelets*. *Communications on Pure and Applied Mathematics* 45 (5): 485–560.
- D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar. 2003. *Handbook of Fingerprint Recognition*, Springer, New York.
- Daubechies, Ingrid. 1988. *Orthonormal Bases of Compactly Supported Wavelets*. *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909-996.
- Falasev, RS, et al. 2009. *Pengenalan Sidik Jari Manusia Dengan Matriks Kookurensi Aras Keabuan (Gray Level Co-Ocurrence Matrix)*. Makalah Tugas Akhir S-1, Universitas Diponegoro, Semarang.
- Hendarko, G. 2010. *Identifikasi Citra Sidikjari Menggunakan Alihragam Wavelet Dan Jarak Euclidean*. Skripsi S-1, Universitas Diponegoro, Semarang.
- Jeffry dan Wibowo, Eko. 2010. *Peningkatan performa retrieval citra berbasis konten dengan pendekatan histogram warna termodifikasi*. Skripsi. Fakultas Komputer Sains, Universitas Bina Nusantara.
- Karmilasari, dan Sumarna, Agus. 2011. *Temu Kenali Citra Berbasis Konten Warna*. Diterbitkan dalam Seminar Nasional Aplikasi Teknologi Informasi 2011 (SNATI 2011), ISSN : 1907-5022.
- Khudori, AN., Muhammad Faisal, Ririen Kusumawati. 2012. *Sistem Identifikasi Sidik Jari Dengan Menggunakan Tranformasi Wavelet Dan Artificial Neural Network*. Diterbitkan dalam IEEE International Journal of Science and Technology (IJSTE), Vol. 1, No. 1, Maret 2012.12.17, ISSN : 2252-5297.
- Kusrini, dan Harjoko, Agus. 2008. *Pencarian Citra Visual Berbasis Isi Citra Menggunakan Fitur Warna Citra*. Jurnal. STMIK AMIKOM, Yogyakarta.
- Minarni. (2004). *Klasifikasi sidik jari dengan pemrosesan awal transformasi wavelet*. Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Padang. Diterbitkan dalam Jurnal Transmisi, Vol. 8, No. 2, Desember 2004 : 37 – 41.
- Nelvi, Fitriana. 2013. *Identifikasi Citra Sidik jari Menggunakan Metode Analytical Geometry dan Wavelet Transform*. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Taileb, Mounira. 2013. *Content Based Image Retrieval System Using NOHIS-tree*. King Abdul Aziz University, Faculty of Computing and Technology.
- Taileb, Mounira and Sami Touati. 2011. *NOHIS-Tree: High-Dimensional Index*

Structure for Similarity Search. In International Science Index 59, pp. 350 – 358.

Taileb, M. Lamrous, S., and Touati, S. 2008. *Non Overlapping Hierarchical Index Struture*. In *International Journal of Computer Science*, vol. 3 no. 1, pp. 29-35.

Tarigan, R.Agatha and Hidayatno, Achmad and Zahra, A.Ajulian (2012). *Aplikasi Pengenalan Sidik Jari Dengan Wavelet Symlet Dan Wavelet Daubechies Menggunakan Jaringan Saraf Tiruan Perambatan Balik*. Undergraduate Thesis, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Semarang.

Sediyono, Eko, dan Nataliani,Yessica, dan Mariana,Chrisanty. 2009. *Klasifikasi Sidik Jari dengan Menggunakan Metode Wavelet Symlet*. Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga. Diterbitkan dalam Jurnal Informatika, Vol. 5, No. 2, November 2009.

Sumarna, Agus. 2010. *CBIR Berdasarkan Ekstraksi Fitur Warna Menggunakan Java*. Skripsi. Jurusan Informatika, Fakultas Teknologi Industri ,Universitas Gunadarma, Depok.

LAMPIRAN 1 Database Citra Sidik Jari

ID	Nama	Keterangan	Aproksimasi	Horizontal	Vertikal	Diagonal
100010145	Agung Satrio Buwono	Jempol Kanan	9.995.901.638.729.360	0.0000015241407346497805	0.0000006633024485086695	0.00000000000015671025315715065
100020145	Agung Satrio Buwono	Jempol Kanan	9.994.082.037.878.590	0.000010671157991093403	0.00000009468321118718417	0.00000000000020385068895891217
100040145	Agung Satrio Buwono	Jempol Kanan	9.995.101.498.105.820	0.000003376622923289743	0.00000040372683376861017	0.00000000016310016973736477
100050145	Agung Satrio Buwono	Jempol Kanan	9.993.901.759.795.110	0.000008489501880747525	0.000000017974220491316538	0.000000000033935264395084066
100060145	Agung Satrio Buwono	Jempol Kanan	999.932.072.360.705	0.000002594019493455651	0.0000015856123590195802	0.00000000013785189663075055
100070145	Agung Satrio Buwono	Jempol Kanan	9.996.653.529.865.970	0.000003927847933665159	0.00000009574241333982169	0.0000000000008425167579051298
100080145	Agung Satrio Buwono	Jempol Kanan	9.997.370.032.781.360	0.0000041250946534863855	0.000000515235229152193	0.0000000000033710964222814822
100090145	Agung Satrio Buwono	Jempol Kanan	999.857.054.716.542	0.00000005870261721501551	0.0000009045244671340256	0.0000000003568005707229014
100110145	Agung Satrio Buwono	Jempol Kiri	10.002.752.248.955.800	0.0000005318054315700827	0.0000004228751899411822	0.0000000001231061631030309
100120145	Agung Satrio Buwono	Jempol Kiri	10.001.404.571.806.800	0.0000009285516269211291	0.00000006703573670503222	0.000000000005953540447491735
100160145	Agung Satrio Buwono	Jempol Kiri	9.999.181.167.995.050	0.0000002526139178215119	0.00000009204001883384279	0.000000000007588661914429715
100170145	Agung Satrio Buwono	Jempol Kiri	9.999.314.250.139.480	0.00000013233655175588054	0.000000001134143814393344	0.00000000016325041588706657
100180145	Agung Satrio Buwono	Jempol Kiri	10.003.813.847.332.700	0.0000026360327526468757	0.00000008200021747138256	0.0000000001024834589326785
100190145	Agung Satrio Buwono	Jempol Kiri	9.997.363.752.389.340	0.000000889720960052104	0.00000013520153528659341	0.00000000005274920277805611
100200145	Agung Satrio Buwono	Jempol Kiri	10.001.070.400.030.900	0.0000005853237367005442	0.00000005284776803754509	5,76E-02
100210192	Atim Mufaridah	Jempol Kanan	10.000.540.642.434.300	0.0000007721242230921152	0.0000013164793362973194	0.0000000000027034694787650286
100230192	Atim Mufaridah	Jempol Kanan	10.004.899.549.364.500	0.000007341774395355674	0.00000006916248238402999	0.000000000008360235079205152
100240192	Atim Mufaridah	Jempol Kanan	9.999.292.237.646.870	0.00000040441067116235203	0.0000009815113072956224	0.000000000007959720072551153
100250192	Atim Mufaridah	Jempol Kanan	9.997.651.458.981.300	0.0000005918060587026737	0.00000016438795756580552	0.00000000000015816981217358396
100260192	Atim Mufaridah	Jempol Kanan	9.999.374.868.844.490	0.00000012265207100208635	0.00000043886202452296014	0.00000000000010305270797333137
100270192	Atim Mufaridah	Jempol Kanan	9.991.731.672.656.090	0.000015652462780314607	0.00000003169378044138406	0.00000000000045127794769473735

100280192	Atim Mufarihah	Jempol Kanan	10.003.326.038.699.100	0.000002136857286210335	0.00000004144132104760157	0.000000000006198386492020711
100290192	Atim Mufarihah	Jempol Kanan	10.001.609.713.227.800	0.0000012895601340779255	0.00000010983573168162277	0.00000000000042374506277091535
100300192	Atim Mufarihah	Jempol Kanan	9.996.382.497.832.580	0.000001444760635792346	0.0000003844303343007888	0.00000000001713934427272688
100310192	Atim Mufarihah	Jempol Kiri	9.997.683.598.497.100	0.00000009457735837564649	0.000000725555547612997	0.0000000000011277900431577395
100320192	Atim Mufarihah	Jempol Kiri	10.001.032.275.389.300	0.00000016618854468404768	0.000000015420544457952055	0.00000000024703672067043596
100330192	Atim Mufarihah	Jempol Kiri	9.997.087.241.410.140	0.0000007785041461944394	0.000000329740276475822	0.000000000000005419492758040566
100340192	Atim Mufarihah	Jempol Kiri	9.993.658.559.611.600	0.000008456419381626056	0.00000006948739595656783	0.00000000000013981756439436551
100370192	Atim Mufarihah	Jempol Kiri	9.999.914.241.931.330	0.0000004514727590455278	0.0000005150998373005993	0.0000000000008457677822221943
100380192	Atim Mufarihah	Jempol Kiri	9.998.805.801.960.450	0.00000011500064412776751	0.0000000669700413779763	0.000000000000618278987465987
100390192	Atim Mufarihah	Jempol Kiri	9.999.024.408.256.300	0.00000011153393162373999	0.000000022047233361622877	0.00000000002870494705503745
100400192	Atim Mufarihah	Jempol Kiri	10.003.431.308.878.100	0.0000014364205129319874	0.00000026480585233667526	0.000000000005792552204811804
100410197	Dian Anggraini	Jempol Kanan	9.994.511.081.571.610	0.000003863826627978411	0.0000005580075955261125	0.0000000010352293444471719
100420197	Dian Anggraini	Jempol Kanan	9.998.216.924.746.470	0.000004638740709203086	0.0000015341674219631492	0.000000005560683996749159
100430197	Dian Anggraini	Jempol Kanan	9.997.516.932.160.500	0.0000005230430583514824	0.0000002456985698076774	0.0000000005159228223036572
100440197	Dian Anggraini	Jempol Kanan	1.000.067.899.315.670	0.0000003977140090533136	0.00000009799175122374741	0.0000000004787978252552657
100450197	Dian Anggraini	Jempol Kanan	10.005.915.775.707.900	0.000005239972154114937	0.0000004461615928717382	0.0000000000001589582369589043
100460197	Dian Anggraini	Jempol Kanan	10.001.504.438.655.900	0.000000004100268762605003	0.0000006682990720370424	0.0000000000016160265880765945
100480197	Dian Anggraini	Jempol Kanan	9.996.287.131.464.030	0.0000003088016515298172	0.0000016825948619308965	0.0000000000014127289234998312
100490197	Dian Anggraini	Jempol Kanan	10.002.533.325.199.800	0.0000007277911632392492	0.000000170959729941299	8,12E-03
100500197	Dian Anggraini	Jempol Kanan	10.000.674.982.735.900	0.00000003870059523928264	0.000000019914299579158736	0.00000000000012747018595742568
100520197	Dian Anggraini	Jempol Kiri	10.004.207.660.889.600	0.0000022713395918133332	0.00000031137283889266726	0.0000000014826238760724723
100530197	Dian Anggraini	Jempol Kiri	9.991.632.467.587.760	0.000008424929358670171	0.000001646913675307155	0.0000000000015686657408260625
100540197	Dian Anggraini	Jempol Kiri	9.992.774.609.212.840	0.00001902498891776828	0.0000005174037543286753	0.0000000008472983257814943
100550197	Dian Anggraini	Jempol Kiri	9.999.357.123.296.660	0.0000007352390583437677	0.0000002855422705941797	0.0000000000027415019178803025

100560197	Dian Anggraini	Jempol Kiri	10.004.310.417.805.600	0.0000026716022298349063	0.0000002694217292921101	0.000000000020009209901114747
100570197	Dian Anggraini	Jempol Kiri	9.998.347.312.508.240	0.0000004894748209057239	0.0000023180510237492356	0.000000000012165854191336582
100580197	Dian Anggraini	Jempol Kiri	10.001.364.950.418.400	0.0000021315103284303953	0.0000006405906679041783	0.0000000000522200843677203
100590197	Dian Anggraini	Jempol Kiri	9.999.443.341.771.520	0.000000000035635028763495006	0.00000010917827877666174	0.0000000033707181325146777
100600197	Dian Anggraini	Jempol Kiri	10.001.286.525.716.600	0.00000043426710044000164	0.0000000002658423263373977	0.0000000000003106816548659907
100610011	Yudita Prihatini	Jempol Kanan	9.998.046.663.264.070	0.00000018314830382949495	0.0000003261741551654974	0.0000000004999166740925181
100620011	Yudita Prihatini	Jempol Kanan	1.000.799.768.065.550	0.000012463686216731278	0.00000022247171990276083	0.00000000001618126143048336
100630011	Yudita Prihatini	Jempol Kanan	10.001.769.753.315.400	0.000001341043787716126	0.00000009040866072058442	0.0000000007553367545996149
100640011	Yudita Prihatini	Jempol Kanan	999.864.691.729.202	0.00000005965669096607291	0.00000018975146366894372	0.00000000001080871964790729
100650011	Yudita Prihatini	Jempol Kanan	9.998.154.251.467.980	0.0000001787850834435061	0.0000002510968651249767	0.000000000001017398037122942
100660011	Yudita Prihatini	Jempol Kanan	1.000.926.249.932.670	0.00001545301109391167	0.0000004910364526320641	0.000000000002540856743468462
100670011	Yudita Prihatini	Jempol Kanan	999.609.079.785.664	0.0000006875046745128438	0.000001264997777943511	0.0000000000008315905049796987
100680011	Yudita Prihatini	Jempol Kanan	9.992.820.038.252.380	0.00001344624324146992	0.000000005168654887371814	0.00000000001929922076763175
100690011	Yudita Prihatini	Jempol Kanan	10.002.865.767.264.800	0.000004199699154917014	0.0000003800742747821022	0.0000000000000011642897438915438
100710011	Yudita Prihatini	Jempol Kiri	999.860.042.547.776	0.00000000009652043244060799	0.0000004553094583315425	0.000000000023168011089850384
100730011	Yudita Prihatini	Jempol Kiri	1.000.064.073.179.470	0.000000009351317405215408	0.000000050023149629577833	1,80E-24
100740011	Yudita Prihatini	Jempol Kiri	9.999.812.697.101.420	0.0000000010182335903296453	0.0000000027536495809538266	0.00000000008587519309333379
100750011	Yudita Prihatini	Jempol Kiri	9.997.594.552.544.040	0.0000000016888017856756231	0.0000015472893209329011	9,51E-02
100760011	Yudita Prihatini	Jempol Kiri	999.953.819.645.002	0.0000004180873205962906	0.0000007556256974950327	0.00000000006779593777391995
100770011	Yudita Prihatini	Jempol Kiri	10.004.643.143.973.900	0.000002938874978442436	0.00000037011862459691266	0.0000000000019207602022134445
100790011	Yudita Prihatini	Jempol Kiri	10.000.017.695.761.900	0.00000032358399419235687	0.0000003099011697004786	0.0000000000010948710377349527
100800011	Yudita Prihatini	Jempol Kiri	9.997.593.678.532.060	0.0000014441489603340812	0.00000000006592990765717112	0.00000000009264747413546877
100810220	Ahmad Faizar Rahman	Jempol Kanan	9.995.581.367.686.800	0.000002367212537437014	0.00000044739660419374683	0.000000000004444940721882071
100840220	Ahmad Faizar Rahman	Jempol Kanan	10.000.001.851.610.000	0.00000016955622493705704	0.00000017303372882525026	0.000000000010728057396971528

100850220	Ahmad Faizar Rahman	Jempol Kanan	9.994.400.231.338.490	0.000007397194940029829	0.000000010680623891875774	0.0000000005220844960963611
100860220	Ahmad Faizar Rahman	Jempol Kanan	9.993.030.745.567.220	0.000005237758863395344	0.000001438897399705327	0.00000000000853217914227805
100870220	Ahmad Faizar Rahman	Jempol Kanan	9.995.318.778.858.410	0.0000014454063626381546	0.000001229020039408718	0.00000000009013357864524952
100880220	Ahmad Faizar Rahman	Jempol Kanan	10.001.199.709.556.100	0.00000024562889448125693	0.0000000109443463947902	0.00000000000015059165663127992
100890220	Ahmad Faizar Rahman	Jempol Kanan	9.998.052.731.675.050	0.00000022036011063194817	0.0000002538044693258998	0.00000000000021730755370745545
100900220	Ahmad Faizar Rahman	Jempol Kanan	10.001.203.487.404.800	0.000000018197126830518747	0.0000005252697557263686	0.00000000014083429822174836
100910220	Ahmad Faizar Rahman	Jempol Kiri	9.999.314.651.354.260	0.0000000011693586944990491	0.00000014203558077112577	3,46E-23
100920220	Ahmad Faizar Rahman	Jempol Kiri	10.001.630.238.677.200	0.00000031675309400901325	0.0000000553528365057547	0.00000000028920626336071744
100930220	Ahmad Faizar Rahman	Jempol Kiri	1.000.078.092.375.200	0.0000009356480735859339	0.00000036950305325225973	0.000000000963037670581189
100940220	Ahmad Faizar Rahman	Jempol Kiri	9.997.603.373.550.150	0.0000032657293197908735	0.0000003464821044655477	0.00000000040483709302484004
100950220	Ahmad Faizar Rahman	Jempol Kiri	10.001.523.649.810.000	0.00000033004107314322426	0.0000017539318455040713	0.0000000001422528961758477
100960220	Ahmad Faizar Rahman	Jempol Kiri	10.002.706.329.555.500	0.0000010060837339207224	0.00000011794749521269533	0.0000000000435749746208729
100970220	Ahmad Faizar Rahman	Jempol Kiri	9.994.835.528.737.140	0.0000024017440826563767	0.0000011372973728680973	0.0000000011308743317609853
100980220	Ahmad Faizar Rahman	Jempol Kiri	9.999.821.856.225.700	0.00000005378190205119778	0.0000001037130737124823	0.0000000000011312547367133304
100990220	Ahmad Faizar Rahman	Jempol Kiri	10.001.389.390.714.000	0.00000068788691741379	0.000000016943685795007136	0.0000000002070561474564312
101010219	Fitriana Nelvi	Jempol Kanan	999.289.790.809.186	0.000005699283866728326	0.0000013554033397733185	0.0000000000002001388263192259
101020219	Fitriana Nelvi	Jempol Kanan	10.001.428.188.642.400	0.00000045846900127823103	0.0000000013934007086737179	0.0000000000001316578107237341
101030219	Fitriana Nelvi	Jempol Kanan	9.998.415.286.993.870	0.0000007756959582531995	0.000000010060674386747348	0.00000000014291671785324264
101060219	Fitriana Nelvi	Jempol Kanan	10.005.974.348.731.800	0.00001240281349089361	0.00000028423324491042925	0.00000000000360697169437793
101070219	Fitriana Nelvi	Jempol Kanan	9.999.598.955.611.780	0.0000005930714070697494	0.00000032773980867801243	0.00000000000840159176296371
101080219	Fitriana Nelvi	Jempol Kanan	999.783.589.749.206	0.000000024794988291669943	0.000001561386969505405	0.00000000009961361297965189
101090219	Fitriana Nelvi	Jempol Kanan	10.006.387.536.542.600	0.000013899728520155269	0.0000002947288752681489	0.00000000006262878425104513
101100219	Fitriana Nelvi	Jempol Kanan	9.998.385.429.849.920	0.00000005423572583681216	0.0000010820229320246223	9,73E-26
101110219	Fitriana Nelvi	Jempol Kiri	9.997.878.403.280.360	0.0000003958563309732523	0.00000018490731529023203	0.0000000000028055525214463858

101120219	Fitriana Nelvi	Jempol Kiri	9.995.520.495.397.000	0.0000010713916314892467	0.0000014518386817338245	3,39E-23
101130219	Fitriana Nelvi	Jempol Kiri	10.000.945.581.639.000	0.0000004282002915500489	0.000000031426095979364766	0.00000000001863161097606887
101140219	Fitriana Nelvi	Jempol Kiri	10.001.639.861.569.400	0.00000010946144956756283	0.000000239155449226701	1,53E+00
101160219	Fitriana Nelvi	Jempol Kiri	10.000.084.833.524.300	0.00000013645289610991026	0.00000016360967971518054	0.00000000005365635367296594
101170219	Fitriana Nelvi	Jempol Kiri	10.000.497.572.827.200	0.000000027095054840404186	0.00000017089052711726546	5,52E-26
101180219	Fitriana Nelvi	Jempol Kiri	100.029.703.887.476	0.0000012775152242323945	0.00000013848675017798916	0.00000000030019615720422203
101190219	Fitriana Nelvi	Jempol Kiri	9.999.891.884.182.060	0.000000059255395380785396	0.00000003581866300108391	0.000000000000011678033439804282
101200219	Fitriana Nelvi	Jempol Kiri	9.997.970.342.767.880	0.00000013041087811738905	0.0000018934096355913934	5,84E-02
101210124	Habib Abdullah	Jempol Kanan	10.001.600.881.370.000	0.0000007934655159436622	0.000000008291056710773377	0.0000000000004859188016186595
101220124	Habib Abdullah	Jempol Kanan	1.000.097.606.315.000	0.00000005406671282065291	0.0000000668043956436265	0.0000000000024680223378254103
101230124	Habib Abdullah	Jempol Kanan	9.998.898.403.616.360	0.00000012856406245497152	0.000000033366671490588646	0.0000000000919601820446593
101240124	Habib Abdullah	Jempol Kanan	999.801.967.724.111	0.0000008510624723254934	0.0000000041359594625716465	0.000000000011346841945152773
101250124	Habib Abdullah	Jempol Kanan	999.922.762.978.962	0.000000008326956101531565	0.00000007782729973772594	0.0000000002548799239006183
101270124	Habib Abdullah	Jempol Kanan	9.996.105.466.457.760	0.0000026419784860747117	0.00000010342826750418155	0.00000000000019126789005891977
101290124	Habib Abdullah	Jempol Kanan	9.997.597.405.811.610	0.0000020528119615439533	0.00000005680196802516182	0.00000000004812396358604555
101300124	Habib Abdullah	Jempol Kanan	10.003.941.291.324.600	0.0000016519527911519808	0.0000004923848738270323	0.00000000027334145343886366
101310124	Habib Abdullah	Jempol Kiri	10.001.778.933.163.900	0.0000006187887632127819	0.000000018405446454511128	0.000000001080497472640761
101320124	Habib Abdullah	Jempol Kiri	9.998.682.689.779.670	0.0000037931656709956937	0.0000016685726151240495	0.000000000007862975313204805
101330124	Habib Abdullah	Jempol Kiri	9.999.043.658.911.290	0.00000000434917893730648	0.00000030712532640908696	0.00000000010117560010671432
101350124	Habib Abdullah	Jempol Kiri	9.998.872.103.078.220	0.00000016157189556507537	0.000000026239974651870846	2,80E+00
101370124	Habib Abdullah	Jempol Kiri	10.001.030.735.933.900	0.00000014108288730050972	0.000000018282502177359255	0.00000000020537790704792434
101380124	Habib Abdullah	Jempol Kiri	9.997.493.954.240.410	0.00000383510615971412	0.0000004973652967703656	1,11E-07
101390124	Habib Abdullah	Jempol Kiri	9.995.968.505.256.710	0.000003828777576215249	0.000000005721253511246767	0.00000000026944018291995147
101411197	M. Miftahul Huda	Jempol Kanan	10.000.576.070.528.200	0.00000005947698519349419	0.0000000030969414320905412	0.00000000013220612202450148

101431197	M. Miftahul Huda	Jempol Kanan	9.998.296.291.991.940	0.0000003893349674056487	0.0000000519082220501656	0.000000000000007994102657449332
101441197	M. Miftahul Huda	Jempol Kanan	9.997.508.102.369.290	0.0000006596979764082101	0.00000016695150316018327	0.0000000006356204558056254
101451197	M. Miftahul Huda	Jempol Kanan	10.004.344.899.702.300	0.000005206272817499755	0.000000011920551012559709	0.00000000000010932308957659732
101461197	M. Miftahul Huda	Jempol Kanan	10.003.903.114.725.100	0.000002518589919362544	0.00000013295424086060907	0.00000000000007210748249414835
101471197	M. Miftahul Huda	Jempol Kanan	9.999.594.833.233.680	0.0000000030919415938555693	0.0000000666241955194513	5,58E-23
101481197	M. Miftahul Huda	Jempol Kanan	9.997.886.152.936.400	0.00000148138766426387	0.000000026635378418013977	0.00000000000936470711543708
101501197	M. Miftahul Huda	Jempol Kanan	9.999.940.913.149.550	0.00000006027829015755445	0.00000004720421029654245	0.0000000000016702814232672172
101521197	M. Miftahul Huda	Jempol Kiri	1.000.009.259.018.270	0.0000008026939173280193	0.0000008879981173137117	0.000000000000012026426369517242
101531197	M. Miftahul Huda	Jempol Kiri	10.000.999.366.091.400	0.000000009995874786597484	0.0000002208294702859977	0.000000000003499344807367915
101541197	M. Miftahul Huda	Jempol Kiri	10.000.293.926.993.300	0.00000000008925613317694847	0.000000022485000486116628	2,35E-27
101551197	M. Miftahul Huda	Jempol Kiri	9.997.084.783.471.630	0.00000007194113939014104	0.0000014159803696005526	0.00000000000020582364376894544
101561197	M. Miftahul Huda	Jempol Kiri	9.998.354.561.050.650	0.0000007487020193671875	0.0000000021665457156933847	0.000000000016190955117850283
101581197	M. Miftahul Huda	Jempol Kiri	10.001.107.952.821.200	0.000000308813912222356	0.000000000030597364634954016	2,13E-23
101591197	M. Miftahul Huda	Jempol Kiri	10.001.166.101.692.000	0.00000000017654367075403068	0.0000003556404706718429	0.000000000000012369237693529344
101610010	Sholikhatin Nasliyana	Jempol Kanan	1.000.228.787.894.400	0.0000000010086875043053797	0.0000012370681736769162	0.000000000000014751579250066067
101620010	Sholikhatin Nasliyana	Jempol Kanan	9.998.303.137.990.420	0.0000003433028326544918	0.0000000690694077443679	0.0000000000016506442081943968
101640010	Sholikhatin Nasliyana	Jempol Kanan	10.001.779.236.056.300	0.00000012376484930508814	0.000000289150887615239	0.000000000000023669916554904454
101650010	Sholikhatin Nasliyana	Jempol Kanan	10.002.270.529.746.200	0.00000002122464596476061	0.0000004691809703615881	0.0000000000027286961949825394
101660010	Sholikhatin Nasliyana	Jempol Kanan	10.000.841.330.101.600	0.00000016333903982737984	0.0000006785137758209483	0.0000000000011858668871553101
101670010	Sholikhatin Nasliyana	Jempol Kanan	9.997.985.584.588.560	0.0000027441999320365534	0.00000040973735542338487	0.000000000008458392037353891
101680010	Sholikhatin Nasliyana	Jempol Kanan	9.998.924.616.314.310	0.000000010251878594887396	0.00000040826714710604637	5,63E-23
101690010	Sholikhatin Nasliyana	Jemol Kanan	10.002.941.392.932.500	0.0000018951113109444375	0.000000008845591797824407	0.000000000000008665884326282449
101700010	Sholikhatin Nasliyana	Jempol Kanan	9.998.892.316.695.480	0.00000016052676611283891	0.000000023467479340797014	6,53E-02
101710010	Sholikhatin Nasliyana	Jempol Kiri	10.002.734.996.834.300	0.000008392149037988121	0.0000023465390045714126	0.0000000000054103590703593

101720010	Sholikhatin Nasliyana	Jempol Kiri	10.003.987.162.775.500	0.0000021891438386392968	0.0000002621307213471104	0.0000000000033128234094927473
101730010	Sholikhatin Nasliyana	Jempol Kiri	10.001.608.580.486.500	0.0000026144714836702632	0.0000006494141977107514	0.00000000004640939126410018
101740010	Sholikhatin Nasliyana	Jempol Kiri	10.000.626.578.742.800	0.0000000015771213653583061	0.00000007483551305645848	1,14E-01
101750010	Sholikhatin Nasliyana	Jempol Kiri	10.001.865.837.267.100	0.00000034022434866081705	0.00000010239893248182909	0.0000000008755239315629907
101760010	Sholikhatin Nasliyana	Jempol Kiri	9.995.590.350.361.530	0.000003743226624995734	0.00000006996158925030968	0.000000000033911976468518627
101770010	Sholikhatin Nasliyana	Jempol Kiri	9.996.207.068.222.970	0.0000020485291645718335	0.00000021648449489576504	0.00000000000009688959141051841
101780010	Sholikhatin Nasliyana	Jempol Kiri	10.003.434.773.807.200	0.000001084065778128654	0.00000045622084870868	0.00000000000037630550807658965
101790010	Sholikhatin Nasliyana	Jempol Kiri	9.998.360.009.869.320	0.00000003245305619768004	0.0000004093756197069975	0.000000000000003213842826904192
101810148	Prima Oktava W	Jempol Kanan	9.998.271.712.219.180	0.00000037581098535316954	0.00000005372156329187859	0.0000000003751139983015913
101820148	Prima Oktava W	Jempol Kanan	10.003.983.279.461.400	0.000005660286660755392	0.00000014853104101071188	0.0000000000052762387570138294
101840148	Prima Oktava W	Jempol Kanan	9.999.342.886.771.370	0.00000045040368748846097	0.00000010877515556671993	0.0000000001625221411957297
101850148	Prima Oktava W	Jempol Kanan	1.000.768.626.221.690	0.000014999855383714858	0.0000000010090966492895591	0.0000000000014268653295082186
101860148	Prima Oktava W	Jempol Kanan	10.000.338.329.327.100	0.00000010600701376404503	0.00000002216705642115811	0.00000000005681991381041405
101870148	Prima Oktava W	Jempol Kanan	10.000.840.264.378.700	0.00000010241152511398783	0.000000010847988421990152	0.00000000001638889656983153
101880148	Prima Oktava W	Jempol Kanan	10.001.776.345.384.700	0.0000022721962897519154	0.0000004232113303852616	0.0000000009796646440566167
101890148	Prima Oktava W	Jempol Kanan	9.996.991.496.417.390	0.0000016901271100049446	0.00000004074341249858291	0.000000000006081571428027846
101910148	Prima Oktava W	Jempol Kiri	9.998.365.049.242.560	0.0000013847513814095173	0.00000012825546146319969	0.0000000000012501151719176128
101920148	Prima Oktava W	Jempol Kiri	9.997.009.435.697.190	0.0000000007878686806849645	0.0000022477337197337954	0.000000000586539481784661
101940148	Prima Oktava W	Jempol Kiri	99.960.134.349.056	0.000003697568281898767	0.000000003385654665538088	0.00000000015347888732284842
101950148	Prima Oktava W	Jempol Kiri	10.001.454.988.437.100	0.0000009474208785906921	0.00000006039740420076378	0.000000000000016610675112284205
101960148	Prima Oktava W	Jempol Kiri	9.993.935.479.973.830	0.000005238897848946108	0.0000005309694005139458	0.00000000023041823656729544
101970148	Prima Oktava W	Jempol Kiri	9.990.483.568.328.740	0.000009610306624586178	0.000002706578921421926	0.0000000001996736796467856
101980148	Prima Oktava W	Jempol Kiri	10.001.968.917.693.700	0.0000031937884203085997	0.0000006477504484356476	0.000000000004504995016912794
102010020	Shanti Sanata Islam	Jempol Kanan	1.000.340.970.029.190	0.000003489946768931376	0.000000027699237972477953	0.0000000000089748373314499

102020020	Shanti Sanata Islam	Jempol Kanan	9.999.796.949.877.980	0.00000007594264736245974	0.000000031605758268791845	0.00000000013903389500622854
102030020	Shanti Sanata Islam	Jempol Kanan	1.000.045.063.470.000	0.000000001077463906100928	0.00000004042799151850201	0.00000000007356605258222178
102040020	Shanti Sanata Islam	Jempol Kanan	9.994.829.286.265.450	0.0000047093975888000695	0.0000001701298040846963	0.000000000009663178497579115
102050020	Shanti Sanata Islam	Jempol Kanan	9.997.518.263.835.560	0.0000049021854901632494	0.0000009464293360087633	0.0000000000000881385941320698
102060020	Shanti Sanata Islam	Jempol Kanan	10.005.180.150.357.000	0.0000034361648688640844	0.0000005298019850126281	0.00000000006683607342560666
102070020	Shanti Sanata Islam	Jempol Kanan	10.004.069.049.241.100	0.0000026133174778173303	0.00000017402074649155482	0.00000000000034094968608024116
102080020	Shanti Sanata Islam	Jempol Kanan	10.000.358.670.046.500	0.000001997706519745217	0.0000015475134818927696	0.00000000009844414827438198
102090020	Shanti Sanata Islam	Jempol Kanan	10.005.855.004.232.500	0.000007764927010971753	0.00000020516369455127493	0.000000000007417439505190942
102110020	Shanti Sanata Islam	Jempol Kiri	10.005.329.617.143.900	0.0000023564994010307554	0.0000012578987361814121	0.00000000006089272833324171
102120020	Shanti Sanata Islam	Jempol Kiri	1.000.028.797.209.830	0.000000050490952767009294	0.00000013657120578662072	0.0000000000007542644281323221
102130020	Shanti Sanata Islam	Jempol Kiri	9.997.243.757.821.300	0.0000003899907092927973	0.0000005690047585939877	0.0000000000003604066868885187
102140020	Shanti Sanata Islam	Jempol Kiri	1.000.136.470.095.700	0.0000002368427513658368	0.00000003884163248828541	0.000000000002017915057717566
102160020	Shanti Sanata Islam	Jempol Kiri	9.998.808.610.961.570	0.00000003540349876119142	0.0000001661592731240182	0.000000000000005227451712071633
102170020	Shanti Sanata Islam	Jempol Kiri	9.999.683.940.839.080	0.0000001560697676953742	0.0000003302940876964933	0.000000000467627005016387
102180020	Shanti Sanata Islam	Jempol Kiri	9.998.750.434.254.880	0.0000009484059202099845	0.0000026288760756187233	0.0000000005160702366330664
102190020	Shanti Sanata Islam	Jempol Kiri	9.998.848.200.426.420	0.0000005724730187629376	0.00000003209339472933573	0.0000000000024236344151398645
102210191	Nur Khodijah Purna Sari	Jempol Kanan	10.009.002.930.909.300	0.000010878724508625365	0.0000015563014601991851	0.000000002057213816861895
102220191	Nur Khodijah Purna Sari	Jempol Kanan	10.002.336.667.354.000	0.0000003468695617153592	0.00000034084541587726056	0.000000000020345289612808007
102240191	Nur Khodijah Purna Sari	Jempol Kanan	10.000.191.243.825.800	0.00000003739752386930299	0.00000008387651975669936	0.00000000000037037339946531337
102250191	Nur Khodijah Purna Sari	Jempol Kanan	10.001.061.803.326.300	0.000000043577861303371435	0.0000005509909377555883	0.00000000007006376910048974
102260191	Nur Khodijah Purna Sari	Jempol Kanan	9.995.850.824.524.760	0.0000004888057893671003	0.000001892319457367542	0.0000000000000016384384112798246
102280191	Nur Khodijah Purna Sari	Jempol Kanan	10.001.574.809.815.900	0.00000036835051545075695	0.00000003707694243012348	0.00000000014637853044591738
102290191	Nur Khodijah Purna Sari	Jempol Kanan	9.998.202.729.090.990	0.00000006455189920309764	0.000001330078514925766	0.00000000000029572517432216213
102300191	Nur Khodijah Purna Sari	Jempol Kanan	10.002.189.634.564.000	0.000003344026108680552	0.0000005392257665417838	0.00000000000016764201390102043

102310191	Nur Khodijah Purna Sari	Jempol Kiri	9.996.725.977.558.050	0.0000006412925949258246	0.0000007022390764375199	0.00000000002753254723049022
102320191	Nur Khodijah Purna Sari	Jempol Kiri	9.998.290.975.541.460	0.0000000966052711923056	0.0000013912612042215754	0.00000000020035648419055194
102330191	Nur Khodijah Purna Sari	Jempol Kiri	9.998.488.110.088.530	0.00000007998219989911378	0.00000022404876574496028	0.00000000000003083951377474171
102340191	Nur Khodijah Purna Sari	Jempol Kiri	9.994.839.462.427.660	0.0000013447022040238827	0.000001985258763656434	0.00000000014392075999838395
102360191	Nur Khodijah Purna Sari	Jempol Kiri	9.999.365.522.603.540	0.00000010066534779276452	0.0000000000000011764051423983312	2,58E-24
102370191	Nur Khodijah Purna Sari	Jempol Kiri	10.000.555.701.542.000	0.0000009522492908881184	0.0000004852687033260418	0.0000000000018862226367429705
102390191	Nur Khodijah Purna Sari	Jempol Kiri	9.998.474.783.677.670	0.0000010451111414724477	0.00000006728924832027077	0.00000000000007172778962529966
102400191	Nur Khodijah Purna Sari	Jempol Kiri	10.000.377.598.654.500	0.0000010031663331830147	0.0000014184556579694923	0.00000000000003719787101586603
102410122	M. Syauqil Ilmi	Jempol Kanan	10.005.377.959.770.800	0.000010471622031951956	0.00000029660763558202165	0.000000000007579872470455274
102420122	M. Syauqil Ilmi	Jempol Kanan	9.992.561.415.553.770	0.000004547409656668778	0.0000025200432265980916	0.00000000000002927572364472164
102430122	M. Syauqil Ilmi	Jempol Kanan	10.000.053.225.934.800	0.00000027251117390880586	0.000000241652644102383	0.00000000001467900044989307
102440122	M. Syauqil Ilmi	Jempol Kanan	9.993.856.464.562.090	0.000011128019225255735	0.00000005848105358866998	0.0000000004752562262757514
102450122	M. Syauqil Ilmi	Jempol Kanan	999.739.872.492.389	0.0000012743521122544578	0.000000028402341904511512	0.000000000011023297890935808
102460122	M. Syauqil Ilmi	Jempol Kanan	9.995.672.095.741.800	0.0000020694936560384122	0.0000005216071977551109	0.000000000011485647680671255
102470122	M. Syauqil Ilmi	Jempol Kanan	9.997.775.164.403.650	0.000000743460299784034	0.00000006301868178875254	0.00000000000006351785093401929
102480122	M. Syauqil Ilmi	Jempol Kanan	10.003.293.333.989.800	0.000004916305708146906	0.0000003215901212151304	0.000000000013350111512568285
102490122	M. Syauqil Ilmi	Jempol Kanan	10.003.581.850.957.600	0.000003736959525466417	0.00000002026150640240603	1,86E-01
102510122	M. Syauqil Ilmi	Jempol Kiri	10.000.733.624.013.400	0.0000000496881470395575	0.00000001717769959418034	0.0000000001646903389494205
102520122	M. Syauqil Ilmi	Jempol Kiri	9.999.844.956.887.900	0.0000001966204689177247	0.0000002774708738960649	0.000000000033803970215792493
102530122	M. Syauqil Ilmi	Jempol Kiri	100.033.523.435.921	0.000005296284834617097	0.0000003797769061838077	0.00000000008233929236322576
102540122	M. Syauqil Ilmi	Jempol Kiri	1.000.239.637.679.680	0.0000001332220513525783	0.0000006422347445064762	0.000000001006486116836442
102550122	M. Syauqil Ilmi	Jempol Kiri	9.999.639.174.352.360	0.00000017454623426617882	0.00000035777711182353686	0.000000000000032528681196849553
102560122	M. Syauqil Ilmi	Jempol Kiri	10.000.286.445.802.100	0.0000003672533049728725	0.0000005634254732337281	0.000000000001905988133280294
102580122	M. Syauqil Ilmi	Jempol Kiri	10.000.979.492.588.000	0.000000131880805648189	0.000000014962451333993396	0.000000000018141066633569216

102590122	M. Syaquil Ilmi	Jempol Kiri	9.997.640.756.502.960	0.000001191694944092453	0.000000005819240276890192	0.0000000001382876192301603
102600122	M. Syaquil Ilmi	Jempol Kiri	9.995.908.058.778.410	0.000003977566505453151	0.0000000016515804905196048	0.00000000012446907156431776
102610135	M. Agung Tarecha	Jempol Kanan	9.999.893.635.968.040	0.00000040461398123788985	0.0000004634023260161274	0.00000000007290252788697434
102620135	M. Agung Tarecha	Jempol Kanan	10.001.398.839.829.100	0.0000008514187695616796	0.00000005339009263198998	0.00000000005982803798658409
102630135	M. Agung Tarecha	Jempol Kanan	10.002.846.736.556.700	0.0000008143865488599309	0.00000027114468653902496	0.000000000000013867506801929135
102640135	M. Agung Tarecha	Jempo Kanan	10.000.908.302.721.200	0.00000011347274147542048	0.000000013755408920266295	1,94E-03
102650135	M. Agung Tarecha	Jempol Kanan	9.996.802.820.525.380	0.000002978561535634678	0.00000001779606238495262	0.00000000003929353223690383
102670135	M. Agung Tarecha	Jempol Kanan	9.997.190.693.602.990	0.000003487672045001261	0.0000002125831434802773	0.0000000000029300597093449534
102680135	M. Agung Tarecha	Jempol Kanan	9.998.364.106.761.730	0.0000012888356562754778	0.00000010286632923896055	0.000000000011826072843500361
102690135	M. Agung Tarecha	Jempol Kanan	9.996.829.830.260.350	0.00000008561805122561365	0.000001645941724418072	0.00000000009335978389462861
102700135	M. Agung Tarecha	Jempol Kanan	10.000.501.420.773.400	0.000000014579185793498427	0.000000015424851591064126	0.000000000033248126030002324
102710135	M. Agung Tarecha	Jempol Kiri	9.996.639.398.360.710	0.00000019013144285638296	0.000001532845035496054	0.00000000003994466676500619
102720135	M. Agung Tarecha	Jempol Kiri	10.001.977.079.390.900	0.0000031803448537781965	0.0000006315577344714716	0.000000000000024096921865270173
102730135	M. Agung Tarecha	Jempol Kiri	9.997.283.954.285.290	0.0000016400948699218563	0.000000010603000354176642	0.0000000006511518720715209
102740135	M. Agung Tarecha	Jempol Kiri	10.001.772.351.282.600	0.000000616391250901174	0.00000001020672131625588	4,73E-03
102760135	M. Agung Tarecha	Jempol Kiri	9.998.879.167.614.760	0.00000015988360479817002	0.000000023087285944742713	0.00000000007451846232174892
102770135	M. Agung Tarecha	Jempol Kiri	1.000.054.809.303.040	0.00000004816382118572582	0.0000002436692116057097	0.000000000000015198767350709426
102780135	M. Agung Tarecha	Jempol Kiri	100.009.593.841.404	0.0000001721820912066859	0.0000000041902662730173075	2,34E-25
102790135	M. Agung Tarecha	Jempol Kiri	9.999.674.227.311.780	0.0000000013578441341127914	0.00000004069584537950925	0.000000000003982042234075885
102810119	Tino Arif Cahya	Jempol Kanan	9.999.501.115.102.940	0.0000001357144365326852	0.000000010613540474185373	0.00000000025366311877416747
102820119	Tino Arif Cahya	Jempol Kanan	9.994.441.985.654.900	0.000003916012941541429	0.0000006396133850795121	0.0000000000005524174640376188
102830119	Tino Arif Cahya	Jempol Kanan	10.003.161.561.146.900	0.0000001696200015349435	0.0000013661861957646742	0.0000000000000011234788123692215
102840119	Tino Arif Cahya	Jempol Kanan	9.996.423.641.053.380	0.0000019456675650108474	0.00000015478281692566229	0.000000000000001851214632214075
102850119	Tino Arif Cahya	Jempol Kanan	9.998.932.951.751.190	0.00000063349317593013	0.00000006894024959528546	0.00000000000003249555786148409

102870119	Tino Arif Cahya	Jempol Kanan	999.919.394.023.379	0.000000475489857255295	0.00000007644338589869253	0.00000000010071628464423963
102880119	Tino Arif Cahya	Jempol Kanan	9.998.563.617.864.470	0.0000005750033219098724	0.000000001988546900971027	0.000000000020432677252219022
102890119	Tino Arif Cahya	Jempol Kanan	10.001.033.149.602.700	0.0000002565883669241741	0.00000000010150357854498781	0.0000000000000034765062624692424
102900119	Tino Arif Cahya	Jempol Kanan	9.991.874.650.712.780	0.000017011368547544168	0.000000007680387608386133	0.0000000007104374148594459
102910119	Tino Arif Cahya	Jempol Kiri	9.998.779.249.189.770	0.00000005036003592774784	0.00000015055769845515292	0.0000000000041373048340310744
102920119	Tino Arif Cahya	Jempol Kiri	10.004.943.849.627.000	0.0000070747003201740234	0.000000042041482967111707	0.0000000002832046379599472
102930119	Tino Arif Cahya	Jempol Kiri	10.000.352.246.392.400	0.00000031989996324513946	0.0000001515320481734372	0.0000000000000416879573542651
102940119	Tino Arif Cahya	Jempol Kiri	10.001.984.418.340.900	0.0000004811613183127057	0.0000000908564864927109	0.0000000000008538504812228878
102950119	Tino Arif Cahya	Jempol Kiri	9.998.380.313.115.870	0.0000007734022317073695	0.000000004827099012228914	0.0000000000000062946939867323115
102960119	Tino Arif Cahya	Jempol Kiri	9.999.387.277.384.530	0.0000009291867420368091	0.00000041727287485216165	0.00000000013480853998352394
102970119	Tino Arif Cahya	Jempol Kiri	9.999.216.609.042.660	0.00000026803526623141103	0.000000018829823639222407	0.00000000012552357335573813
102980119	Tino Arif Cahya	Jempol Kiri	1.000.043.815.947.410	0.0000001476368095234069	0.0000000254187491564621	0.00000000003278060164493118
102990199	Tino Arif Cahya	Jempol Kiri	10.002.442.032.516.800	0.0000007502955608002397	0.00000012586100740796425	5,41E-02
103010057	Ahmad Farid Rahman	Jempol Kanan	10.000.597.781.620.300	0.00000045098736388169487	0.000000139014519631129	0.00000000000003130259401240866
103020057	Ahmad Farid Rahman	Jempol Kanan	10.004.501.111.484.200	0.000005906935525061781	0.00000003242759449820072	0.000000000000001569120618180087
103030057	Ahmad Farid Rahman	Jempol Kanan	10.000.073.051.282.700	0.000000005022810249234083	0.000000011317115984487986	0.0000000000010310932150590705
103040057	Ahmad Farid Rahman	Jempol Kanan	10.000.520.879.333.200	0.00000008286699753240266	0.0000000007524053857423994	5,07E-25
103050057	Ahmad Farid Rahman	Jempol Kanan	9.998.357.632.461.830	0.000000012282360939581343	0.0000005046563326015176	2,00E-23
103060057	Ahmad Farid Rahman	Jempol Kanan	9.995.813.903.042.970	0.0000027729924674722076	0.00000018321576942184668	4,35E-22
103070057	Ahmad Farid Rahman	Jempol Kanan	10.003.053.680.904.200	0.0000027105422224311938	0.000000014332136523064693	0.000000000000004673908599210139
103080057	Ahmad Farid Rahman	Jempol Kanan	999.900.084.549.236	0.0000015062357360892132	0.0000005295447087222084	5,95E-23
103090057	Ahmad Farid Rahman	Jempol Kanan	9.998.492.228.963.220	0.00000018062422617558166	0.00000010818497728982743	4,58E-25

LAMPIRAN 2 Listing *Source Code* Proses *Similarity Search* menggunakan
NOHIS-Search

```
% Load database
conn= database('db_cbir','root','','
'com.mysql.jdbc.Driver','jdbc:mysql://localhost/db_cbir');
setdbprefs('DataReturnFormat','cellarray');
ArrayDB=fetch(conn, 'select * from data');

ID = cell2mat(ArrayDB(:,1));
APP= cell2mat(ArrayDB(:,5));
HOR= cell2mat(ArrayDB(:,6));
VER= cell2mat(ArrayDB(:,7));
DIAG= cell2mat(ArrayDB(:,8));

NewArray = [ID HOR VER];

load apr_t
load hor_t
load ver_t
load diag_t

% Proses Similarity Search pada Diagram NOHIS-Tree
n1K =NewArray;
[n1k1,n1k2,n1k3,n1k4]=metode_nohis(NewArray,true);
[n1K,kuadran]=metode_nohis_2(n1K,false,hor_total,ver_total);

ketemu = false;
indexFig = 2;
figure(indexFig)
SubKuadran_Ke = 3;
SubKuadran_Ke = kuadran
grid on
hold on
plot(n1K(:,2),n1K(:,3),'b.')
plot(hor_total,ver_total,'ro')

while ~ketemu
    [n1K,kuadran]=metode_nohis_2(n1K,false,hor_total,ver_total);
    indexFig = indexFig+1;
    figure(indexFig)
    SubKuadran_Ke = SubKuadran_Ke+1;
    SubKuadran_Ke = kuadran
    grid on
    hold on
    plot(n1K(:,2),n1K(:,3),'b.')
    plot(hor_total,ver_total,'ro')
    if size(n1K,1) < 20 %batas
        size(n1K,1);
        ketemu=true;
    end
end
end
save nnq n1K;
```

```

% --- Seleksi Hasil --- %
EuclideanSorting = zeros(1,4);
for i=1:size(n1K,1)
    HasilEuclideanMinimum = abs(sqrt ( ((hor_total-
n1K(i,2))^2) + ((ver_total-n1K(i,3))^2) ));
    EuclideanSorting(i,1)=n1K(i,1);
    EuclideanSorting(i,2)=n1K(i,2);
    EuclideanSorting(i,3)=n1K(i,3);
    EuclideanSorting(i,4)=HasilEuclideanMinimum;
end

for i=1:size(EuclideanSorting,1)
    for j=i+1:size(EuclideanSorting,1)
        if EuclideanSorting(i,4) > EuclideanSorting(j,4)
            temp = EuclideanSorting(i,:);
            EuclideanSorting(i,:) = EuclideanSorting(j,:);
            EuclideanSorting(j,:) = temp;
        end
    end
end

save EucSort EuclideanSorting;

% --- Menempatkan 9 Terkecil --- %
conn= database('db_cbir','root','','
'com.mysql.jdbc.Driver','jdbc:mysql://localhost/db_cbir');
setdbprefs('DataReturnFormat','cellarray');

if size(EuclideanSorting,1) > 0
    perintah='select * from data where id=';
    variabel=int2str(EuclideanSorting(1));
    sql= strcat(perintah,variabel);
    ArrayDBhasil=fetch(conn, sql);
    handles.data1 = imread(char(ArrayDBhasil(1,4)));
    guidata(hObject,handles);
    axes(handles.ax_h1);
    imshow(handles.data1);
    set(handles.filehasil1, 'String', ArrayDBhasil(1,4));
    set(handles.id_1, 'String', variabel);
    set(handles.nm_1, 'String', ArrayDBhasil(1,2));
    set(handles.ket_1, 'String', ArrayDBhasil(1,3));
    set(handles.Hor_1, 'String', ArrayDBhasil(1,6));
    set(handles.Ver_1, 'String', ArrayDBhasil(1,7));
    set(handles.euc_1, 'String', EuclideanSorting(1,4));
    set(handles.uipanel01,'Visible','on');
else
    set(handles.uipanel01,'Visible','off');
end

if size(EuclideanSorting,1) > 1
    perintah='select * from data where id=';

```

```

variabel=int2str(EuclideanSorting(2));
sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h2);
imshow(handles.data1);
set(handles.filehasil2, 'String', ArrayDBhasil(1,4));
set(handles.id_2, 'String', variabel);
set(handles.nm_2, 'String', ArrayDBhasil(1,2));
set(handles.ket_2, 'String', ArrayDBhasil(1,3));
set(handles.Hor_2, 'String', ArrayDBhasil(1,6));
set(handles.Ver_2, 'String', ArrayDBhasil(1,7));
set(handles.euc_2, 'String', EuclideanSorting(2,4));
set(handles.uipanel02,'Visible','on');
else
set(handles.uipanel02,'Visible','off');
end

if size(EuclideanSorting,1) > 2
perintah='select * from data where id=';
variabel=int2str(EuclideanSorting(3));
sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h3);
imshow(handles.data1);
set(handles.filehasil3, 'String', ArrayDBhasil(1,4));
set(handles.id_3, 'String', variabel);
set(handles.nm_3, 'String', ArrayDBhasil(1,2));
set(handles.ket_3, 'String', ArrayDBhasil(1,3));
set(handles.Hor_3, 'String', ArrayDBhasil(1,6));
set(handles.Ver_3, 'String', ArrayDBhasil(1,7));
set(handles.euc_3, 'String', EuclideanSorting(3,4));
set(handles.uipanel03,'Visible','on');
else
set(handles.uipanel03,'Visible','off');
end

if size(EuclideanSorting,1) > 3
perintah='select * from data where id=';
variabel=int2str(EuclideanSorting(4));
sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h4);
imshow(handles.data1);
set(handles.filehasil4, 'String', ArrayDBhasil(1,4));
set(handles.id_4, 'String', variabel);
set(handles.nm_4, 'String', ArrayDBhasil(1,2));
set(handles.ket_4, 'String', ArrayDBhasil(1,3));

```

```

set(handles.Hor_4, 'String', ArrayDBhasil(1,6));
set(handles.Ver_4, 'String', ArrayDBhasil(1,7));
set(handles.euc_4, 'String', EuclideanSorting(4,4));
set(handles.uipanel04,'Visible','on');
else
set(handles.uipanel04,'Visible','off');
end

if size(EuclideanSorting,1) > 4
perintah='select * from data where id=';
variabel=int2str(EuclideanSorting(5));
sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h5);
imshow(handles.data1);
set(handles.id_5, 'String', variabel);
set(handles.nm_5, 'String', ArrayDBhasil(1,2));
set(handles.ket_5, 'String', ArrayDBhasil(1,3));
set(handles.filehasil5, 'String', ArrayDBhasil(1,4));
set(handles.Hor_5, 'String', ArrayDBhasil(1,6));
set(handles.Ver_5, 'String', ArrayDBhasil(1,7));
set(handles.euc_5, 'String', EuclideanSorting(5,4));
set(handles.uipanel05,'Visible','on');
else
set(handles.uipanel05,'Visible','off');
end

if size(EuclideanSorting,1) > 5
perintah='select * from data where id=';
variabel=int2str(EuclideanSorting(6));
sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h6);
imshow(handles.data1);
set(handles.filehasil6, 'String', ArrayDBhasil(1,4));
set(handles.id_6, 'String', variabel);
set(handles.nm_6, 'String', ArrayDBhasil(1,2));
set(handles.ket_6, 'String', ArrayDBhasil(1,3));
set(handles.Hor_6, 'String', ArrayDBhasil(1,6));
set(handles.Ver_6, 'String', ArrayDBhasil(1,7));
set(handles.euc_6, 'String', EuclideanSorting(6,4));
set(handles.uipanel06,'Visible','on');
else
set(handles.uipanel06,'Visible','off');
end

if size(EuclideanSorting,1) > 6
perintah='select * from data where id=';
variabel=int2str(EuclideanSorting(7));

```

```

sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h7);
imshow(handles.data1);
set(handles.filehasil7, 'String', ArrayDBhasil(1,4));
set(handles.id_7, 'String', variabel);
set(handles.nm_7, 'String', ArrayDBhasil(1,2));
set(handles.ket_7, 'String', ArrayDBhasil(1,3));
set(handles.Hor_7, 'String', ArrayDBhasil(1,6));
set(handles.Ver_7, 'String', ArrayDBhasil(1,7));
set(handles.euc_7, 'String', EuclideanSorting(7,4));
set(handles.uipanel07,'Visible','on');
else
set(handles.uipanel07,'Visible','off');
end

if size(EuclideanSorting,1) > 7
perintah='select * from data where id=';
variabel=int2str(EuclideanSorting(8));
sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h8);
imshow(handles.data1);
set(handles.filehasil8, 'String', ArrayDBhasil(1,4));
set(handles.id_8, 'String', variabel);
set(handles.nm_8, 'String', ArrayDBhasil(1,2));
set(handles.ket_8, 'String', ArrayDBhasil(1,3));
set(handles.Hor_8, 'String', ArrayDBhasil(1,6));
set(handles.Ver_8, 'String', ArrayDBhasil(1,7));
set(handles.euc_8, 'String', EuclideanSorting(8,4));
set(handles.uipanel08,'Visible','on');
else
set(handles.uipanel08,'Visible','off');
end

if size(EuclideanSorting,1) > 8
perintah='select * from data where id=';
variabel=int2str(EuclideanSorting(9));
sql= strcat(perintah,variabel);
ArrayDBhasil=fetch(conn, sql);
handles.data1 = imread(char(ArrayDBhasil(1,4)));
guidata(hObject,handles);
axes(handles.ax_h9);
imshow(handles.data1);
set(handles.filehasil9, 'String', ArrayDBhasil(1,4));
set(handles.id_9, 'String', variabel);
set(handles.nm_9, 'String', ArrayDBhasil(1,2));
set(handles.ket_9, 'String', ArrayDBhasil(1,3));
set(handles.Hor_9, 'String', ArrayDBhasil(1,6));

```

```
set(handles.Ver_9, 'String', ArrayDBhasil(1,7));  
set(handles.euc_9, 'String', EuclideanSorting(9,4));  
set(handles.uipanel09,'Visible','on');  
else  
set(handles.uipanel09,'Visible','off');  
end
```



LAMPIRAN 3 Listing *Source Code Visualisasi Ranking Identifikasi*

```

% % ---- Background Axes Input ---- %
handles.output = hObject;
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide00.png');
guidata(hObject,handles);
axes(handles.ax_1);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide00.png');
guidata(hObject,handles);
axes(handles.ax_2);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide00.png');
guidata(hObject,handles);
axes(handles.ax_3);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide00.png');
guidata(hObject,handles);
axes(handles.ax_4);
imshow(handles.data1);
% ---- Background Axes Similarity Search ---- %
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide1.png');
guidata(hObject,handles);
axes(handles.ax_h1);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide2.png');
guidata(hObject,handles);
axes(handles.ax_h2);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide3.png');
guidata(hObject,handles);
axes(handles.ax_h3);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide4.png');
guidata(hObject,handles);
axes(handles.ax_h4);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide5.png');
guidata(hObject,handles);
axes(handles.ax_h5);
imshow(handles.data1);
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-
CBIR\Symbol CBIR\Slide6.png');

```

```
guidata(hObject,handles);  
axes(handles.ax_h6);  
imshow(handles.data1);  
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-  
CBIR\Symbol CBIR\Slide7.png');  
guidata(hObject,handles);  
axes(handles.ax_h7);  
imshow(handles.data1);  
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-  
CBIR\Symbol CBIR\Slide8.png');  
guidata(hObject,handles);  
axes(handles.ax_h8);  
imshow(handles.data1);  
handles.data1 = imread('F:\Skripsi\Skripsi - 09650145-  
CBIR\Symbol CBIR\Slide9.png');  
guidata(hObject,handles);  
axes(handles.ax_h9);  
imshow(handles.data1);
```