

**APLIKASI PENDETEKSIAN PENYAKIT PERIODONTITIS  
KRONIS CITRA PANORAMIC PADA GIGI MANUSIA BERBASIS  
MATLAB MENGGUNAKAN ALGORITMA LINE STRENGHT  
DAN LINE TRACKING**

**SKRIPSI**



**Disusun Oleh :**

**KADEK SUARJUNA BATU BULAN  
09.18.117**

**JURUSAN TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2013**

RECEIVED  
MAY 19 1964  
U.S. DEPARTMENT OF AGRICULTURE  
WASHINGTON, D.C.

RECEIVED

RECEIVED

RECEIVED  
MAY 19 1964

RECEIVED  
MAY 19 1964  
U.S. DEPARTMENT OF AGRICULTURE  
WASHINGTON, D.C.



**LEMBAR PERSETUJUAN**

**APLIKASI PENDETEKSIAN PENYAKIT PERIODONTITIS KRONIS CITRA PANORAMIC PADA GIGI MANUSIA BERBASIS MATLAB MENGGUNAKAN ALGORITMA LINE STRENGHT DAN LINE TRACKING**

**SKRIPSI**


*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai gelar Sarjana Komputer*

**Disusun Oleh :**  
**KADEK SUARJUNA BATU BULAN**  
**09.18.117**

**Diperiksa dan Disetujui,**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

  
**Ir. Yusuf Ismail Nakhoda, MT**

**NIP.Y. 1018800189**

  
**Suryo Adi Wibowo, ST**

**NIP.P. 1031000438**

**Mengetahui**

**Kepala Jurusan Teknik Informatika S-1**

  
**Joseph Dedy Irawan, ST, MT**

**NIP. 197404162005011002**

**JURUSAN TEKNIK INFORMATIKA S-1**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**INSTITUT TEKNOLOGI NASIONAL MALANG**  
**2013**

**APLIKASI PENDETEKSIAN PENYAKIT PERIODONTITIS KRONIS CITRA PANORAMIC PADA GIGI MANUSIA BERBASIS MATLAB MENGGUNAKAN ALGORITMA LINE STRENGTH DAN LINE TRACKING**

**Kadek Suarjuna Batu Bulan (0918117)**

Teknik Informatika ITN Malang

Fakultas Teknologi Industri

Institut Teknologi Nasional Malang

Jl. Raya Karanglo Km. 2 Tasikmadu-Malang

[kadek.suarjuna@yahoo.com](mailto:kadek.suarjuna@yahoo.com)

**Dosen Pembimbing : I. Ir. Yusuf Ismail Nakhoda, MT**

**II. Suryo Adi Wibowo, ST**

***Abstrak***

*Periodontitis kronis merupakan suatu penyakit peradangan pada jaringan periodontal. Proses deteksi penyakit ini biasanya dilakukan secara klinis, namun jika dilakukan secara kasat mata dari citra dental panoramic radiograph, penyakit ini sangat sulit di deteksi.*

*Dalam penelitian ini akan dibangun sebuah system computer vision untuk pendeteksian penyakit periodontitis kronis. Proses deteksi diawali dengan menggunakan algoritma line strength untuk mendeteksi pola garis pada citra dengan sudut tertentu. Selanjutnya dilakukan proses line tracking sebagai metode integrasi untuk melacak garis dari setiap piksel pada citra dengan diameter tertentu.*

*Penyakit periododontitis kronis pada citra dental panoramic dapat diidentifikasi dengan menggunakan algoritma line strength dengan integrasi metode line tracking berbasis multiscale line tracking. Penggabungan citra hasil integrasi metode line tracking berbasis multiscale line tracking dapat merepresentasikan nilai piksel-piksel dari setiap level citra line strength dan line tracking. Berdasarkan hasil uji coba yang dilakukan pada kinerja program dapat menghitung nilai line strength dan line tracking setelah diproses data citra menunjukkan kesesuaian dengan ciri khas diagnosa penyakit periodontitis kronis. Jika bone value lebih kecil sama dengan threshold maka diagnosanya adalah periodontitis dan jika bone value lebih besar dari threshold maka aiagnosanya normal.*

*Kata kunci: Line Strenght, Line Tracking, Periodontitis kronis, Dental Panoramic Radigraphs.*



# **Application Detection of Chronic Periodontitis Disease Panoramic Image Based On Human Teeth Using MATLAB Algorithms Line strength and Line Tracking**

**Kadek Suarjuna Batu Bulan**

Teknik Informatika ITN Malang

Fakultas Teknologi Industri

Institut Teknologi Nasional Malang

Jl. Raya Karanglo Km. 2 Tasikmadu-Malang

[kadek.suarjuna@yahoo.com](mailto:kadek.suarjuna@yahoo.com)

## ***Abstract***

*Chronic periodontitis is an inflammatory disease of the periodontal tissues. The process of detection of the disease is usually made clinically, but if done by naked eye from the image of dental panoramic radiograph, the disease is very difficult to detect.*

*In this research will be built a computer vision system for the detection of chronic periodontitis disease. The process begins with the detection of line strength using an algorithm to detect the line pattern on the image to a particular angle. Further line tracking process as a method of integration to trace a line from each pixel in the image with a certain diameter.*

*Periododontitis chronic disease dental panoramic image can be identified using algorithms with integrated line strength line tracking method based on multiscale line tracking. Merging the results of the integration of image-based tracking methods multiscale line tracking line to represent the value of pixels of each image level line strength and line tracking. Based on the results of experiments performed on the performance of the program can calculate the value of line strength and line tracking after processing image data demonstrate conformity with the typical diagnosis of chronic periodontitis. If the bone is smaller value equal to the threshold then the diagnosis is periodontitis and if the bone is greater than the threshold value then the normal diagnosis.*

**Keywords:** *Strength line, Line Tracking, Chronic Periodontitis, Dental Panoramic radiographs.*

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas karunia, rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian yang berjudul **“APLIKASI PENDETEKSIAN PENYAKIT PERIODONTITIS KRONIS CITRA PANORAMIC PADA GIGI MANUSIA BERBASIS MATLAB MENGGUNAKAN ALGORITMA LINE STRENGHT DAN LINE TRACKING”**.

Skripsi ini dapat terselesaikan tidak terlepas dari dukungan berbagai pihak. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

2. Bapak Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT selaku Kepala Jurusan Teknik Informatika S-1 Institut Teknologi Nasional Malang.
4. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Dosen Pembimbing I yang telah memberikan saran dan bimbingannya dalam penyusunan laporan ini.
5. Bapak Suryo Adi Wibowo, ST selaku Dosen Pembimbing II yang telah memberikan saran dan bimbingannya dalam penyusunan laporan ini.
6. Ibu Sandy Nataly, ST selaku Dosen penguji I yang telah menguji program dan laporan skripsi.
7. Ibu Karina Auliasari, ST, MT selaku Dosen penguji 2 yang telah menguji program dan laporan skripsi.
8. Bapak dan Ibu Dosen yang telah mengajar penulis selama studi di Institut Teknologi Nasional Malang.
9. Rekan-rekan Teknik Informatika angkatan 2009 dan berbagai pihak yang turut membantu dalam penyelesaian laporan ini.

Semoga apa yang telah disajikan dapat memberikan manfaat dan pengetahuan bagi para pembaca. Segala kritik dan saran yang bersifat membangun, diterima dengan senang hati sebagai tambahan ilmu pengetahuan.

Malang, 21 Februari 2013

Penulis



## DAFTAR ISI

Halaman

<b>LEMBAR PERSETUJUAN</b> .....	i
<b>ABSTRAKSI</b> .....	ii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	v
<b>DAFTAR TABEL</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	viii

### **BAB I PENDAHULUAN**

1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian .....	3
1.4. Batasan Masalah .....	3
1.5. Metode Penelitian .....	3
1.6. Sistematika Penulisan .....	4

### **BAB II LANDASAN TEORI**

2.1. Dental Panoramic Radiographs.....	6
2.2. periodontitis Kronis.....	6
2.2.1. Manfaat Dan Kontribusi.....	7
2.2.2. Metodologi .....	7
2.3. Kriptografi.....	8
2.4. Line Strenght Dan Line Tracking.....	10
2.5. Computer Vision .....	11
2.6. MATLAB .....	13
2.6.1. Operasi Dasar Pengolahan Citra .....	13
2.6.2. Konvolusi .....	18
2.6.3. Perbaikan Kualitas Citra .....	23
2.6.4. Pendeteksian Tepi .....	26
2.7. Jenis-Jenis Foto Rontgen .....	31
2.7.1. Rontgen Intra Oral.....	31

2.7.2. Rontgen periapikal .....	31
2.7.3. BITE WING .....	31
2.7.4. Rontgen Oklusal.....	31
2.7.5. Rontgen Ekstra Oral.....	32
2.7.6. Rontgen Panoramic .....	32
2.7.7. Lateral .....	32
2.7.8. Postero Anterior .....	32
2.7.9. Antero posterior .....	32
2.7.10. Chephalometri .....	33
2.7.11. Proyeksi Water's.....	33
2.7.12. Proyeksi Reserve-Towne .....	33
2.7.13. Proyeksi Submentovertex.....	33

### **BAB III ANALISA DAN PERANCANGAN**

3.1. Perancangan Sistem .....	34
3.1.1. Diagram Alir Program Utama.....	35
3.1.2. Diagram Alir Proses Line Strenght.....	36
3.2.3. Diagram Alir Proses Line Tracking .....	37
3.2. Perancangan Struktur Menu.....	38
3.3. Analisa Sistem.....	38
3.3.1. Analisa Gaussian Piramida .....	39
3.3.2. Integrasi Metode Line Strenght Dengan Line Tracking .....	40
3.4. Desain Perancangan GUI (Graphical User Interface).....	43
3.4.1. Rancangan Database .....	43
3.4.2. Rancangan Halaman Tabel Petugas 1 .....	44
3.4.3. Rancangan Halaman Tabel Petugas 2 .....	45
3.4.4. Rancangan Halaman Memasukkan Citra Untuk Diproses.....	45
3.4.5. Rancangan Halaman Memproses Citra.....	46
3.4.6. Rancangan Halaman Hasil Proses Aplikasi Program .....	47

### **BAB IV HASIL DAN PENGUJIAN**

4.1. Implementyasi GUI.....	48
4.1.1. Implementasi Tabel Petugas .....	48
4.1.2. Implementasi Halaman Input Dan Crop Citra .....	49



4.1.3. Implementasi Halaman Proses Citra .....	49
4.1.4. Implementasi Halaman Hasil Akhir Proses .....	50
4.2. Pengujian Program Aplikasi .....	50
4.2.1. Halaman Database MY SQL.....	50
4.2.2. Halaman Utama.....	51
4.2.3. Halaman Input Dan Crop Citra .....	53
4.2.4. Halaman Memilih Citra .....	53
4.2.5. Halaman Inputan Citra .....	54
4.2.6. Halaman Proses Crop Citra .....	54
4.2.7. Halaman Hasil Crop Citra .....	55
4.2.8. Halaman Proses Citra .....	55
4.2.9. Halaman Akhir Proses.....	56
4.3. Hasil Pengujian Program .....	57

## **BAB V KESIMPULAN DAN SARAN**

5.1. Kesimpulan .....	62
5.2. Saran.....	62

<b>DAFTAR PUSTAKA.....</b>	<b>63</b>
----------------------------	-----------

## **LAMPIRAN**

### **DAFTAR TABEL**

Halaman

## **BAB II LANDASAN TEORI**

<b>Tabel 2.1.</b> Tabel Kebenaran Logika AND .....	16
<b>Tabel 2.2.</b> Tabel Kebenaran Logika OR.....	16
<b>Tabel 2.3.</b> Tabel Kebenaran Logika XOR.....	16

## **BAB IV IMPLEMENTASI DAN PENGUJIAN**

<b>Tabel 4.1.</b> Hasil Pengujian Implementasi Program.....	58
<b>Tabel 4.2.</b> Hasil Identifikasi Berdasarkan Nilai LS dan Nilai LT .....	59

## DAFTAR GAMBAR

Halaman

### BAB II LANDASAN TEORI

<b>Gambar 2.1.</b> Integrasi Metode .....	7
<b>Gambar 2.2.</b> Metodologi.....	8
<b>Gambar 2.3.</b> Proses Enkripsi dan Dekripsi.....	9
<b>Gambar 2.4.</b> Ilustrasi Proses Konvolusi Pada Elemen Piksel .....	21
<b>Gambar 2.5.</b> Permasalahan Padding Pada Batas Citra .....	22
<b>Gambar 2.6.</b> Ilustrasi Penerapan Zero Padding Pada Proses Konvolusi .....	22
<b>Gambar 2.7.</b> Piksel Bernilai 35 Terkena Derau.....	24
<b>Gambar 2.8.</b> Piksel Bernilai 35 Diganti Dengan Nilai 10 .....	25

### BAB III ANALISIS DAN PERANCANGAN

<b>Gambar 3.1.</b> Flowchart Diagram Alir Program Utama .....	35
<b>Gambar 3.2.</b> Flowchart Diagram Alir Line Strenght.....	36
<b>Gambar 3.3.</b> Flowchart Diagram Alir Line Tracking.....	37
<b>Gambar 3.4.</b> Rancangan Struktur Menu .....	38
<b>Gambar 3.5.</b> Proses Gaussian Pyramid, (a) Reduce, (b) Expand .....	39
<b>Gambar 3.6.</b> Inisialisasi Parameter Line Tracking .....	40
<b>Gambar 3.7.</b> Update Matrik Cw .....	41
<b>Gambar 3.8.</b> Proses Map Quantization.....	42
<b>Gambar 3.9.</b> Hasil Output Line Tracking.....	43
<b>Gambar 3.10.</b> Perancangan Database .....	44
<b>Gambar 3.11.</b> Perancangan Form Menu Data Tabel Petugas 1.....	44
<b>Gambar 3.12.</b> Perancangan Form Tabel Petugas 2.....	45
<b>Gambar 3.13.</b> Perancangan Memasukkan Gambar, Crop dan Mulai Proses.....	46
<b>Gambar 3.14.</b> Perancangan Proses Aplikasi Program .....	47
<b>Gambar 3.15.</b> Perancangan Hasil Proses Aplikasi Program.....	47

### BAB IV IMPLEMENTASI DAN PENGUJIAN

<b>Gambar 4.1.</b> Form Implementasi Tabel Petugas Baru dan Tabel Petugas .....	48
<b>Gambar 4.2.</b> Form Implementasi Halaman Input Dan Crop Citra .....	48



<b>Gambar 4.3.</b> Form Implementasi Halaman Proses Citra.....	49
<b>Gambar 4.4.</b> Form Implementasi Halaman Akhir Proses.....	50
<b>Gambar 4.5.</b> Form Halaman Database MY SQL .....	50
<b>Gambar 4.6.</b> Form Aplikasi Tabel Petugas Baru dan Tabel Petugas .....	51
<b>Gambar 4.7.</b> Tampilan Halaman Input Citra Dan Crop .....	53
<b>Gambar 4.8.</b> Tampilan Menu Memilih <i>File Image</i> .....	53
<b>Gambar 4.9.</b> Preview File Image Yang Dijadikan Sebagai Awal Proses.....	54
<b>Gambar 4.10.</b> Gambar Proses Mengecrop Citra.....	54
<b>Gambar 4.11.</b> Hasil Crop Citra.....	55
<b>Gambar 4.12.</b> Halaman Proses Citra .....	56
<b>Gambar 4.13.</b> Halaman Hasil Akhir .....	57

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Jaringan komputer dan internet telah mengalami perkembangan yang sangat pesat. Teknologi ini mampu menyambungkan hampir semua komputer di dunia sehingga bisa saling berkomunikasi dan bertukar informasi. Bentuk informasi tersebut dapat berupa data teks, gambar, gambar bergerak ataupun suara.

File-file berbentuk gambar selalu memiliki perpindahan melalui berbagai macam jaringan. Dengan pesatnya perkembangan jaringan komputer dan kemajuan yang tinggi di dunia digital, jumlah file digital yang dipertukarkan pun semakin tinggi. Terkadang sebagian besar data yang dipertukarkan tersebut bersifat privat, rahasia, yang pada akhirnya menghasilkan tingginya kebutuhan akan teknik penyembunyian data yang kuat.

*Periodontitis kronis* merupakan penyakit peradangan pada jaringan *periodontal*. Proses deteksi penyakit ini biasanya dilakukan secara klinis, namun jika dilakukan secara kasat mata dari *citra panoramic radiographs*, penyakit ini sangat sulit dideteksi. Gigi merupakan salah satu bagian dari tubuh manusia yang unik, karena menjadi satu-satunya bagian tubuh yang mirip tulang dan letaknya berada di dalam rongga mulut serta terlihat menonjol

*Dental panoramic radiographs* adalah sebuah alat yang digunakan untuk mendapat citra rongga gigi pada manusia. Para dokter ahli di bidang gigi telah melakukan berbagai macam analisis secara medis tentang gigi, mengingat betapa pentingnya peran dari gigi terutama pada *dental panoramic radiographs*. Hubungannya adalah ketika para pakar mencoba melakukan beberapa macam diagnosa penyakit, ternyata banyak yang identik dengan kondisi dari gigi seseorang pasien. Hal inilah yang mendorong banyak peneliti medis maupun non medis untuk melakukan pengkajian lebih jauh tentang diagnosa penyakit dari citra



gigi. Sehingga wajar jika *dental panoramic radiographs* sangat sering diambil dari pasien untuk keperluan diagnosis oleh para dokter gigi.

*Line strength* adalah untuk mendeteksi pola garis pada citra dengan sudut tertentu. Karena belum optimalnya binerisasi hasil deteksi *linear structure* pada saat penentuan nilai *line strength* berdasarkan nilai maksimal dari penghitungan *strength*, sehingga perlu digunakan metode lain yang mampu melakukan binerisasi dari penghitungan nilai maksimal untuk hasil *line strength*

*Line tracking* sebagai metode integrasi untuk melacak garis dari setiap piksel pada citra dengan diameter tertentu. berdasarkan metode *line tracking*, yaitu dengan memberikan nilai kepercayaan untuk setiap piksel yang dilacak dengan parameter sudut dan diameter tertentu yang kemudian akan dilakukan proses penilaian untuk setiap piksel yang dilacak. Selanjutnya diberikan nilai kepercayaan untuk piksel yang mempunyai nilai bobot paling tinggi dan melebihi *threshold* yang telah ditetapkan.

### **1.2. Rumusan Masalah**

Berdasarkan uraian pada latar belakang diatas, maka permasalahannya dirumuskan sebagai berikut :

- a. Bagaimana mengukur *linear structure trabecular bone* pada *dental panoramic radiographs* ?
- b. Bagaimana melakukan pendeteksian penyakit periodontitis kronis pada *dental panoramic radiographs* berdasarkan analisa *line strength* dan *line tracking* ?

### **1.3. Tujuan**

Tujuan dari penulisan ini adalah :

1. Membangun suatu sistem *computer-aided* yang mampu mengukur *linear structure*.
2. melakukan pendeteksian penyakit *periodontitis kronis* pada *dental panoramic radiographs* berdasarkan algoritma *line strength* dan *line tracking*.

#### **1.4. Batasan Masalah**

Batasan masalah yang diberikan berdasarkan permasalahan yang telah dikemukakan diatas adalah sebagai berikut :

- a. Pengukuran *linear structure* hanya dilakukan pada *dental panoramic radiographs* dengan ukuran citra 128×128, yang di dalam citra tersebut terdapat *trabecular bone*.
- b. Deteksi penyakit periodontitis kronis pada *dental panoramic radiographs* dengan menggunakan fitur *line strength* dan *line tracking*.
- c. Implementasi perangkat lunak menggunakan Matlab 7.7.0.

#### **1.5. Metode Penelitian**

Metode yang digunakan oleh penulis dalam penyusunan tugas akhir ini adalah sebagai berikut :

##### **1. Metode Interview**

Pada metode ini penulis melakukan wawancara dan tanya jawab secara langsung kepada pihak terkait dengan objek data penelitian. Metode ini bertujuan untuk memperoleh penjelasan secara langsung tentang data – data yang dipelajari dengan metode pengamatan.

##### **2. Metode Observasi**

Pada metode ini penulis mengadakan pengamatan objek dengan cara melihat secara langsung pada kegiatan yang dilakukan. Metode ini diterapkan dengan tujuan untuk mendapatkan gambaran dan data – data yang jelas dan akurat.

##### **3. Metode Studi Literatur**

Pada metode ini penulis juga mencari data dari sumber – sumber bacaan seperti : buku, jurnal, referensi, web page, blog, dan karya tulis ilmiah.

## **1.6. Sistematika Penulisan**

Sistematika penyusunan proposal ditujukan untuk memberikan gambaran dan uraian dari proposal skripsi secara garis besar yang meliputi bab-bab sebagai berikut :

### **1. BAB I : PENDAHULUAN**

Pada Bab ini membahas tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Maksud dan Tujuan Penelitian, Manfaat Penelitian, Metode Penelitian dan Sistematika Penulisan Laporan Penelitian.

### **2. BAB II : LANDASAN TEORI**

Pada Bab ini membahas tentang Landasan Teori yang merupakan tinjauan pustaka, menguraikan teori-teori yang mendukung judul, dan pembahasan secara detail. Landasan teori dapat berupa definisi-definisi atau model yang langsung berkaitan dengan ilmu atau masalah yang diteliti. Pada bab ini juga dituliskan tentang software (komponen) yang digunakan dalam pembuatan Program atau keperluan saat penelitian.

### **3. BAB III: ANALISIS DAN PERANCANGAN SISTEM**

Bab ini berisi antara lain: Tinjauan Umum yang menguraikan tentang gambaran umum objek penelitian, misalnya gambaran umum Instansi (struktur organisasi, Pengelolaan dll), atau gambaran umum produk, serta data yang dipergunakan untuk memecahkan masalah-masalah yang dihadapi, berkaitan dengan kegiatan penelitian.

Pada Bab ini juga membahas “analisis masalah”, yang akan menguraikan tentang analisis terhadap permasalahan yang terdapat pada kasus yang sedang di teliti. Meliputi analisis terhadap masalah sistem yang sedang berjalan, analisis hasil solusinya, dan analisis kebutuhan penelitian.

### **4. BAB IV : IMPLEMENTASI DAN PEMBAHASAN**

Pada Bab ini akan membahas paparan implementasi dan analisis hasil uji coba program. serta memaparkan hasil-hasil dari tahapan penelitian, dari



tahap analisis, desain, implementasi desain, hasil testing dan implementasinya, berupa penjasanteoritik, baik secara kualitatif, kuantitatif, atau secara statistik. Dan Selain membandingkan dengan hasil penelitian yang masih manual.

## **5. BAB V : PENUTUP**

Pada Bab ini berisi kesimpulan dan saran. Kesimpulan didapat dari ulasan data – data penelitian, menyimpulkan bukti-bukti yang diperoleh dan akhirnya menarik intisari apakah hasil yang didapat (dikerjakan), layak untuk digunakan (diimplementasikan).

## BAB II

### LANDASAN TEORI

#### **2.1. *Dental Panoramic Radiographs.***

Gigi merupakan salah satu bagian dari tubuh manusia yang unik, karena menjadi satu-satunya bagian tubuh yang mirip tulang dan letaknya berada di dalam rongga mulut serta terlihat menonjol. Para dokter ahli di bidang gigi telah melakukan berbagai macam analisis secara medis tentang gigi, mengingat betapa pentingnya peran dari gigi terutama pada *dental panoramic radiographs*. Hubungannya adalah ketika para pakar mencoba melakukan beberapa macam diagnosa penyakit, ternyata banyak yang identik dengan kondisi dari gigi seseorang pasien. Artinya bahwa kualitas kesehatan gigi seorang pasien dengan diagnosa beberapa penyakit yang diderita mempunyai hubungan timbal balik yang selaras. Hal inilah yang mendorong banyak peneliti medis maupun non medis untuk melakukan pengkajian lebih jauh tentang diagnosa penyakit dari citra gigi. Sehingga wajar jika *dental panoramic radiographs* sangat sering diambil dari pasien untuk keperluan diagnosis oleh para dokter gigi.

#### **2.2. *Periodontitis Kronis***

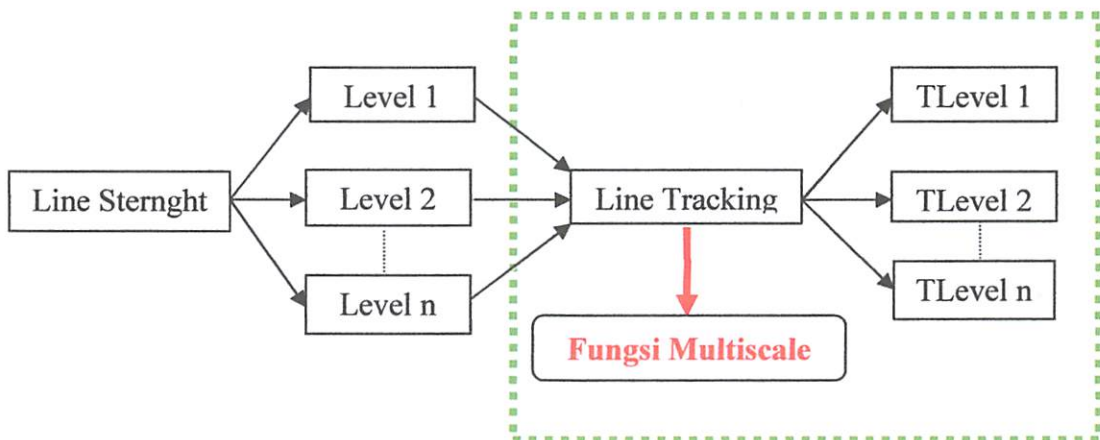
*Periodontitis* adalah peradangan atau infeksi pada *jaringan periodontal*. *Jaringan periodontal* adalah jaringan di sekitar perlekatan gigi yang mempunyai fungsi menyangga gigi. Jaringan ini terdiri dari *gingiva*, *sementum*, *ligamen periodontal* dan tulang *alveol*. Penyebab dari *periodontitis* adalah banyak faktor. Adapun tanda klinis dari *periodontitis* adanya inflamasi *gingiva* dan perdarahan, poket, resesi *gingiva*, mobilitas gigi, migrasi gigi, nyeri dan kerusakan tulang *alveol*. *Periodontitis* dibagi menjadi dua yaitu *periodontitis* kronis dan *periodontitis* agresif. *Periodontitis* kronis biasanya terjadi pada usia dewasa, perjalanan penyakitnya lambat, penyebabnya adalah faktor lokal. Gambaran radiografi pada *periodontitis* kronis yaitu tampak adanya kerusakan tulang horizontal. *Periodontitis* agresif biasanya terjadi pada usia muda, dibawah usia 30

tahun, perjalanan penyakitnya cepat, penyebabnya dari respon imun. Gambaran radiografi periodontitis agresif yaitu tampak adanya kerusakan tulang vertikal angular.

### 2.2.1. Manfaat dan Kontribusi

Memberikan sebuah solusi metode pendeteksian penyakit *periodontitis* kronis pada *dental panoramic radiographs* secara otomatis berdasarkan analisa *line strength* dan *line tracking*.

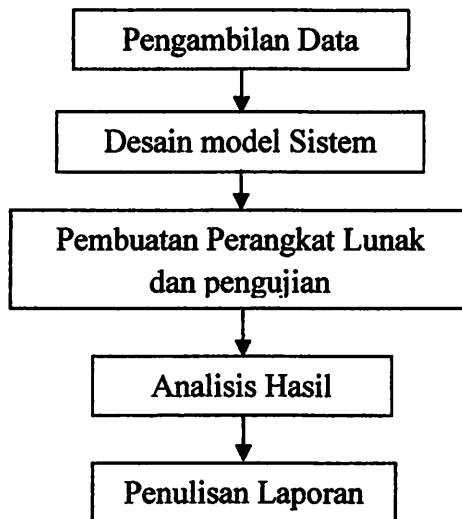
Mengintegrasikan proses *multiscale line tracking* untuk menajamkan batas garis dan memperjelas piksel di dalam diameter garis pada hasil citra *line strength*.



Gambar 2.1. Integrasi Metode

### 2.2.2 Metodologi

Metodologi program yang pertama diambil adalah melalui proses pengambilan data yang sudah ada dan dikaji dengan baik. Setelah pengambilan data yang diperlukan yaitu desain model system yang menarik dan mudah untuk dimengerti oleh pemakai. Selanjutnya membuat perangkat lunak dan pengujiannya untuk diketahui apakah sudah baik juga selesai dalam pengerjaannya baru akan dapat dianalisis hasilnya sesuai dengan yang diinginkan. Setelah semua selesai dalam pengerjaan dilanjutkan dengan penulisan laporan yang sesuai dengan program yang dikerjakan. Contoh diagram gambar 2.2:



Gambar 2.2. Metodologi

### 2.3 Kriptografi

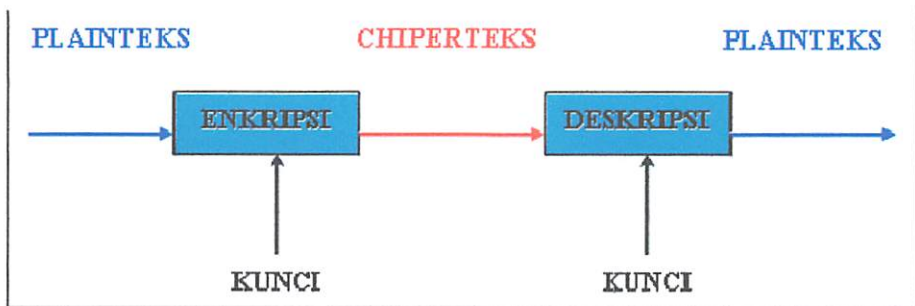
Kriptografi berasal dari bahasa Yunani. Menurut bahasa tersebut kata kriptografi dibagi menjadi dua kata, yaitu *kripto* dan *graphia*. Kripto berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan berita. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data.

Didalam kriptografi kita akan menemukan berbagai istilah atau terminologi. Beberapa istilah yang penting untuk diketahui diantaranya :

1. **Plaintext** (*P*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya.
2. **Ciphertext** (*C*) adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi.
3. **Enkripsi** (fungsi *E*) adalah proses perubahan *plaintext* menjadi *ciphertext*.
4. **Dekripsi** (fungsi *D*) adalah kebalikan dari enkripsi yakni mengubah *ciphertext* menjadi *plaintext* sehingga berupa data awal/asli.

5. **Kunci (K)** adalah parameter yang digunakan untuk transformasi *enchiperling* dan *dechiperling*. Kunci biasanya berupa *string* atau deretan bilangan.

Secara umum, kriptografi terdiri dari dua buah bagian utama yaitu bagian enkripsi dan bagian dekripsi. Enkripsi adalah proses transformasi informasi menjadi bentuk lain sehingga isi pesan yang sebenarnya tidak dapat dipahami, hal ini dimaksudkan agar informasi tetap terlindung dari pihak yang tidak berhak menerima. Sedangkan dekripsi adalah proses kebalikan enkripsi, yaitu transformasi data terenkripsi ke data bentuk semula. Proses transformasi dari plainteks menjadi cipherteks akan dikontrol oleh kunci. Peran kunci sangatlah penting, kunci bersama-sama dengan algoritma matematisnya akan memproses plainteks menjadi cipherteks dan sebaliknya. Adapun blok proses enkripsidekripsi secara umum dapat kita lihat pada gambar di bawah ini :



Gambar 2.3 Proses enkripsi dan dekripsi

Dengan menggunakan simbol  $P$  untuk *plaintext*  $C$  untuk *chiphertext*  $E$  untuk enkripsi  $D$  untuk dekripsi dan  $K$  untuk *key*, maka fungsi matematik dari enkripsi dan dekripsi dapat dituliskan dengan persamaan sebagai berikut :

$$EK(P) = C \text{ dan } DK(C) = P$$

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi, yaitu :

1. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun terutama yang menjaga otoritas atau kunci rahasia untuk membuka informasi yang telah disandikan.



2. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data system harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan dan pensubtitusian data lain kedalam data sebenarnya.
3. Autentifikasi, adalah berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan system maupun kesatuan informasi itu sendiri.
4. Non-repudiasi, adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengirimkan atau yang membuat.

#### **2.4 Line Strength Dan Line Tracking**

Dalam penelitian ini mengusulkan sebuah metode integrasi *line strength* dan *line tracking* yang menggunakan fitur terbaik dari kedua metode, yaitu deteksi garis berdasarkan intensitas piksel citra pada arah sudut tertentu dan deteksi garis berdasarkan lebar yang bervariasi. Metode ini adalah integrasi dari algoritma *line strength* yang memanfaatkan fungsi *multiscale* pada algoritma *line tracking* untuk melakukan deteksi kepadatan tulang dibawah akar gigi, tepatnya disebelah kanan dan kiri foramen. Proses deteksi diawali dengan proses deteksi *linear structure* untuk menghasilkan nilai *line strength*. Karena belum optimalnya binerisasi hasil deteksi *linear structure* pada saat penentuan nilai *line strength* berdasarkan nilai maksimal dari penghitungan *strength*, sehingga perlu digunakan metode lain yang mampu melakukan binerisasi dari penghitungan nilai maksimal untuk hasil *line strength* berdasarkan metode *line tracking*, yaitu dengan memberikan nilai kepercayaan untuk setiap piksel yang dilacak dengan parameter sudut dan diameter tertentu yang kemudian akan dilakukan proses penilaian untuk setiap piksel yang dilacak. Selanjutnya diberikan nilai kepercayaan untuk piksel yang mempunyai nilai bobot paling tinggi dan melebihi *threshold* yang telah ditetapkan. Hal ini diharapkan agar sistem mampu membedakan mana piksel yang merupakan garis dan yang bukan garis.

## 2.5 Computer Vision

*Computer vision* adalah ilmu dan metode aplikasi dalam menggunakan computer untuk memahami isi citra (image content). Area permasalahan computer vision adalah pengukuran dan pemrosesan, yang dapat dilakukan dengan berbagai metode. Beberapa area permasalahan computer vision adalah sebagai berikut:

- a. Recognition/Pengenalan bertujuan mengenali objek data citra, aplikasi seperti Content Based Image Retrieval (CBIR), Optical Character Recognition.
- b. Motion/gerakan bertujuan mengenali data citra bergerak. Aplikasinya seperti Egomotion yang membagi gerakan 3D dari kamera, Tracking yang memperkirakan satu atau beberapa objek dalam citra bergerak.
- c. Restorasi citra, bertujuan untuk mendapatkan data citra, citra bergerak atau objek 3D tanpa noise.

*Content Based Image Retrieval* (Temu Kembali Citra Berbasis Isi) adalah salah satu aplikasi dari *computer vision* dalam permasalahan temu kembali citra dalam jumlah data citra yang besar. Arti "Berbasis Isi" adalah pencarian citra berdasarkan isi citra. Terminologi CBIR dikenalkan pada tahun 1992 oleh T.Kato, dalam ujicobanya dijelaskan sistem temu kembali citra dari *database* citra berdasarkan warna dan bentuk. Sejak itu, terminologi CBIR digunakan dalam menjelaskan proses temu kembali citra yang diinginkan dari koleksi citra yang besar. Sistem CBIR yang ideal dalam prespektif user adalah pencarian citra berdasarkan teks. Hingga saat ini telah dikembangkan beberapa sistem temu kembali citra berbasis isi dalam berbagai aplikasi sebagai berikut:

- a. *Query By Image Content* (QBIC) dikembangkan oleh IBM. Dalam QBIC, query/permintaan user bisa dilakukan dengan menggunakan citra contoh, sket yang digambar user, atau tekstur dan warna yang dipilih *user*.
- b. *Netra* dikembangkan di UC Santa Barbara. Informasi mengenai warna, tekstur, bentuk dan lokasi dari *region* (bagian-bagian citra yang memiliki ciri-ciri yang unik) citra digunakan untuk menemukan

kembali *regionregion* yang mirip dari citra-citra yang tersimpan dalam *database(system penyimpanan dan pengelolaan data)*.

- c. *Blobworld* dikembangkan di UC Berkeley. Tiap citra disegmentasi secara otomatis ke dalam *region* berdasarkan fitur warna dan tekstur. *Query* dilakukan dengan menggunakan atribut-atribut dari satu atau dua *region* yang diinginkan (bersifat lokal), tidak menggunakan atribut-atribut global yang ada dalam satu citra.
- d. *Retrieval Ware* dikembangkan oleh Excalibur Technologies Corp. Pencarian pada *Retrieval Ware* dilakukan berdasarkan fitur warna, bentuk, tekstur isi, kecerahan, struktur warna, dan rasio. *Retrieval Ware* mendukung kombinasi dari fitur-fitur tersebut dengan nilai fitur yang dapat dikontrol oleh user.
- e. *Image Rover* dikembangkan oleh Boston University, mengkombinasikan fitur tekstual dan visual untuk pencarian citra di web. User mengawali pencarian dengan memberikan teks yang spesifik dari citra yang diharapkan.
- f. MARS dikembangkan di University of Illinois at Urbana-Champaign. MARS mengolah berbagai fitur visual ke dalam arsitektur *retrieval* yang berguna dan dapat digunakan pada aplikasi dan user yang berbeda.
- g. PicToSeek adalah *image retrieval* untuk web, dikembangkan oleh ISIS Research Group di Univeristy of Amsterdam. Ide utamanya adalah mengekstrak *invariant fitur* (fitur yang berlainan) dari setiap citra pada *database*, kemudian hasil ekstraksi disesuaikan dengan kumpulan fitur *invariant* dari citra *query*.

Pada sistem-sistem yang sudah dikembangkan sebelumnya, kebanyakan *query* dilakukan dengan menggunakan sket, citra contoh, atau *region* contoh. Penggunaan *query* ini kurang memudahkan user. Seringkali user ingin melakukan *query* dengan *meng-input-kan* deskripsi teks mengenai "objek-objek" yang ada dalam citra. User menginputkan teks "orang", "perahu", "laut", jika user ingin mencari bermacam-macam citra yang bertema laut dan di dalamnya juga memuat orang dan perahu. Untuk memenuhi kebutuhan tersebut, dapat dilakukan

pengindeksan secara manual, artinya sebelum citra disimpan ke dalam database, objek-objek yang terdapat dalam citra tersebut dikenali secara manual terlebih dahulu oleh seorang operator, setelah itu baru disimpan ke dalam database berdasarkan objek yang sudah dikenali. Jika ukuran citra sangat besar, pengindeksan manual akan sulit dilakukan, karena:

- a. Membutuhkan waktu yang sangat lama
- b. Bersifat subyektif, sebuah gambar bisa dimaknai berbeda oleh dua orang operator

Oleh karena itu diperlukan suatu sistem pengenalan objek otomatis untuk keperluan pengorganisasian data citra berukuran besar. Penggunaan CBIR Tiga tahapan Sistem Temu Kembali Citra berbasis isi yaitu pelatihan data, pengenalan objek, dan pencarian data.

## **2.6 MATLAB**

Program berikut telah berhasil diimplementasikan menjadi suatu perangkat lunak sebagai aplikasi pengolahan citra digital. Dengan menggunakan pemrograman MATLAB (Matrix Laboratory) 2008 dibuat sebuah *project* untuk watermark citra pada id card. *Project* pada MATLAB (Matrix Laboratory) 2008 adalah suatu *file* yang memuat informasi tentang sekumpulan form, unit dan beberapa hal yang lain dalam program aplikasi. *File* utama *project* disimpan dengan ekstensi *.m*. Pada saat dijalankan, *file project* inilah yang selalu dikompilasi menjadi *file* yang dapat dijalankan. Tahap pertama dalam proses pembuatan aplikasi adalah melakukan proses pengambilan citra berbeda dalam komputer dengan ukuran 8 bit agar dapat diproses dengan baik. Berikut beberapa macam matlab yang digunakan dalam program ini:

### **2.6.1 Operasi Dasar Pengolahan Citra**

Citra digital direpresentasikan dengan matriks. Operasi pada citra digital pada dasarnya adalah memanipulasi elemen-elemen matriks.

## 1. Operasi Negatif

Operasi negatif bertujuan untuk mendapatkan citra negatif dengan cara mengurangi nilai intensitas piksel dari nilai keabuan maksimum. Secara umum persamaannya adalah sebagai berikut :

$$f(x,y)' = 255 - f(x,y)$$

## 2. Operasi Clipping

Yang dimaksud operasi clipping adalah operasi pemotongan jika nilai intensitas piksel hasil suatu operasi pengolahan citra terletak di bawah nilai intensitas minimum atau di atas nilai intensitas maksimum.

$$f(x,y)' = \begin{cases} 255 & f(x,y) > 255 \\ f(x,y), & 0 \leq f(x,y) \leq 255 \\ 0, & f(x,y) < 0 \end{cases}$$

## 3. Operasi Pengambangan

Operasi pengambangan (thresholding) adalah operasi memetakan nilai intensitas piksel ke salah satu dari dua nilai,  $a_1$  atau  $a_2$ , berdasarkan nilai ambang (threshold)  $T$

$$f(x,y)' = \begin{cases} a_1, & f(x,y) < T \\ a_2, & f(x,y) \geq T \end{cases}$$

## 4. Operasi Aritmatika

Karena Citra digital adalah matriks, maka operasi-operasi aritmatika matriks juga berlaku pada citra. Operasi matriks yang dilakukan adalah :

### a. Penjumlahan Citra

Penjumlahan citra adalah operasi menjumlahkan dua matriks yang berukuran sama. Secara umum, persamaannya adalah sebagai berikut :

$$C(x,y) = A(x,y) + B(x,y)$$



Perlu diingat bahwa syarat penjumlahan dua buah matriks adalah ukuran kedua matriks harus sama. Jika hasil penjumlahan intensitas lebih besar dari 255, maka intensitasnya dibulatkan ke 255.

#### **b. Pengurangan Citra**

Pengurangan citra adalah operasi saling mengurangi dua matriks yang berukuran sama. Secara umum, persamaannya adalah sebagai berikut :

$$C(x,y) = A(x,y) - B(x,y)$$

#### **c. Perkalian Citra**

Perkalian citra A dengan scalar c akan menghasilkan citra baru B yang intensitasnya lebih terang dari semula. Kenaikan intensitas setiap piksel sebanding dengan c. Operasi perkalian citra dengan scalar digunakan untuk kalibrasi kecerahan.

$$B(x,y) = A(x,y) * c$$

#### **d. Pembagian citra**

Pembagian citra A dengan scalar c akan menghasilkan citra baru B yang intensitasnya lebih gelap dari semula. Penurunan intensitas setiap piksel berbanding terbalik dengan c. Operasi pembagian citra dengan scalar digunakan untuk normalisasi kecerahan.

$$B(x,y) = A(x,y) / c$$

### **5. Operasi Boolean**

Selain operasi aritmatika, pemrosesan citra digital juga melibatkan operasi boolean (AND, OR, NOT, XOR).

### a. Operasi AND

Tabel 2.1 Tabel kebenaran logika AND

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

### b. Operasi OR

Tabel 2.2 Tabel kebenaran logika OR

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

### c. Operasi XOR

Tabel 2.3 Tabel kebenaran logika XOR

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

## 6. Operasi Geometri

### a. Operasi Translasi

Rumus translasi citra :

$$x' = x + Tx$$

$$y' = y + Ty$$

Yang dalam hal ini,  $T_x$  adalah besar pergeseran dalam arah  $x$ , sedangkan  $T_y$  adalah besar pergeseran dalam arah  $y$ .

### **b. Operasi Cropping**

Rumus operasi cropping pada citra :

$$w' = x_R - x_L$$

$$h' = y_B - y_T$$

Yang dalam hal ini,  $w'$  adalah lebar citra baru yang diperoleh setelah proses cropping. Sedangkan  $h'$  adalah tinggi citra baru.  $x_R$  dan  $x_L$  adalah dua titik disebelah kiri dan kanan pada arah sumbu  $x$ .  $y_B$  dan  $y_T$  adalah dua titik disebelah atas dan bawah pada arah sumbu  $y$ . Keempat titik  $x_R$ ,  $x_L$ ,  $y_B$ ,  $y_T$  akan digunakan sebagai koordinat-koordinat dimana citra akan dipotong.

### **c. Operasi Flipping**

Flipping adalah operasi geometri yang sama dengan pencerminan. Ada dua macam flipping : horizontal dan vertical.

Flipping vertikal adalah pencerminan pada sumbu- $X$  dari citra  $A$  menjadi citra  $B$ , yang diberikan oleh :

$$x' = 2x_c - x$$

$$y' = y$$

Flipping horizontal adalah pencerminan pada sumbu- $Y$  dari citra  $A$  menjadi citra  $B$ , yang diberikan oleh :

$$x' = x$$

$$y' = 2y_c - y$$

#### d. Operasi Rotasi

Rumus rotasi pada citra :

$$\begin{aligned}x' &= Xp + (x - Xp) * \cos \theta - (y - Yp) * \sin \theta \\y' &= Yp + (x - Xp) * \sin \theta + (y - Yp) * \cos \theta\end{aligned}$$

Yang dalam hal ini,  $Xp$  dan  $Yp$  merupakan titik pivot rotasi. Pivot yang dimaksud adalah koordinat titik tengah dari citra.

#### e. Operasi Scalling

Penskalaan citra, disebut juga scalling, yaitu pengubahan ukuran citra.

Rumus penskalaan citra :

$$x' = ShX * x$$

$$y' = ShY * y$$

Dalam hal ini,  $ShX$  dan  $ShY$  adalah factor skala masing-masing dalam arah  $x$  dan arah  $y$  pada koordinat citra.

Source code di atas disimpan dengan nama m-file '*perbesar.m*'. Selanjutnya untuk menguji keberhasilan source code di atas, buatlah suatu m-file lagi dan tuliskan source code di bawah ini

#### 2.6.2 Konvolusi

Konvolusi merupakan operasi yang mendasar dalam pengolahan citra. Konvolusi dua buah fungsi  $f(x)$  dan  $g(x)$  didefinisikan sebagai berikut :

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a)g(x - a)da$$

Yang dalam hal ini, tanda  $*$  menyatakan operator konvolusi, dan peubah (variabel)  $a$  adalah peubah bantu. Untuk fungsi diskrit, konvolusi didefinisikan sebagai :

$$h(x) = f(x) * g(x) = \sum_{a=-\infty}^{\infty} f(a)g(x-a)$$

Pada operasi konvolusi di atas,  $g(x)$  disebut kernel konvolusi atau kernel penapis (filter). Kernel  $g(x)$  merupakan suatu jendela yang dioperasikan secara bergeser pada sinyal masukan  $f(x)$ , yang dalam hal ini, jumlah perkalian kedua fungsi pada setiap titik merupakan hasil konvolusi yang dinyatakan dengan keluaran  $h(x)$ .

Untuk fungsi dua peubah (fungsi dua dimensi atau dwimatra), operasi konvolusi didefinisikan sebagai berikut :

a) Untuk fungsi malar

$$h(x, y) = f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b)g(x-a, y-b)dad b$$

b) Untuk fungsi diskrit

$$h(x, y) = f(x, y) * g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b)g(x-a, y-b)$$

Fungsi penapis  $g(x,y)$  disebut juga *convolution filter*, *convolution mask*, *convolution kernel*, atau *template*. Di dalam MATLAB, filter linear diimplementasikan melalui konvolusi dua dimensi. Dalam konvolusi, nilai dari sebuah piksel keluaran dihitung dengan mengalikan elemen-elemen pada dua buah matriks dan menjumlahkan hasilnya. Satu dari matriks tersebut merepresentasikan citra, sementara matriks yang lainnya adalah filter-nya.

## 1. Algoritma

Ukuran matriks kernel biasanya lebih kecil dari ukuran matriks citra. Sebagai contoh misalkan terdapat sebuah matriks citra  $I$  dan sebuah kernel  $k$  sebagai berikut :



$$\begin{bmatrix} -4 & 0 & 2 & 3 & 5 & -2 & -3 \\ -4 & 2 & 0 & -1 & 3 & 0 & 0 \\ 1 & 1 & 3 & 2 & 3 & 3 & 4 \\ 2 & 5 & 5 & 3 & 2 & 0 & 5 \\ -4 & 4 & 3 & 5 & -4 & -2 & -4 \end{bmatrix}$$

Matriks  $I$

$$\begin{bmatrix} 4 & -3 & 1 \\ 4 & 6 & 2 \end{bmatrix}$$

Kernel  $k$

Algoritma atau langkah-langkah untuk melakukan konvolusi terhadap matriks  $I$  adalah sebagai berikut :

1. Memutar *convolution kernel* 180 derajat untuk menghasilkan sebuah *computational molecule*

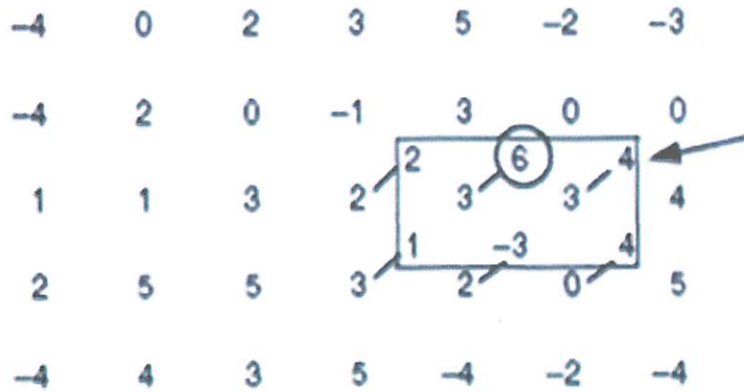
$$\begin{bmatrix} 4 & -3 & 1 \\ 4 & 6 & 2 \end{bmatrix} \xrightarrow{\text{Rotasi } 180^\circ} \begin{bmatrix} 2 & 6 & 4 \\ 1 & -3 & 4 \end{bmatrix}$$

2. Menentukan piksel tengah dari *computational molecule*.

Untuk menggunakan *computational molecule*, kita pertama-tama harus menentukan piksel tengah. Piksel tengah didefinisikan sebagai  $\text{floor}((\text{size}(h)+1)/2)$ . Sebagai contoh dalam sebuah molekul berukuran  $2 \times 3$ , maka piksel tengah adalah (1, 2).

$$\begin{bmatrix} 2 & 6 & 4 \\ 1 & -3 & 4 \end{bmatrix}$$

3. Menerapkan *computational molecule* pada setiap piksel citra masukan. Nilai dari piksel yang ditentukan dalam matriks keluaran B dihitung dengan menerapkan *computational molecule*  $h$  dengan piksel yang berkorespondensi pada matriks  $I$ . Kita dapat menggambarkan hal ini dengan menumpuk  $h$  pada  $I$ , dengan piksel tengah  $h$  menimpa piksel yang akan dihitung pada  $I$ . Kemudian kita mengalikan setiap elemen dari  $h$  dengan piksel pada  $I$  dan kemudian menjumlahkan hasilnya.



Gambar 2.4 Ilustrasi proses konvolusi pada elemen piksel (3,5)

Sebagai contoh kita akan menentukan nilai dari elemen piksel (3,5) pada matriks  $B$ , dengan cara menimpa  $h$  pada matriks  $I$  dengan piksel tengah dari  $h$  menutupi elemen piksel (3,5) dari matriks  $I$ . Piksel tengah diberi tanda lingkaran pada gambar.

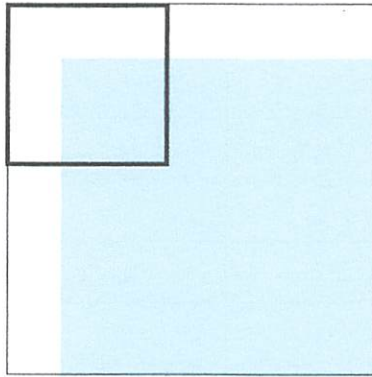
Terlihat ada enam piksel yang tertutupi oleh  $h$ , untuk setiap piksel-piksel ini, kalikan nilai dari piksel tersebut dengan nilai pada  $h$ . Jumlahkan hasilnya, lalu letakkan penjumlahan ini pada  $B(3,5)$ .

$$B(3,5) = 2 \cdot 2 + 3 \cdot 6 + 3 \cdot 4 + 3 \cdot 1 + 2 \cdot (-3) + 0 \cdot 4 = 31$$

Lakukan hal tersebut diatas untuk setiap piksel pada matriks  $I$  untuk menghasilkan setiap piksel yang berkorespondensi pada matriks  $B$ .

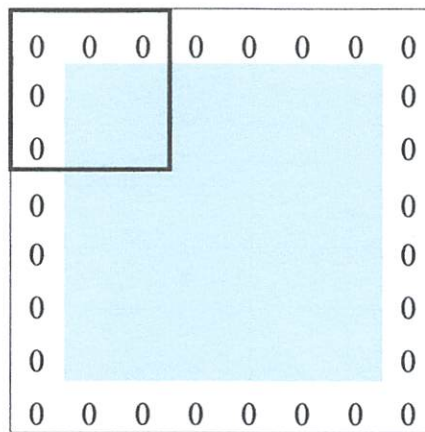
## 2. Proses Padding Pada Batas Citra

Ketika kita menerapkan filter pada piksel-piksel yang berada pada pinggir sebuah citra, beberapa elemen dari *computational molecule* tidak dapat menimpa piksel citra. Sebagai contoh jika molekul berukuran  $3 \times 3$  dan kita sedang menghitung untuk piksel pada ujung kiri atas dari citra, beberapa elemen pada molekul berada diluar batas citra, seperti terlihat pada gambar 1.2 berikut ini.



Gambar 2.5 Permasalahan padding pada batas citra

Untuk menghitung kondisi seperti di atas, proses konvolusi menerapkan suatu metode yang disebut dengan *zero padding*. *Zero padding* mengasumsikan bahwa piksel-piksel yang tidak dapat tertimpa oleh *computational molecule* bernilai nol.



Gambar 2.6 Ilustrasi penerapan zero padding pada proses konvolusi

Contoh perintah untuk melakukan konvolusi terhadap matriks  $I$  yang berukuran  $5 \times 5$  dan kernel yang berukuran  $3 \times 3$ . Operasi konvolusi yang dilakukan adalah dengan melibatkan *zero padding* di dalamnya.

### 2.6.3 Perbaikan Kualitas Citra

Perbaikan kualitas citra (*image enhancement*) merupakan salah satu proses awal dalam pengolahan citra (*image preprocessing*). Perbaikan kualitas citra diperlukan karena seringkali citra yang dijadikan objek pembahasan mempunyai

kualitas yang buruk, misalnya citra mengalami derau (*noise*) pada saat pengiriman, citra kurang tajam, kabur, dan sebagainya. Proses-proses yang termasuk ke dalam perbaikan kualitas citra adalah pelembutan citra (*image smoothing*) dan penajaman citra (*image sharpening*).

### 1. Pelembutan Citra (Image Smoothing)

Pelembutan citra bertujuan untuk menekan gangguan (*noise*) pada citra. Gangguan tersebut biasanya muncul sebagai akibat dari hasil penerokan yang tidak bagus. Gangguan pada citra umumnya berupa variasi intensitas suatu piksel yang tidak berkorelasi dengan piksel-piksel tetangganya. Secara visual, gangguan mudah dilihat oleh mata karena tampak berbeda dengan piksel tetangganya.

Piksel yang mengalami gangguan umumnya memiliki frekuensi tinggi. Komponen citra yang berfrekuensi rendah umumnya mempunyai nilai piksel konstan atau berubah sangat lambat. Operasi pelembutan citra dilakukan untuk menekan komponen berfrekuensi tinggi dan meloloskan komponen berfrekuensi rendah.

### 2. Filter Linier

Pada prinsipnya, filter yang digunakan dalam filter linear adalah *neighborhood averaging* merupakan salah satu jenis *low-pass filter*, yang bekerja dengan cara mengganti nilai suatu piksel pada citra asal dengan nilai rata-rata dari piksel tersebut dan lingkungan tetangganya. Penapis rata-rata adalah salah satu penapis lolos rendah yang paling sederhana. Aturan untuk penapis lolos rendah adalah :

1. Semua koefisien penapis harus positif
2. Jumlah semua koefisien harus sama dengan satu

Berikut beberapa penapis rata-rata, yaitu penapis lolos rendah yang sering digunakan pada operasi pelembutan.

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad \begin{bmatrix} \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{5} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}$$

### 3. Filter Median

Filter median sangat bermanfaat untuk menghilangkan *outliers*, yaitu nilai-nilai piksel yang ekstrim. Filter median menggunakan *sliding neighborhood* untuk memproses suatu citra, yaitu suatu operasi dimana filter ini akan menentukan nilai masing-masing piksel keluaran dengan memeriksa tetangga  $m \times n$  di sekitar piksel masukan yang bersangkutan. Filter median mengatur nilai-nilai piksel dalam satu tetangga dan memilih nilai tengah atau median sebagai hasil.

Median filter merupakan salah satu jenis *low-pass filter*, yang bekerja dengan mengganti nilai suatu piksel pada citra asal dengan nilai median dari piksel tersebut dan lingkungan tetangganya. Dibandingkan dengan *neighborhood averaging*, filter ini lebih tidak sensitif terhadap perbedaan intensitas yang ekstrim.

Pada penapis median, suatu ‘jendela’ (*window*) memuat sejumlah piksel. Jendela digeser titik demi titik pada seluruh daerah citra. Pada setiap pergeseran dibuat jendela baru. Titik tengah dari jendela ini diubah dengan nilai median dari jendela tersebut.

13	10	15	14	18
12	10	10	10	15
11	11	35	10	10
13	9	12	10	12
13	12	9	8	10

Gambar 2.7 Piksel bernilai 35 terkena derau

Sebagai contoh, tinjau jendela berupa kelompok piksel (daerah berwarna biru cerah) pada sebuah citra pada gambar di atas. Piksel yang sedang diproses adalah yang mempunyai intensitas 35.

10 10 10 11 35 10 9 12 10

Urutkan piksel-piksel tersebut:

9 10 10 10 10 10 11 12 35



Median dari kelompok tersebut adalah 10 (dicetak tebal). Titik tengah dari jendela (35) sekarang diganti dengan nilai median (10). Hasil dari penapis median diperlihatkan pada gambar di bawah ini. Jadi, penapis median menghilangkan nilai piksel yang sangat berbeda dengan piksel tetangganya.

13	<b>10</b>	15	14	18
12	<b>10</b>	<b>10</b>	<b>10</b>	15
11	<b>11</b>	<b>10</b>	<b>10</b>	10
13	9	12	10	12
13	12	9	8	10

Gambar 2.8 Piksel bernilai 35 diganti dengan nilai 10

#### 4. Penajaman Citra (Image Sharpening)

Inti dari penajaman (*sharpening*) citra adalah memperjelas tepi pada objek di dalam citra. Penajaman citra merupakan kebalikan dari operasi pelembutan citra karena operasi ini menghilangkan bagian citra yang lembut. Metode atau *filtering* yang digunakan adalah *high-pass filtering*.

Operasi penajaman dilakukan dengan melewatkan citra pada penapis lolos tinggi (*high-pass filter*). Penapis lolos tinggi akan meloloskan (atau memperkuat) komponen yang berfrekuensi tinggi (misalnya tepi atau pinggiran objek) dan akan menurunkan komponen berfrekuensi rendah. Akibatnya, pinggiran objek terlihat lebih tajam dibandingkan sekitarnya.

Karena penajaman citra lebih berpengaruh pada tepi (*edge*) objek, maka penajaman citra sering disebut juga penajaman tepi (*edge sharpening*) atau peningkatan kualitas tepi (*edge enhancement*).

Penapis pada operasi penajaman citra disebut penapis lolos tinggi. Aturan dari penapis lolos tinggi adalah sebagai berikut:

1. Koefisien boleh positif, negatif, atau nol
2. Jumlah semua koefisien adalah satu

Berikut contoh-contoh penapis lolos tinggi yang sering digunakan dalam penajaman citra :



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

#### 2.6.4 Pendeteksian Tepi

Tepi adalah perubahan intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang menampakkan rincian pada gambar. Tepi biasanya terdapat pada batas antara dua daerah berbeda pada suatu citra.

Tujuan operasi pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Karena tepi termasuk komponen berfrekuensi tinggi, maka pendeteksian tepi dapat dilakukan dengan penapis lolos tinggi.

##### 1. Operator Gradien Pertama

Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan yang besar. Kemiringan fungsi biasanya dilakukan dengan menghitung turunan pertama (gradient). Karena citra  $f(x,y)$  adalah fungsi dwimatra dalam bentuk diskrit, maka turunan pertamanya adalah secara parsial, masing-masing dalam arah- $x$  dan dalam arah- $y$ , sebagai berikut :

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$$

Dalam hal ini,

$$G_x = \frac{\partial f(x,y)}{\partial x} = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

$$G_y = \frac{\partial f(x,y)}{\partial y} = \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$

Biasanya  $\Delta x = \Delta y = 1$ , sehingga persamaan turunan menjadi :

$$G_x = \frac{\partial f(x, y)}{\partial x} = f(x+1, y) - f(x, y)$$

$$G_y = \frac{\partial f(x, y)}{\partial y} = f(x, y+1) - f(x, y)$$

Kedua turunan tersebut dapat dipandang sebagai dua buah mask konvolusi sebagai berikut :

$$G_1(x) = [-1 \ 1]$$

$$G_1(y) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Berdasarkan konvolusi dengan kedua mask tersebut, kita menghitung kekuatan tepi,  $G[f(x,y)]$ , yang merupakan magnitudo dari gradien.

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2}$$

Output yang dihasilkan di atas berturut-turut Gradien\_x, Gradien\_y, Magnitudo, dan hasil akhir setelah dilakukan thresholding adalah sebagai berikut :

Gradien\_x =

```

0 0 0 0 1
0 0 0 0 1
0 0 1 0 0
0 1 0 0 0
0 1 0 0 0

```

Gradien\_y =

```

0 0 0 0 0
0 0 0 1 1
0 0 1 0 0
0 0 0 0 0

```

1 1 0 0 0

Magnitudo =

0 0 0 0 1.0000

0 0 0 1.0000 1.4142

0 0 1.4142 0 0

0 1.0000 0 0 0

1.0000 1.4142 0 0 0

Hsl =

0 0 0 0 1

0 0 0 1 1

0 0 1 0 0

0 1 0 0 0

1 1 0 0 0

## 2. Operator Turunan Kedua

Operator turunan kedua disebut juga operator Laplace. Operator Laplace mendeteksi lokasi tepi yang lebih akurat khususnya pada tepi yang curam. Pada tepi yang curam, turunan keduanya memiliki persilangan nol (zero-crossing), yaitu titik di mana terdapat pergantian tanda nilai turunan kedua, sedangkan pada tepi yang landai tidak terdapat persilangan nol. Persilangan nol merupakan lokasi tepi yang akurat.

Turunan kedua fungsi dengan dua peubah adalah :

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Dengan menggunakan definisi hampiran selisih-mundur (*backward difference approximation*) :

$$G_3(x) = \frac{\partial f(x, y)}{\partial x} = \frac{f(x, y) - f(x - \Delta x, y)}{\Delta x}$$

$$G_3(y) = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y) - f(x, y - \Delta y)}{\Delta y}$$

Maka :

$$\begin{aligned} \nabla^2 f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &= G_1(G_3(x)) + G_1(G_3(y)) \\ &= \frac{1}{\Delta x} G_1(f(x, y)) - G_1(f(x - \Delta x, y)) + \frac{1}{\Delta y} G_1(f(x, y)) - G_1(f(x, y - \Delta y)) \\ &= \frac{1}{\Delta x} \left\{ \frac{f(x + \Delta x, y) - f(x, y) - f(x, y) + f(x - \Delta x, y)}{\Delta x} \right\} \\ &\quad + \frac{1}{\Delta y} \left\{ \frac{f(x, y + \Delta y) - f(x, y) - f(x, y) + f(x, y - \Delta y)}{\Delta y} \right\} \\ &= \frac{f(x + \Delta x, y) - 2f(x, y) + f(x - \Delta x, y)}{(\Delta x)^2} \\ &\quad + \frac{f(x, y + \Delta y) - 2f(x, y) + f(x, y - \Delta y)}{(\Delta y)^2} \end{aligned}$$

Dengan mengasumsikan  $\Delta x = \Delta y = 1$ , maka diperoleh :

$$\begin{aligned} \nabla^2 f(x, y) &= f(x + 1, y) - 2f(x, y) + f(x - 1, y) + f(x, y + 1) - 2f(x, y) + f(x, y - 1) \\ &= f(x, y - 1) + f(x - 1, y) - 4f(x, y) + f(x + 1, y) + f(x, y + 1) \end{aligned}$$

Atau dapat dinyatakan sebagai mask :

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Output yang dihasilkan di atas berturut-turut hasil konvolusi matriks citra dengan mask dan hasil akhir setelah dilakukan thresholding adalah sebagai berikut

Tepi =

```
-8 -4 0 -12 -8 -8 -16
-4 0 4 -4 0 0 -8
-4 0 4 -4 0 0 -8
-4 0 4 -4 0 0 -8
-8 -4 0 -12 -8 -8 -16
```

Hsl =

```
0 0 0 0 0 0 0
0 0 1 0 0 0 0
0 0 1 0 0 0 0
0 0 1 0 0 0 0
0 0 0 0 0 0 0
```

Perhatikan bahwa pendeteksian sisi yang dilakukan oleh operator laplacian menghasilkan tepi yang lebih akurat dibandingkan dengan operator gradien meskipun dengan nilai *threshold luminance* yang sama.

## 2.7 JENIS-JENIS FOTO RONTGEN

Secara garis besar foto rontgen gigi berdasarkan teknik pemotretan dan penempatan film, dibagi menjadi dua yaitu foto *Rontgen Intra Oral* dan foto *Rontgen Extra Oral*.

### 2.7.1 Rontgen Intra Oral

Teknik *radiografi intra oral* adalah pemeriksaan gigi dan jaringan sekitar secara radiografi dan filmnya ditempatkan di dalam mulut pasien. Untuk mendapatkan gambar lengkap rongga mulut yang terdiri dari 32 gigi diperlukan

kurang lebih 14 sampai 19 foto. Ada tiga pemeriksaan *radiografi intra oral* yaitu: *interproksimal* dan *oklusal*.

### **2.7.2 Rontgen periapikal**

Teknik ini digunakan untuk melihat keseluruhan mahkota serta akar gigi dan tulang pendukungnya. Ada dua teknik pemotretan yang digunakan untuk memperoleh foto periapikal yaitu teknik parallel dan bisektris yang sering digunakan di RSGM adalah teknik bisektris.

### **2.7.3 BITE WING**

Teknik ini digunakan untuk melihat mahkota gigi rahang atas dan rahang bawah daerah anterior dan posterior sehingga dapat digunakan untuk melihat permukaan gigi yang berdekatan dan puncak tulang alveolar. Teknik pemotretannya yaitu pasien dapat menggigit sayap dari film untuk stabilisasi film didalam mulut.

### **2.7.4 Rontgen Oklusal**

Teknik ini digunakan untuk melihat area yang luas baik pada rahang atas maupun rahang bawah dalam satu film. Film yang digunakan adalah film oklusal. Teknik pemotretannya yaitu pasien diinstruksikan untuk mengoklusikan atau menggigit bagian dari film tersebut.

### **2.7.5 Rontgen Ekstra Oral**

Foto rontgen ekstra oral digunakan untuk melihat area yang luas pada rahang dan tengkorak, film yang digunakan diletakkan diluar mulut. Foto rontgen ekstra oral yang paling umum dan paling sering digunakan adalah foto rontgen panoramic, sedangkan contoh foto rontgen ekstra oral lainnya adalah foto lateral, foto antero posterior, foto postero anterior, foto cephalometri, proyeksi-waters, proyeksi reserve-towne, proyeksi submentovertex.

### **2.7.6 Rontgen Panoramic**

Foto panoramic merupakan foto rontgen ekstra oral yang menghasilkan gambaran yang memperlihatkan struktur facial termasuk mandibula dan maksila beserta struktur pendukungnya. Foto rontgen ini dapat digunakan untuk mengevaluasi gigi impaksi, pola erupsi, pertumbuhan dan perkembangan gigi geligi, mendeteksi penyakit dan mengevaluasi trauma.

### **2.7.7 Lateral**

Foto rontgen ini digunakan untuk melihat keadaan sekitar lateral tulang muka, diagnose fraktur dan keadaan patologis tulang tengkorak dan muka.

### **2.7.8 postero Anterior**

Foto rontgen ini digunakan untuk melihat keadaan penyakit, trauma atau kelainan pertumbuhan dan perkembangan tengkorak. Foto rontgen ini juga dapat memberikan gambaran struktur wajah, antara lain sinus frontalis dan ethmoidalis, fossanasalis dan orbita.

### **2.7.9 Antero Posterior**

Foto rontgen ini digunakan untuk melihat kelainan pada bagian depan maksila dan mandibula, gambaran sinus frontalis, sinus ethmoidalis, serta tulang hidung.

### **2.7.10 Cephalometri**

Foto rontgen ini digunakan untuk melihat tengkorak tulang wajah akibat trauma penyakit dan kelainan pertumbuhan perkembangan. Foto ini juga dapat digunakan untuk melihat jaringan lunak nasofaringeal, sinus paranasal dan palarum keras.

### **2.7.11 Proyeksi Water's**

Foto rontgen ini digunakan untuk melihat sinus maksilaris, sinus ethmoidalis, sinus frontalis, sinus orbita, surura zigomatiko frontalis dan rongga masal.

### **2.7.12 Proyeksi Reserve-Towne**

Foto rontgen ini digunakan untuk pasien yang kondilusnya mengalami perpindahan tempat dan juga dapat digunakan untuk melihat dinding postero lateral pada maksila.

### **2.7.13 Proyeksi Submentovertex**

Foto ini bisa digunakan untuk melihat dasar tengkorak, posisi kondilus, sinus sphenoidalis, lengkung mandibula, dinding lateral sinus maksila dan arcus zigomatikus.



## BAB III

### PERANCANGAN SISTEM

#### 3.1. Perancangan Sistem

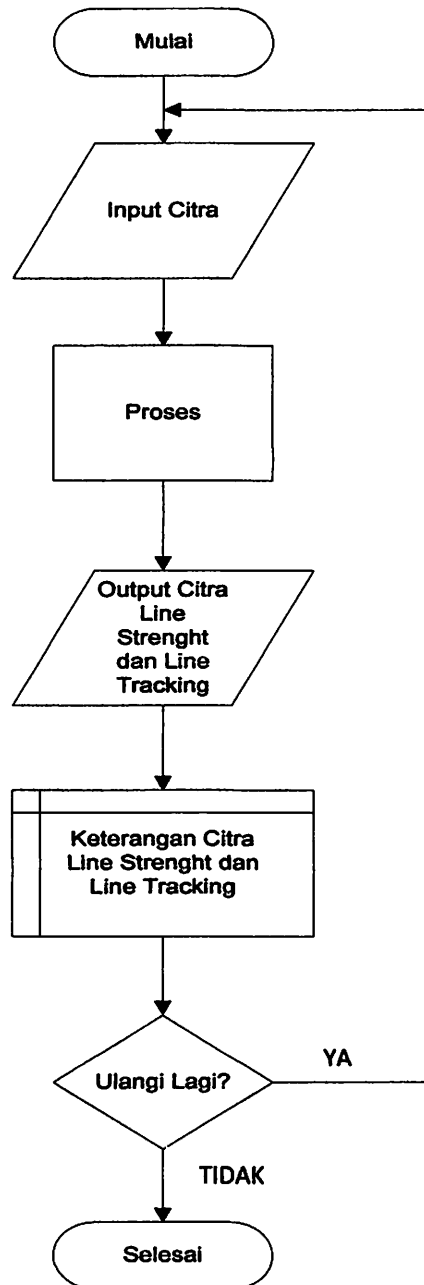
Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Kata logis merupakan kata kunci dalam algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar. Kertas coret-coretan itu akan digunakan untuk menyusun algoritma (langkah-langkah penyelesaian masalah), flowcharting (alur logika perintah, yang merupakan aplikasi dari algoritma), maupun menuliskan perintah sesuai dengan kaidah dari bahasa pemrograman yang akan digunakannya. Sewaktu menyusun algoritma, kita tidak perlu tahu (atau tidak perlu menyesuaikan dengan) bahasa pemrograman yang nanti akan kita gunakan. Hal utama yang kita pikirkan adalah kaidah (hirarki) dari komputer itu sendiri, yaitu input-proses-output.

Periodontitis kronis merupakan penyakit peradangan pada jaringan periodontal. Proses deteksi penyakit ini biasanya dilakukan secara klinis, namun jika dilakukan secara kasat mata dari citra panoramic radiographs, penyakit ini sangat sulit dideteksi. Hubungannya adalah ketika para pakar mencoba melakukan beberapa macam diagnosa penyakit, ternyata banyak yang identik dengan kondisi dari gigi seseorang pasien. Hal inilah yang mendorong banyak peneliti medis maupun non medis untuk melakukan pengkajian lebih jauh tentang diagnosa penyakit dari citra gigi. Sehingga wajar jika *dental panoramic radiographs* sangat sering diambil dari pasien untuk keperluan diagnosis oleh para dokter gigi.

Pada program yang akan dibangun sistem komputer vision ini terdapat dua algoritma utama yaitu *line strength* dan *line tracking* untuk pendeteksian penyakit periodontitis. Proses deteksi diawali dengan menggunakan algoritma *line strength* untuk mendeteksi pola garis pada citra dengan sudut tertentu. Selanjutnya dilakukan proses *line tracking* sebagai metode integrasi untuk melacak garis dari setiap piksel pada citra dengan diameter tertentu.

### 3.1.1 Diagram Alir Program Utama

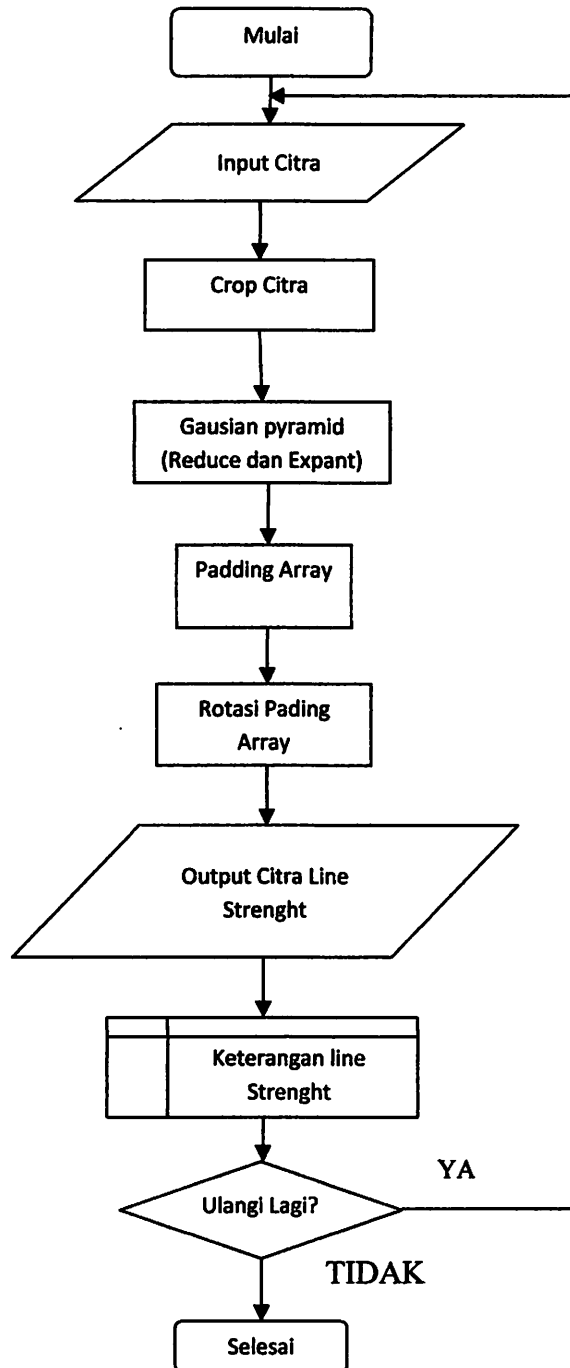
Diagram alir program utama dari program aplikasi pendeteksian penyakit periodontitis kronis pada citra *panoramic* pada gigi manusia berbasis matlab menggunakan algoritma *line strenght* dan *line tracking*. Gambar di bawah ini adalah diagram alir program utama dengan algoritma *Line Strenght* dan *Line Tracking* untuk mendeteksi penyakit *periodontitis kronis* sebagai berikut:



Gambar 3.1. Flowchart Diagram Alir Program Utama

### 3.1.2 Diagram Alir Proses Line Strenght

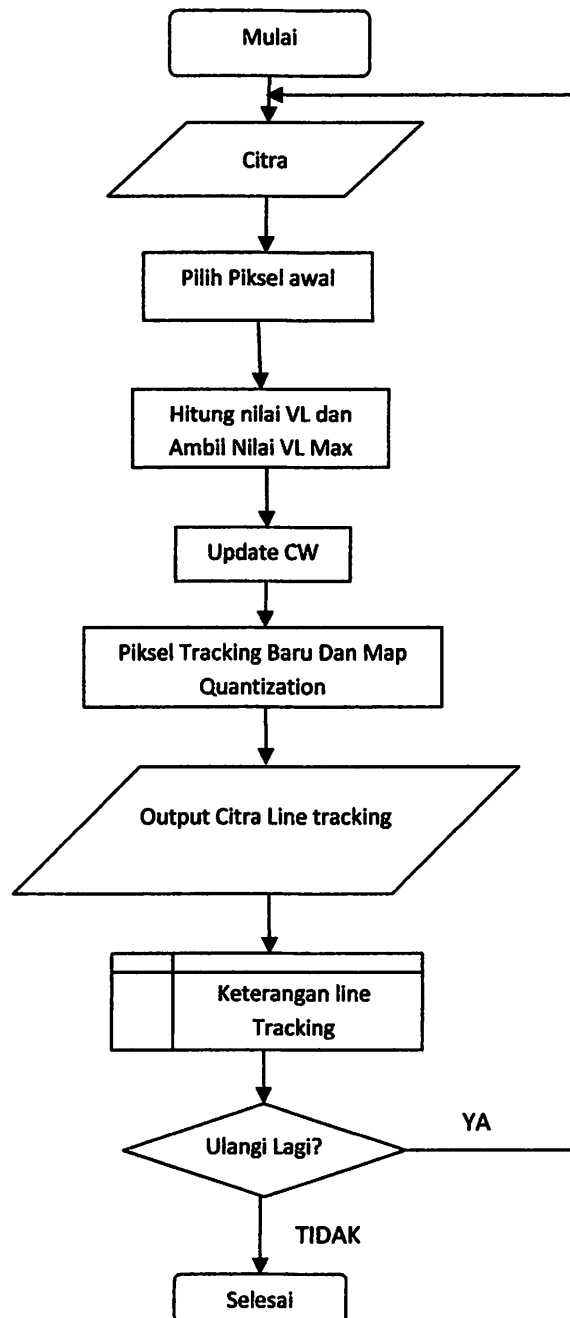
Diagram alir proses deteksi diawali dengan menggunakan *gaussian pyramid* yang dibagi menjadi *reduce* dan *expand* selanjutnya proses algoritma *line strenght* untuk mendeteksi pola garis pada citra dengan sudut tertentu. sebagai berikut:



Gambar 3.2. Flowchart Diagram Alir Line Strenght

### 3.1.3 Diagram Alir Proses Line Tracking

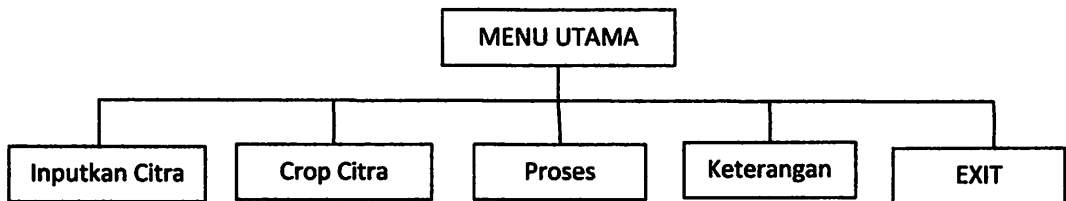
Diagram alir pertama diawali dengan memilih piksel awal selanjutnya menghitung nilai VL dan mendapatkan piksel tracking baru juga map quantization setelah itu baru proses *Line Tracking* sebagai metode integrasi untuk melacak garis dari setiap piksel pada citra dengan diameter tertentu. Dibawah ini adalah gambar diagram alir *Line Tracking*:



Gambar 3.3. Flowchart Diagram Alir Line Tracking

### 3.2 Perancangan Struktur Menu

Rancangan struktur menu merupakan prosedur-prosedur program dari sebuah program aplikasi. Menu-menu yang terdapat dalam program aplikasi pendeteksian penyakit *periodontitis kronis* pada *citra panoramic* pada gigi manusia berbasis matlab menggunakan algoritma *line strenght* dan *line tracking* dapat dilihat dalam bentuk rancangan struktur menu pada gambar berikut ini :



Gambar 3.4. Rancangan Struktur Menu

#### 1. *Menu Utama*

Menu Utama adalah tampilan awal pada aplikasi program berupa table petugas.

#### 2. *Inputkan Citra*

Inputkan citra adalah memasukkan citra untuk diproses.

#### 3. *Crop citra*

Crop citra adalah memotong citra yang di butuhkan untuk di proses.

#### 4. *Proses*

Proses adalah menganalisa citra untuk menemukan hasil atau outputnya.

#### 5. *Keterangan*

Keterangan adalah Analisa hasil gambar yang sudah di proses.

#### 6. *Exit*

Exit adalah perintah untuk mengakhiri jalannya program aplikasi.

### 3.3 Analisa Sistem

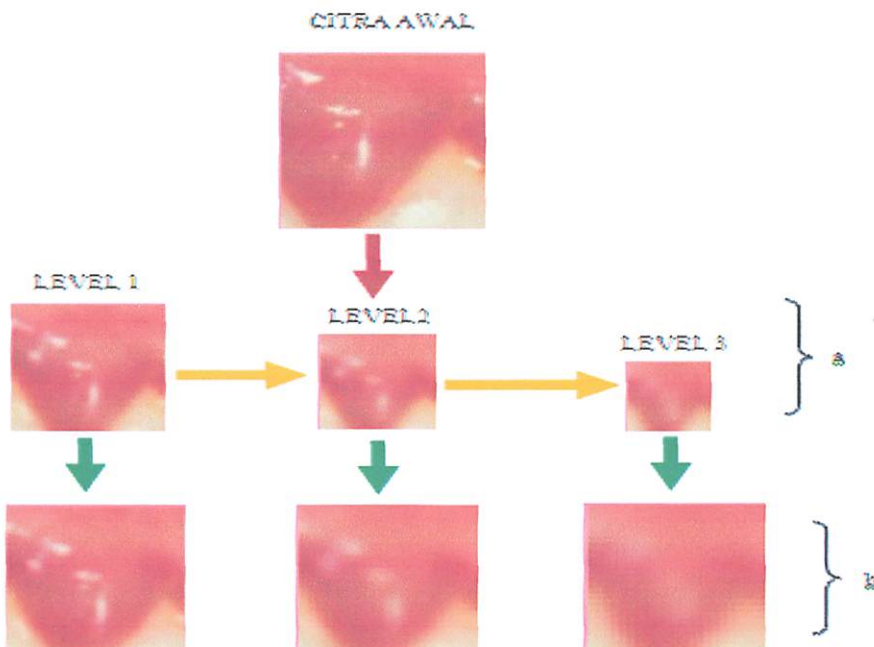
Analisa sistem dilakukan dengan mendefinisikan masalah yang ada, menganalisis dan mengamati secara keseluruhan bagaimana program aplikasi ini terbentuk. Dalam implementasi program aplikasi ini diperlukan suatu analisis

yang benar, agar menjadikan program aplikasi ini menjadi menarik dan sesuai dengan yang diharapkan.

### 3.3.1 Analisa Gaussian Piramida

*Lowpass Pyramid (Gaussian Pyramid)* adalah pemrosesan sebuah citra dimana sebuah citra akan di perhalus dan di-subsampling secara berangsur-angsur sebanyak nilai level yang diinginkan. Nilai density dan resolusi citra tersebut akan berkurang dari level ke level dan jika diilustrasikan akan sebuah piramida. Semakin tinggi level yang digunakan akan semakin kecil pula ukuran citra yang dihasilkan. Pada level yang paling kecil, citra yang dihasilkan akan memiliki resolusi yang paling tinggi, sebaliknya pada level yang paling tinggi maka citra yang di hasilkan akan memiliki nilai resolusi paling rendah.

Gambar dibawah merupakan citra hasil dari operasi lowpas pyramid dengan beberapa level, G0 adalah citra hasil dari level nol atau sama dengan citra asli, untuk menghasilkan citra G1, pada citra G0 dilakukan proses lowpass-filtered dan kemudian dilanjutkan dengan subsampling, jika citra G0 mempunyai ukuran  $i \times j$ , maka ukuran citra G1 menjadi  $i/2 \times j/2$ . Proses yang sama dilakukan untuk mendapatkan G2, G3 dan seterusnya.



Gambar 3.5. Proses Gaussian Pyramid, (a) Reduce, (b) Expand

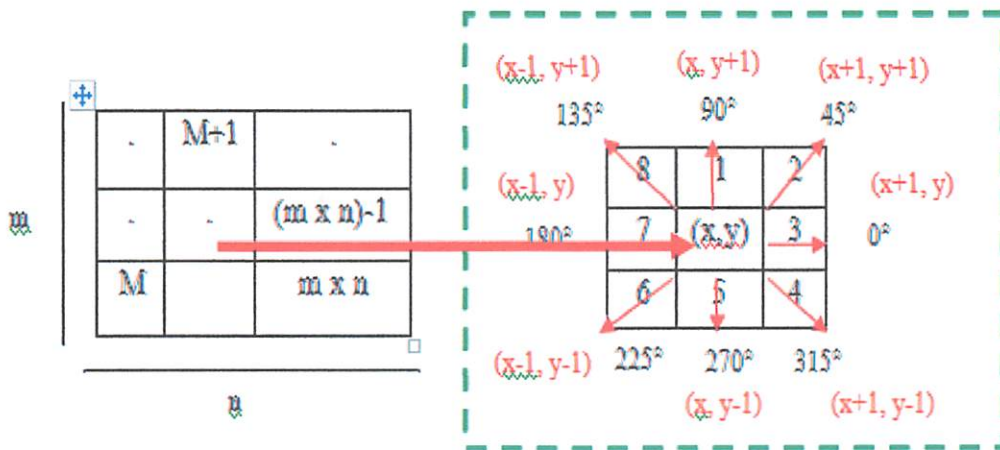
### 3.3.2 Integrasi Metode Line Strength dengan Line Tracking

Pembentukan citra *line strength* telah dikembangkan sesuai dengan metode pada penelitian sebelumnya. Namun perbedaan yang paling menonjol dengan penelitian ini adalah terletak pada pembuatan desain system yang digunakan untuk mendeteksi pola garis pada citra hasil gambar 5. Dengan arah sudut tertentu dan diameter garis tertentu.

Proses awal integrasi metode *line tracking* adalah mendapatkan delapan tetangga dari setiap piksel pada matrik gambar secara bersamaan (karena berada dalam satu matrik) sesuai dengan gambar 6. Berdasarkan delapan arah sudut, yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ,  $180^\circ$ ,  $225^\circ$ ,  $270^\circ$  dan  $315^\circ$ . Kemudian menghitung nilai VL (Cross Sectional) dari setiap delapan tetangga dari setiap satu piksel pada matrik gambar secara langsung berdasarkan delapan arah sudut. Berikut penghitungan VL setiap pikselnya.

$$VL((x, y)) = \begin{cases} l(x \pm 1, y+w)+l(x \pm 1, y-w)-2l(x \pm 1, y) \\ l(x+w, y \pm 1)+l(x-w, y \pm 1)-2l(x, y \pm 1) \\ l(x \pm 1+w, y \pm 1-w)+l(x \pm 1-w, y \pm 1+w)-2l(x \pm 1, y \pm 1) \\ l(x \pm 1+w, y \mp 1+w)+l(x \pm 1-w, y \mp 1-w)-2l(x \pm 1, y \mp 1) \end{cases}$$

Dimana  $l(x, y)$  adalah intensitas grayscale sesuai dengan alamat setiap piksel pada matrik citra. Kemudian  $w$  adalah diameter tracking pola garis dari satu piksel ke piksel yang lain. Penghitungan VL ini didasarkan pada arah sudut yang dilacak.



Gambar 3.6. Inisialisasi Parameter Line Tracking



Langkah kedua adalah menampung hasil nilai VL yang paling maksimum dan sekaligus menampung satu dari delapan arah sudut tersebut yang mempunyai VL yang paling maksimum dan mengecek piksel mana saja mempunyai  $VL_{Max} > T.T$  adalah threshold untuk menyeleksi  $VL_{Max}$  dari setiap piksel citra. Kemudian menandai piksel tersebut dan mengupdate nilai matrik kepercayaan (CW) setiap piksel terpilih menjadi satu.

		0°	180°	90°	270°	45°	225°	315°	135°
<div style="display: flex; align-items: center;"> <span style="font-size: 2em; margin-right: 5px;">}</span> <span style="writing-mode: vertical-rl; transform: rotate(180deg);">m x n</span> </div>	1	VL1	<u>VLMax</u>	VL1	VL1	VL1	VL1	VL1	VL1
	2	VL2	VL2	VL2	<u>VLMax</u>	VL2	VL2	VL2	VL2
	.	.	.	.	.	.	.	<u>VLMax</u>	.
	.	.	<u>VLMax</u>	.	.	.	.	.	.
	.	<u>VLMax</u>	.	.	.	.	.	.	.
	.	.	.	.	.	<u>VLMax</u>	.	.	.
	.	.	.	.	.	.	<u>VLMax</u>	.	.
	.	.	.	.	.	.	.	<u>VLMax</u>	.
	(m x n)-1	VL(m x n)-1	VL(m x n)-1	VL(m x n)-1	VL(m x n)-1	VL(m x n)-1	VL(m x n)-1	VL(m x n)-1	<u>VLMax</u>
	m x n	VL(m x n)	VL(m x n)	VL(m x n)	<u>VLMax</u>	VL(m x n)	VL(m x n)	VL(m x n)	VL(m x n)

1
8

Gambar 3.7. Update Matrik Cw

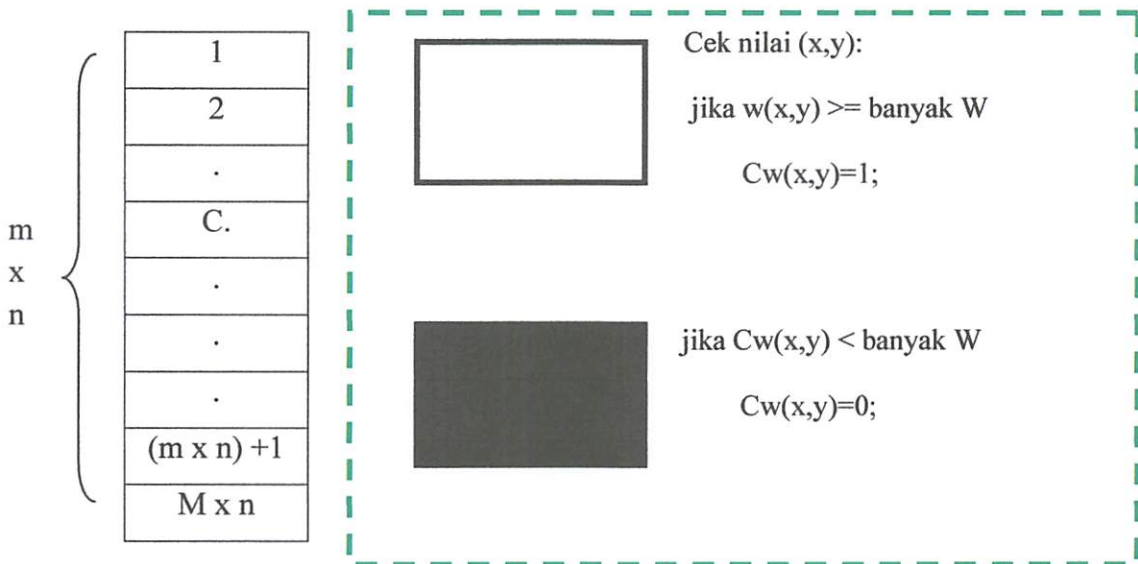
Untuk penentuan piksel tracking yang baru, digunakan rumus sebagai berikut:

$$(Xc, Yc) = (x + r \cos \theta, y + r \sin \theta)$$

Dimana  $(Xc, Yc)$  adalah piksel tracking baru yang dihitung dengan memanfaatkan nilai alamat piksel  $(X, Y)$ , panjang diameter  $(r)$  dan arah sudut  $(\theta)$  yang sebelumnya telah dideteksi sebagai piksel garis.

Proses line tracking di atas akan dijalankan sebanyak  $W$  (banyak diameter garis) yang telah ditentukan. Dan setiap hasil dari 1 ukuran  $W$ , akan selalu ditampung dan di update terus menerus ke dalam matrik kepercayaan  $(Cw)$ . Setelah melewati  $W$  yang terakhir, maka di dapatkan matrik kepercayaan  $(Cw)$  yang akan di update dengan aturan (map quantization rules) sebagai berikut:



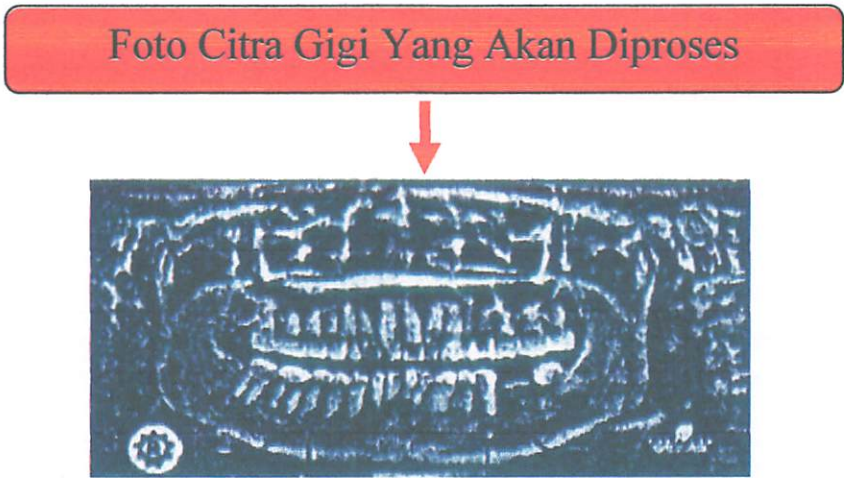


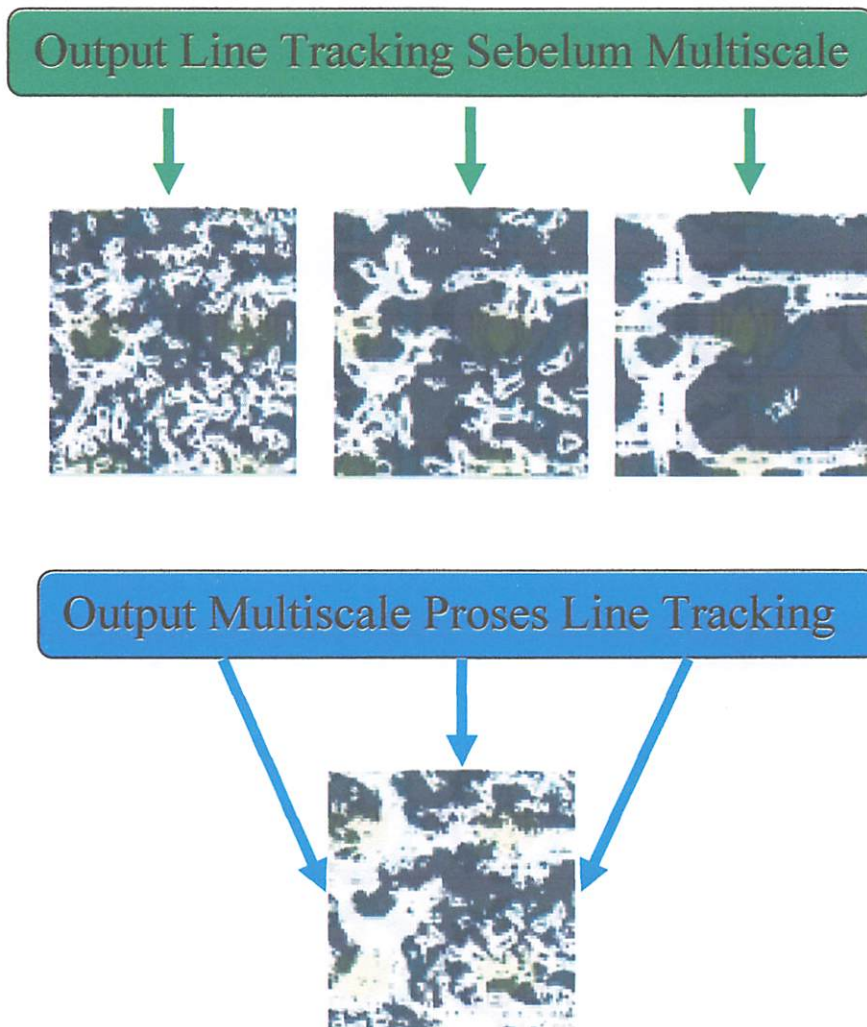
Gambar 3.8. Proses Map Quantization

Hasil akhir dari proses line tracking tersebut merupakan hasil proses *map quantization* dari beberapa piksel citra yang dideteksi sebagai piksel garis. Dan citra *line tracking* ini sudah berupa citra biner.

### Pengambilan Gambar

Pada *dental panoramic radiograph* akan dipilih daerah pada bagian rahang manusia. Dari format citra berupa .tif akan diambil sampel daerah disebelah kanan dan kiri foramen sebesar 128x128 piksel. Untuk lebih jelasnya dapat dilihat pada gambar 9. Letak pengambilan daerah sampel ini berdasarkan rekomendasi dari pakar di bidang kedokteran gigi.





Gambar 3.9. Hasil Output Line Tracking

### 3.4. Desain Perancangan GUI (Graphical User Interface)

Rancangan struktur merupakan suatu prosedur-prosedur program dari sebuah program aplikasi. Pada program Aplikasi pendeteksian penyakit periodontitis kronis pada citra panoramic pada gigi manusia berbasis matlab menggunakan algoritma line strenght dan line tracking. Pada form pertama ini berisi table petugas yang harus diisi oleh petugas untuk memulai aplikasi program yang dapat dilakukan dalam program aplikasi ini. Perancangannya digambarkan sebagai berikut :

#### 3.4.1. Rancangan Database

Di halaman tabel database diberi fasilitas untuk mengakses beberapa menu yang disediakan. Selain itu juga pada halaman ini berfungsi untuk menyimpan

data-data petugas. Gambar rancangan halaman database ditunjukkan pada gambar 3.10:

Nama_Field	Type	Panjang	Keterangan
Kode Petugas	Varchar	20	Nomer Petugas
Nama Petugas	Varchar	30	Nama Petugas
Password	Varchar	20	Password Petugas

Gambar 3.10. Perancangan Database

### 3.4.2. Rancangan Halaman Tabel Petugas 1

Di halaman table petugas ke 1 diberi fasilitas untuk mengakses beberapa menu yang disediakan. Selain itu juga pada halaman ini terkoneksi langsung ke database untuk menyimpan data petugas. Gambar rancangan halaman petugas 1 ditunjukkan pada gambar 3.11.

**APLIKASI PENDETEKSIAN PENYAKIT PERIODONTITIS KRONIS PADA CITRA PANORAMIC PADA GIGI MANUSIA BERBASIS MATLAB MENGGUNAKAN ALGORITMA LINE STRENGHT DAN LINE TRACKING**

**TABEL PETUGAS**

KODE PETUGAS	
NAMA PETUGAS	
ID PETUGAS	

LOGIN	CANCEL
-------	--------

Gambar 3.11. Perancangan Form Menu Data Tabel Petugas 1

### 3.4.3. Rancangan Halaman Tabel Petugas 2

Di halaman tabel petugas ke 2 diberi fasilitas untuk mengakses beberapa menu yang disediakan untuk masuk kedalam proses. Selain itu juga pada halaman ini terkoneksi langsung ke database yang datanya sudah disimpan. Gambar rancangan halaman table petugas 2 ditunjukkan pada gambar 3.12.

**APLIKASI PENDETEKSIAN PENYAKIT PERIODONTITIS KRONIS CITRA PANORAMIC PADA GIGI MANUSIA BERBASIS MATLAB MENGGUNAKAN ALGORITMA LINE STRENGHT DAN LINE TRACKING**

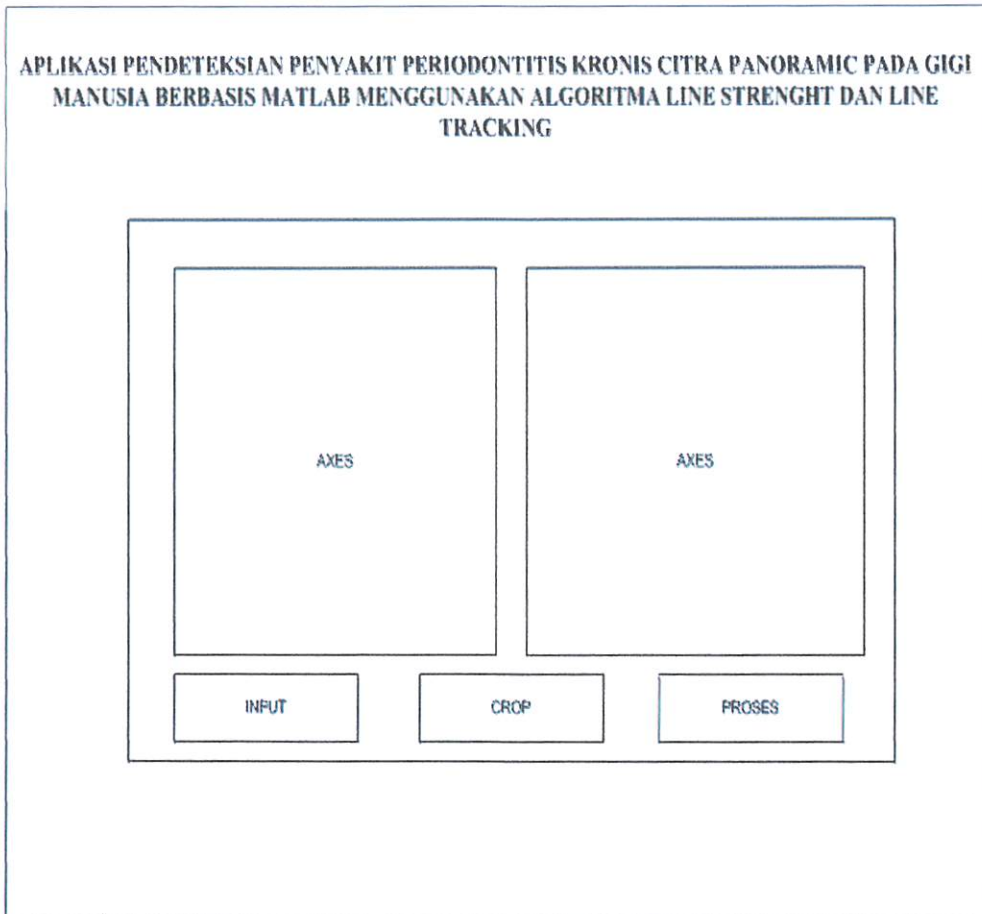
**TABEL PETUGAS**

NAMA PETUGAS	
ID PETUGAS	
LOGIN	CANCEL

Gambar 3.12 Perancangan Form Tabel Petugas 2

### 3.4.4. Rancangan Halaman Memasukkan Citra Untuk Diproses

Di halaman tabel memasukkan citra diberi fasilitas tombol untuk mengakses beberapa menu yang disediakan. Selain itu juga pada halaman ini citra yang telah diinputkan akan dicrop pada bagian yang dibutuhkan untuk diproses di halaman berikutnya. Gambar rancangan halaman memasukkan citra ditunjukkan pada gambar 3.13.

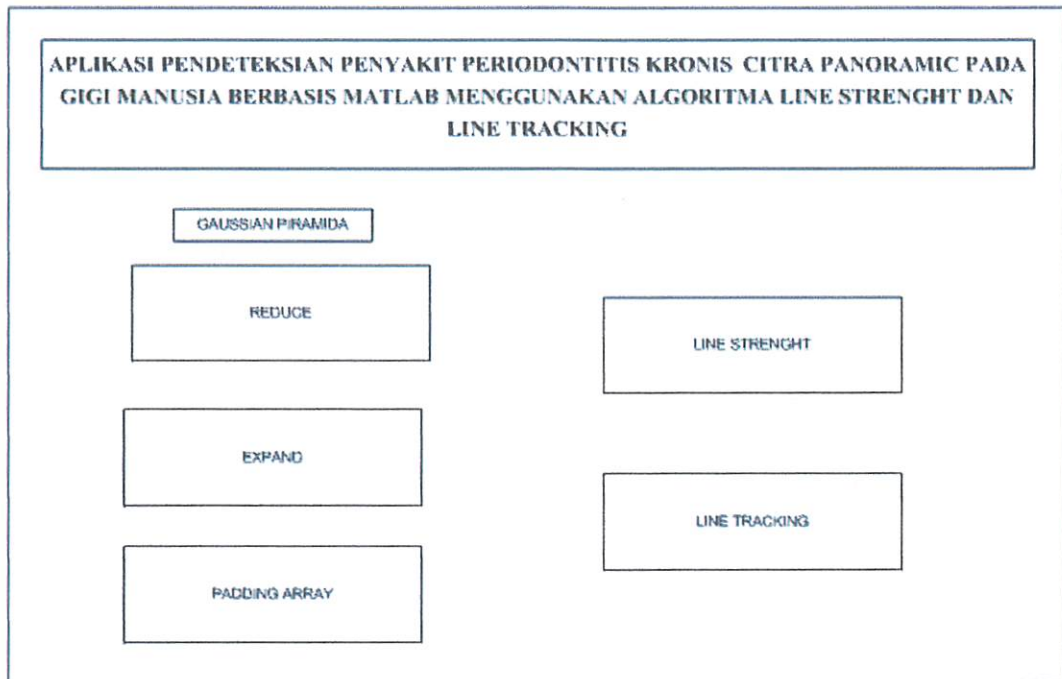


Gambar 3.13. Perancangan Memasukkan Gambar, Crop dan Mulai Proses

### 3.4.5. Rancangan Halaman Memproses Citra

Di halaman tabel proses citra diberi fasilitas untuk memelihat beberapa proses yang disediakan secara otomatis. Selain itu juga pada halaman ini terdapat beberapa proses citra yaitu *Gaussian piramida*, *line strength* dan *line tracking* berfungsi untuk menemukan hasil akhir analisa citra gigi tersebut. Gambar rancangan halaman memproses citra ditunjukkan pada gambar 3.14:

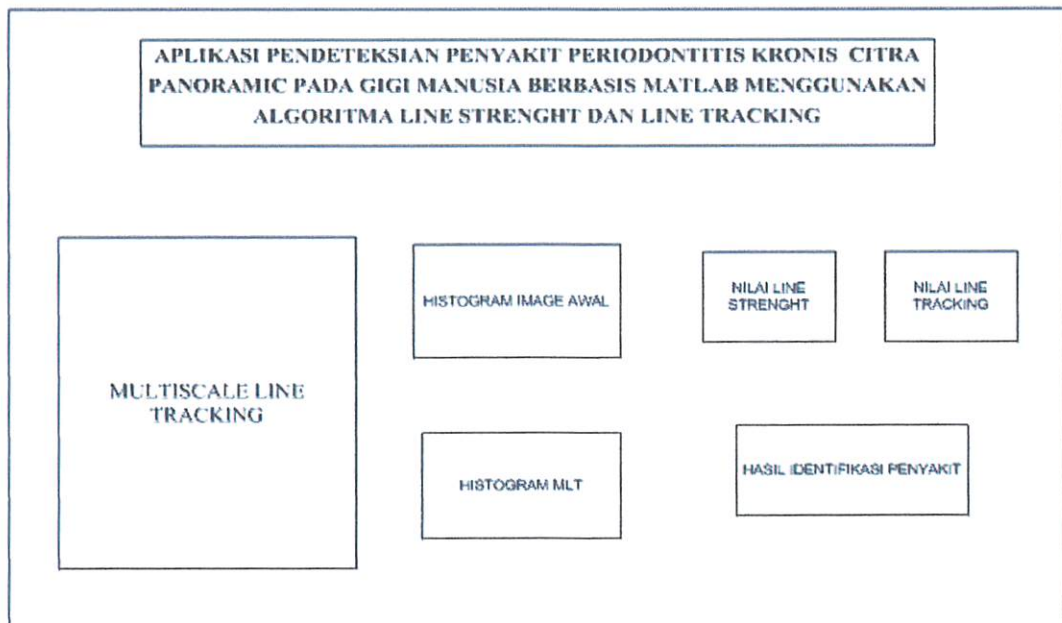




Gambar 3.14. Perancangan Proses Aplikasi Program

### 3.4.6. Rancangan Halaman Hasil Proses Aplikasi Program

Di halaman tabel hasil proses aplikasi program diberi fasilitas untuk memelihat beberapa hasil proses yang disediakan secara otomatis. Selain itu juga pada halaman ini terdapat analisa citra untuk mengetahui bahwa citra gigi yang diproses itu sehat atau sedang mengalami sakit periodontitis kronis. Gambar rancangan halaman memproses citra ditunjukkan pada gambar 3.15



Gambar 3.15. Perancangan Hasil Proses Aplikasi Program

## BAB IV

### HASIL DAN PENGUJIAN

Setelah melakukan tahapan analisis dan perancangan, maka tahapan selanjutnya adalah pengujian program. Pengujian program ini bertujuan untuk mengimplementasikan rancangan yang telah dibuat. Hasil implementasi tersebut adalah sebagai berikut:

#### 4.1 Implementasi GUI

##### 4.1.1 Implementasi Tabel Petugas

Dari rancangan yang telah dibuat pada bab sebelumnya, maka tahap selanjutnya yaitu mengimplementasikannya menjadi sebuah tampilan yang *user friendly*.

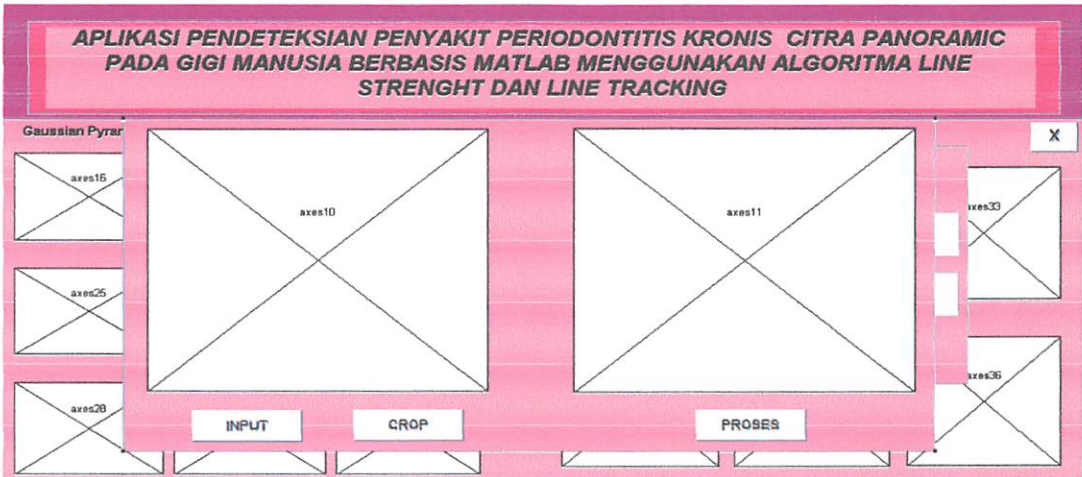
Form tabel petugas yang berupa tampilan menu untuk login ke dalam program aplikasi dapat dilihat seperti pada gambar 4.1 sebagai berikut :

The image shows a screenshot of a MATLAB GUI application. At the top, a title bar contains the text: "APLIKASI PENDETEKSIAN PENYAKIT PERIODONTITIS KRONIS CITRA PANORAMIC PADA GIGI MANUSIA BERBASIS MATLAB MENGGUNAKAN ALGORITMA LINE STRENGHT DAN LINE TRACKING". Below the title bar, there are two overlapping windows. The left window is titled "TABEL PETUGAS BARU" and contains three input fields labeled "KODE PETUGAS :", "NAMA PETUGAS :", and "ID PETUGAS :". Below these fields are two buttons: "SAVE" and "CANCEL". The right window is titled "TABEL PETUGAS" and contains two input fields labeled "NAMA PETUGAS :" and "ID PETUGAS :". Below these fields are two buttons: "LOGIN" and "CANCEL". At the bottom right of the right window, there is a "Detail" button. The background of the GUI is light blue with a grid pattern.

Gambar 4.1 Form Implementasi Tabel Petugas Baru dan Tabel Petugas

### 4.1.2 Implementasi Halaman Input Dan Crop Citra

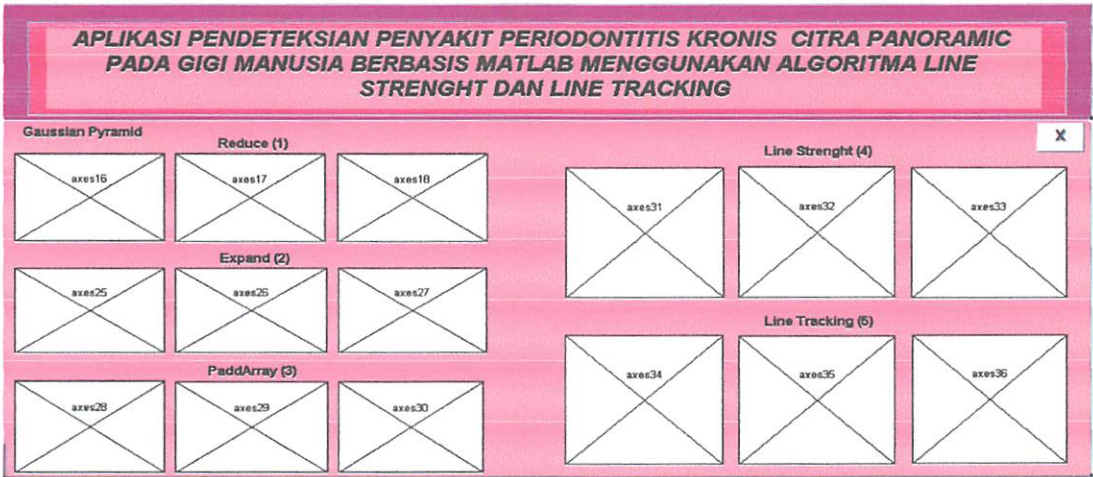
Form halaman input dan *crop* citra yang berupa tampilan menu untuk memulai proses identifikasi penyakit citra dapat dilihat seperti pada gambar 4.2 sebagai berikut :



Gambar 4.2 Form Implementasi Halaman Input Dan Crop Citra

### 4.1.3 Implementasi Halaman Proses Citra

Form halaman proses citra untuk memproses citra dengan beberapa proses dapat dilihat seperti pada gambar 4.3 sebagai berikut :

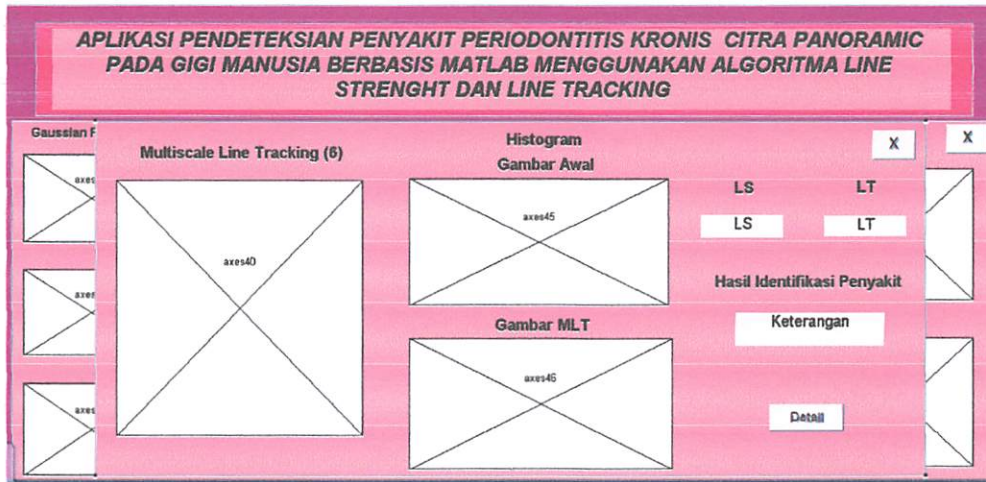


Gambar 4.3 Form Implementasi Halaman Proses Citra



#### 4.1.4 Implementasi Halaman Hasil Akhir Citra

Form halaman hasil akhir citra yang berupa tampilan proses untuk mengetahui hasil identifikasi penyakit pada citra dapat dilihat seperti pada gambar 4.4 sebagai berikut :

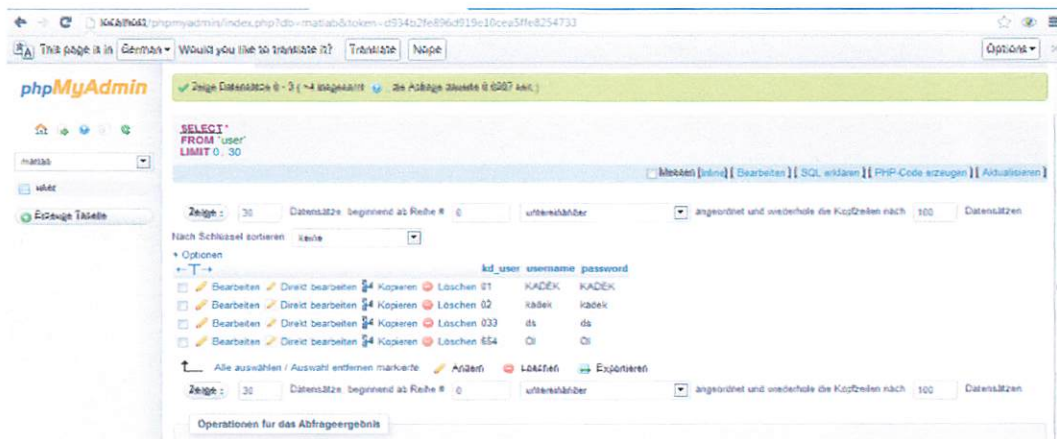


Gambar 4.4 Form Implementasi Halaman Akhir Proses

## 4.2 Pengujian Program Aplikasi

### 4.2.1 Halaman Database MY SQL

Form untuk menyimpan database petugas setelah di inputkan di tabel petugas baru dan outputnya dikeluarkan di tabel petugas untuk memulai program aplikasi. Untuk memulai proses aplikasi dapat dilihat seperti pada gambar 4.5 sebagai berikut :

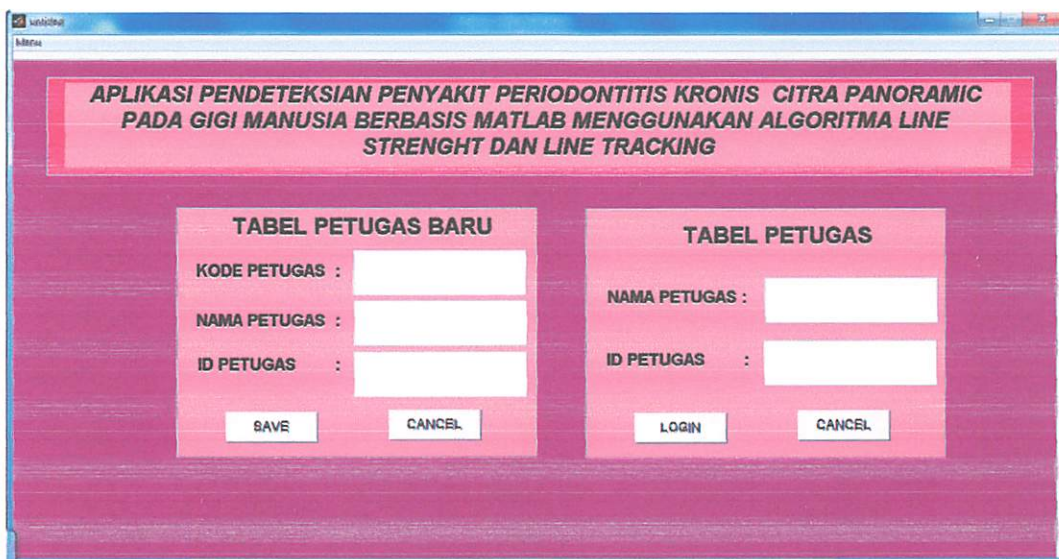


Gambar 4.5 Form Halaman Database MY SQL

#### 4.2.2 Halaman Utama

Dari rancangan yang telah dibuat pada maka tahap selanjutnya yaitu menjadikan sebuah tampilan tabel petugas baru untuk disimpan dalam database dan tahap selanjutnya yaitu mengimplementasikannya menjadi sebuah tampilan tabel petugas untuk login.

Form masukkan tabel petugas baru yang berupa tampilan menu untuk menyimpan data petugas ke dalam file database *mysql* dan Form masukkan tabel petugas yang berupa tampilan menu untuk login data petugas yang sudah disimpan pada database *mysql*. Untuk memulai proses aplikasi dapat dilihat seperti pada gambar 4.6 sebagai berikut :



Gambar 4.6 Form Aplikasi Tabel Petugas Baru dan Tabel Petugas

1. Form tabel petugas baru ini terdiri dari menu aplikasi sebagai berikut:
  - a. Kode petugas: menu ini untuk mengisi kode petugas yang akan disimpan dalam database *mysql*.
  - b. Nama petugas: menu ini untuk mengisi nama petugas yang akan disimpan di database *mysql*.
  - c. Id petugas: menu ini untuk mengisi id petugas yang akan disimpan dalam database *mysql*.

- d. **Save:** menu aplikasi yang digunakan untuk menyimpan file hasil pengisian menu-menu yang ada dalam tabel petugas baru.
- e. **Cancel:** menu aplikasi yang digunakan untuk menghapus pengisian yang tidak jadi di proses untuk disimpan dalam database.

2. Form tabel petugas baru ini terdiri dari menu aplikasi sebagai berikut:

- a. **Kode petugas:** menu ini untuk mengisi kode petugas yang akan disimpan dalam database *mysql*.
- b. **Nama petugas:** menu ini untuk mengisi nama petugas yang akan disimpan di database *mysql*.
- c. **Id petugas:** menu ini untuk mengisi id petugas yang akan disimpan dalam database *mysql*.
- d. **Save:** menu aplikasi yang digunakan untuk menyimpan file hasil pengisian menu-menu yang ada dalam tabel petugas baru.
- e. **Cancel:** menu aplikasi yang digunakan untuk menghapus pengisian yang tidak jadi di proses untuk disimpan dalam database.

#### **4.2.3 Halaman Input Dan *Crop* Citra**

Setelah memasukkan nama dan id petugas dalam tabel petugas akan muncul tampilan input citra dan tampilan *crop* citra digunakan untuk memulai program aplikasi yang pertama memasukkan citra dan setelah itu *dicrop* seperti pada gambar 4.7:

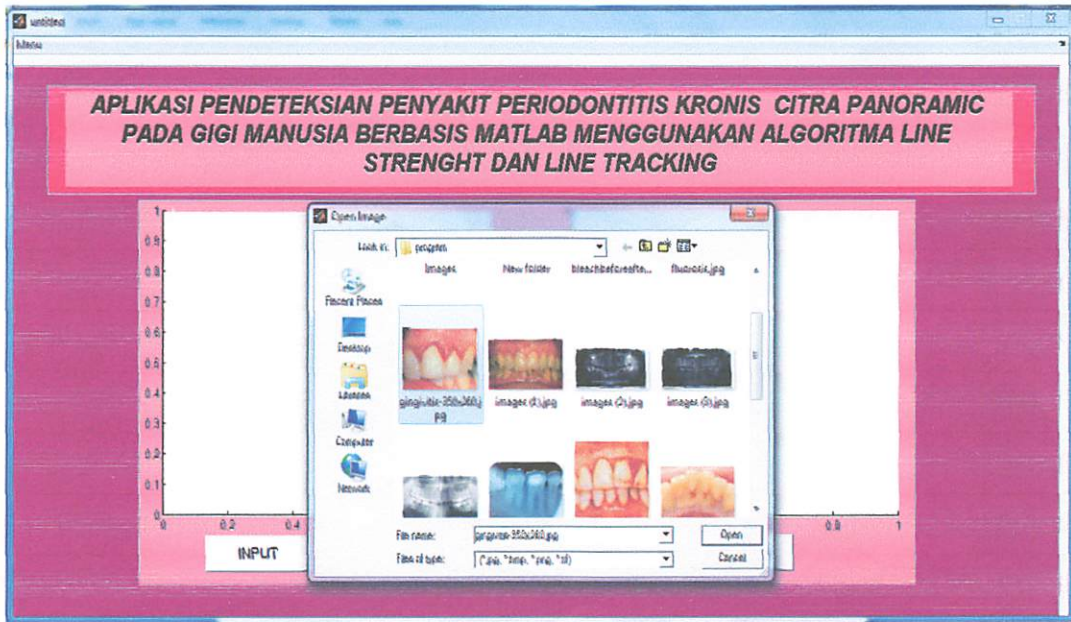




Gambar 4.7 Tampilan Halaman Input Citra Dan *Crop*

#### 4.2.4 Memilih Citra

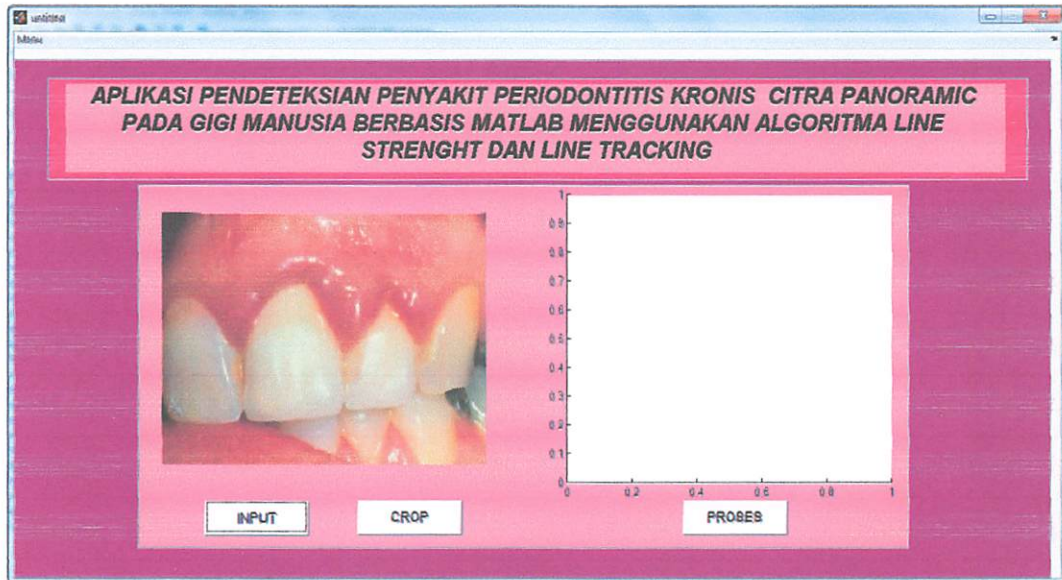
Untuk memilih citra dan membuka file image yang akan digunakan sebagai media aplikasi yang akan diproses diklik pada menu browse dan akan muncul tampilan seperti pada gambar 4.8:



Gambar 4.8. Tampilan Menu Memilih *File Image*

#### 4.2.5 Inputan Citra

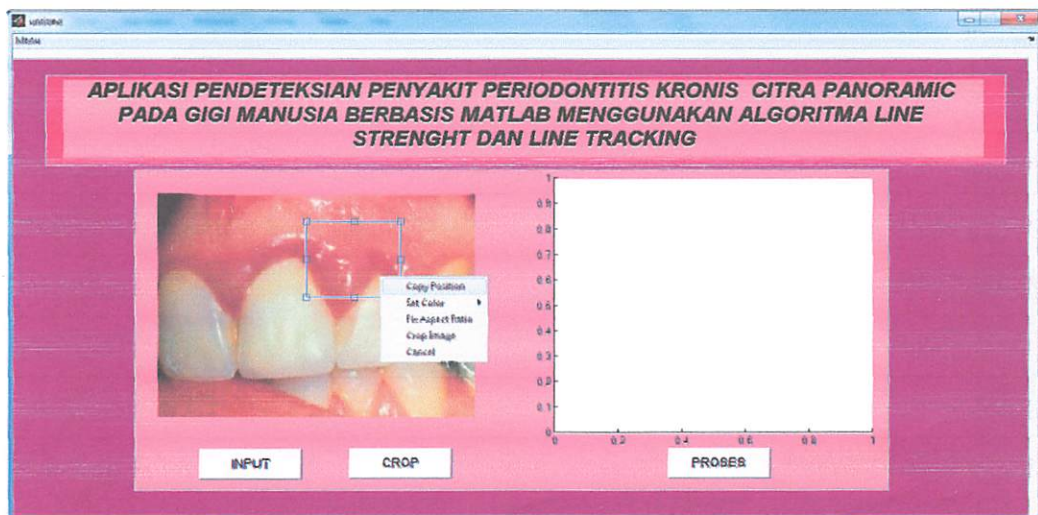
Setelah file image dipilih dan diklik open maka akan muncul preview dari file tersebut seperti yang tampak pada gambar 4.9:



Gambar 4.9 Preview File Image Yang Dijadikan Sebagai Awal Proses

#### 4.2.6 Proses *Crop* Citra

Untuk melakukan proses aplikasi diklik menu *crop* untuk memilih bagian mana yang diperlukan pada *image*, maka muncul gambar dari hasil proses *crop* seperti pada gambar 4.10:



Gambar 4.10 Gambar Proses Mengecrop Citra



#### 4.2.7 Hasil Crop Citra

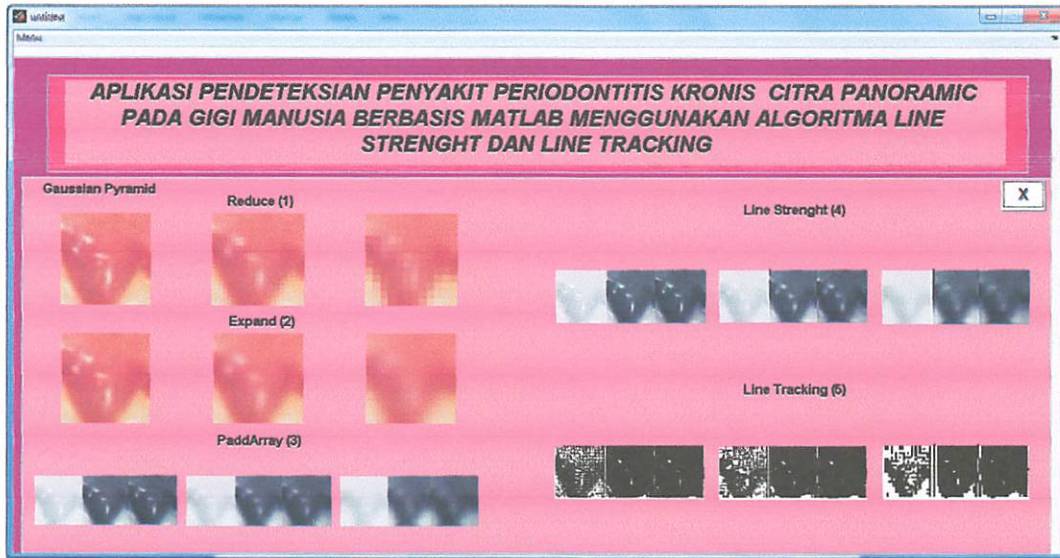
Setelah menu *crop* selesai diklik maka akan muncul hasil *crop* tampilan dan memulai proses *image* untuk mengetahui identifikasi penyakit *periodontitis kronis* seperti pada gambar 4.11:



Gambar 4.11 Hasil Crop Citra

#### 4.2.8 Proses Citra

Setelah hasil *crop* selanjutnya yaitu memulai proses detailnya yaitu *gaussian pyramid* yang ada dua proses, maka muncul gambar dari proses pertama *reduce* yang berarti menjadikan image dibagi menjadi tiga level semakin besar level image semakin kecil image tersebut dan untuk menghilangkan noise pada image. Proses kedua *expand* adalah proses *gaussian pyramid* kedua yang berarti mengembalikan image mulai level kecil menjadi level image menjadi besar dan tetap dibagi tiga juga untuk memperhalus citra setelah mengalami proses *reduce*. Proses ketiga *padding array* adalah menambahkan piksel dalam baris dan kolom juga untuk memperbesar image. Proses keempat selanjutnya yaitu *line strength* adalah untuk mendeteksi pola garis pada citra dengan sudut tertentu. Proses kelima selanjutnya yaitu *line tracking* adalah metode integrasi untuk melacak garis dari setiap piksel pada citra dengan diameter tertentu Seperti pada gambar 4.12:

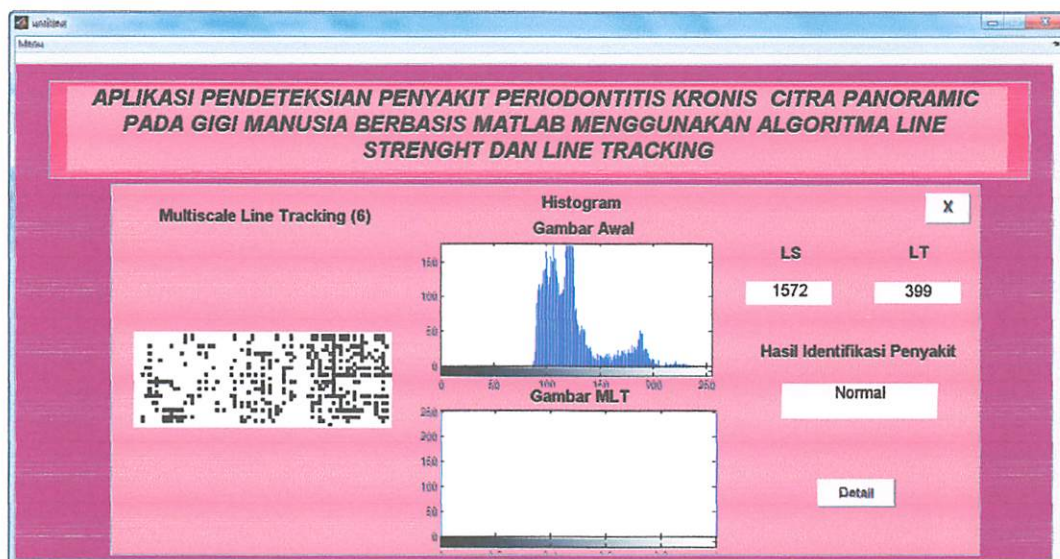


Gambar 4.12 Halaman Proses Citra

#### 4.2.9 Akhir Proses

Setelah proses selesai selanjutnya maka dilakukan proses *multiscale line tracking* yang artinya menggabungkan proses *line strength* dan *line tracking* menjadi satu. Setelah dilakukan proses *multiscale line tracking*, dilakukanlah perbedaan *histogram* gambar awal dan gambar MLT untuk mengetahui bahwa sudah memenuhi *multiscale line tracking*. Selanjutnya melakukan proses *multiscale line tracking* dan *histogram*, maka dilakukan penilaian pada image *line strength* dan *line tracking*. Terakhir melakukan proses untuk mengetahui hasil identifikasi citra yang diproses apakah citra tersebut mengidap periodontitis kronis atau normal. Seperti pada gambar 4.13:





Gambar 4.13 Halaman Hasil Akhir

### 4.3 Hasil Pengujian Program

Pengujian aplikasi program algoritma *line strength* dan *line tracking* dilakukan dengan beberapa proses yaitu menggunakan *gaussian pyramid* meliputi *reduce*, *expand* dan *padarray*. Proses pengujian dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil Pengujian Implementasi Program

No	Proses Pengujian	Hasil
1	Membuka halaman tabel petugas dan melakukan login. Setelah masuk ke halaman tabel petugas, selanjutnya kehalaman utama yang berisi input gambar dan <i>Crop</i> image.	Bisa masuk ke halaman utama atau berikutnya.
2	Membuka halaman utama yang berisi tentang input image, <i>Crop</i> dan proses image yang sudah <i>diCrop</i> sesuai dengan image yang dibutuhkan.	Halaman selanjutnya terbuka dan mulai pemrosesan hasil gambar <i>Crop</i> yang dibutuhkan.



3	Membuka halaman hasil akhir program aplikasi meliputi <i>multiscale line tracking</i> , <i>histogram</i> , <i>nilai line strength</i> , <i>nilai line tracking</i> dan identifikasi penyakit periodontitis yang merupakan akhir program.	Halaman ini merupakan hasil akhir dari aplikasi program.
4	Membuka halaman detail untuk mengetahui proses sebelum hasil akhir yang meliputi <i>Gaussian pyramid (expand dan reduce)</i> , <i>padd array</i> , <i>line strength</i> dan <i>line tracking</i> .	halaman ini merupakan proses sebelum <i>multiscale line tracking</i> .

Pada penelitian ini, data yang digunakan adalah dental panoramic radiograph. Untuk spesifikasi perangkat lunak yang digunakan dalam implementasi perangkat lunak yakni Microsoft windows 2007, visio professional serta matlab 7.7.0.

Data di uji coba dengan enam macam orientasi sudut sebagai referensi deteksi penyakit kronis yang telah dilakukan. Hal ini bertujuan untuk mendapatkan sudut orientasi dan nilai threshold identifikasi dengan nilai akurasi, sensitibity dan specificity tertinggi. Proses uji coba dilakukan dengan cara mengubah nilai threshold identifikasi secara iterative. Nilai dari threshold di dapatkan dari hasil rata-rata nilai kepadatan tulang pada setiap citra. Proses iteratif ini juga dilakukan pada semua sudut orientasi.

Hasil deteksi pola garis yang dilakukan dengan metode line strength terhadap gambar 4.15. yang merupakan hasil proses expand pada proses gaussian pyramid dapat dilihat pada gambar 4.13, sedangkan pada gambar 4.8. deteksi pola garis yang dilakukan dengan integrasi metode line strength dengan line tracking berbasis multiscale tampak lebih jelas dan tidak banyak piksel yang hilang jika dibandingkan dengan hasil binerisasi secara langsung dari citra line strength. Setelah melakukan proses deteksi pola garis, maka dilakukan uji coba deteksi pola garis dengan enam macam orientasi sudut. Jika bone value lebih kecil sama dengan threshold maka diagnosanya adalah periodontitis dan jika bone value lebih

besar dari threshold maka diagnosanya normal. Proses pengujian dapat dilihat pada tabel 4.2:

Tabel 4.2 Hasil Identifikasi Berdasarkan Nilai LS dan Nilai LT

NO	Nama Citra	Spesifikasi	Nilai LS	Nilai LT	Identifikasi penyakit
1	Citra 1	Nama File: Fluorosis Format File: JPG Ukuran File: 365x252 Jenis Citra: Foto	281	878	Periodontitis
2	Citra 2	Nama File: Gingivitis Format File: JPG Ukuran File: 350x260 Jenis Citra: Foto	3425	698	Normal
3	Citra 3	Nama File: Image (1) Format File: JPG Ukuran File: 201x123 Jenis Citra: Foto	238	169	Periodontitis
4	Citra 4	Nama File: Image (2) Format File: JPG Ukuran File: 317x159 Jenis Citra: X-Ray	289	1156	Periodontitis
5	Citra 5	Nama File: Image (3) Format File: JPG Ukuran File: 317x159 Jenis Citra: X-Ray	281	1776	Periodontitis
6	Citra 6	Nama File: Image (4) Format File: JPG Ukuran File: 300x168 Jenis Citra: X-Ray	1389	2286	Normal
7	Citra 7	Nama File: Image (5)	626	2433	Normal

		Format File: JPG Ukuran File: 264x191 Jenis Citra: X-Ray			
8	Citra 8	Nama File: Image Format File: JPG Ukuran File: 120x120 Jenis Citra: Foto	318	23	Periodontitis
9	Citra 9	Nama File: Karang Gigi Format File: JPG Ukuran File: 208x 142 Jenis Citra: Foto	836	581	Normal
10	Citra 10	Nama File: Karies1 Format File: JPG Ukuran File: 329x153 Jenis Citra: Foto	382	443	Periodontitis

Berdasarkan table 4.2 di atas terlihat bahwa prosentase perbandingan nilai LS dan nilai LT untuk menentukan identifikasi penyakit periodontitis atau normal dapat dilihat dibawah ini.

1. Identifikasi Periodontitis Kronis

Nilai LS < 500  
 Nilai LT < 500

} Disesuaikan dengan nilai threshold LS dan LT

2. Identifikasi Normal

Nilai LS > 500  
 Nilai LT > 500

} Disesuaikan dengan nilai threshold LS dan LT

Pengujian dilakukan pada sepuluh citra dengan spesifikasi yang berbeda seperti yang ditunjukkan pada tabel 4.2. Dari sepuluh pengujian tersebut, menunjukkan bahwa nilai rata-rata nilai LS ditiap citra yang bernilai dibawah 500

diidentifikasi sebagai citra periodontitis dan diidentifikasi normal apabila rata-rata nilai LS diatas 500. Identifikasi citra periodontitis didasarkan pada penentuan threshold yang bernilai kurang dari 500 untuk nilai line strength (LS) karena semakin kecil pixel pada citra menunjukkan nilai bone value rendah dan ditentukan normal apabila nilai threshold diatas 500 untuk nilai line strength (LS).

Pengujian identifikasi normal pada sepuluh citra yang berbeda pada tabel diatas menunjukkan bahwa rata-rata nilai line tracking (LT) ditiap citra diatas 500 diidentifikasi sebagai citra normal dan untuk rata-rata nilai dibawah 500 citra diidentifikasi sebagai periodontitis. Identifikasi citra normal berdasarkan dari penentuan nilai threshold diatas 500 untuk nilai line tracking (LT) karena semakin besar nilai pixel pada citra menunjukkan nilai bone value tinggi dan dikatakan citra periodontitis berdasarkan nilai threshold dibawah 500 untuk nilai line tracking (LT).

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil penelitian ini dapat disimpulkan beberapa hal sebagai berikut:

1. Berdasarkan hasil pengujian didapatkan kesimpulan bahwa citra yang diidentifikasi periodontitis mewakili rata-rata nilai LS dibawah 500 dan nilai rata-rata LT dibawah 500 berdasarkan threshold bone value rendah.
2. Diidentifikasi normal mewakili rata-rata nilai LS diatas 500 dan nilai rata-rata LT diatas 500 berdasarkan threshold bone value tinggi.

#### **5.2 Saran**

Saran yang didapatkan dari percobaan dan evaluasi yang telah dilakukan adalah sebagai berikut:

1. Untuk pengembangan aplikasi selanjutnya sebaiknya ditambahkan fitur-fitur preprosesing data citra dental panoramic radiograph dengan mengurangi efek pencahayaan yang kurang merata.
2. Dalam pengembangan aplikasi selanjutnya diharapkan proses cropping dibagian gigi dapat dilakukan secara otomatis dari program.
3. Dalam pengembangan data aplikasi selanjutnya diharapkan menggunakan dataset yang dapat dilakukan secara optimal yang lebih banyak dan bervariasi.

## DAFTAR PUSTAKA

- [1] Achmad, Balsa, M.Sc.E, Firdausy, Kartika,S.T., M.T.,*Teknik Pengolahan Citra Digital Menggunakan DELPHI*, Ardi Publising, Yogyakarta, 2005.
- [2] Ahmad, Usman, *Pengolahan Citra Digital dan Teknik Pemrogramannya*, Graha Ilmu, Yogyakarta, 2005.
- [3] Munir, Rinaldi, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika, Bandung, 2004.
- [4] Pratt, William K. *Digital Image Processing Second Edition*. New York: John Wiley & Sons, Inc., 1991.
- [5] Irawan, Feriza A. *Pemrograman Matlab*, Mediakom, Yogyakarta, 2012.
- [6] Tim Asisten Laboratorium Multimedia dan Pengolahan Citra Digital. *Modul Praktikum Matlab*, 2011. ITN Malang.



PROGRAM STUDI TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL  
MALANG

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Kadek Suarjuna Batu Bulan  
Nim : 09.18.117  
Program Studi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul:

**“Aplikasi Pendeteksian Penyakit Periodontitis Kronis Citra Panoramic Pada Gigi Manusia Berbasis Matlab Menggunakan Algoritma Line Streight Dan Line Tracking”**

Adalah Skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 06 Maret 2013

Yang membuat pernyataan

METERAI  
TEMPER  
PAJAK MEMBANDUNG  
TGL:  
43D72ABF011836481  
ENAM RIBU RUPIAH  
6000 DJP  
Kadek Suarjuna



**INSTITUT TEKNOLOGI NASIONAL MALANG**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**PROGRAM STUDI TEKNIK INFORMATIKA S-1**  
**Jl. Raya Karanglo Km. 2 Malang**

---

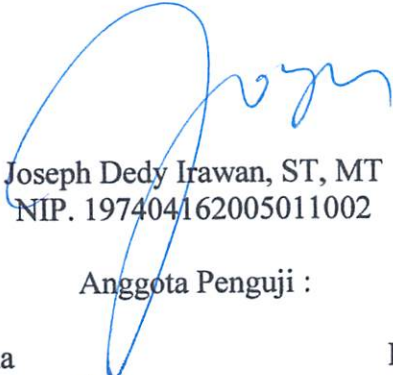
**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Kadek Suarjuna Batu Bulan  
NIM : 0918117  
Jurusan : Teknik Informatika S-1  
Judul : Aplikasi pendeteksian Penyakit Periodontitis Kronis Citra Panoramic  
Pada Gigi Manusia Berbasis Matlab Menggunakan Algoritma Line  
Strenght Dan Line Tracking

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :  
Hari : Senin  
Tanggal : 18 Februari 2013  
Nilai : 89.07 (A)

Panitia Ujian Skripsi :

Ketua Majelis Penguji

  
Joseph Dedy Irawan, ST, MT  
NIP. 197404162005011002

Anggota Penguji :

Penguji Pertama



Karina Auliasari, ST, M.Eng  
NIP.P. 1031000426

Penguji Kedua



Sandy Nataly Mantja, S.Kom  
NIP.P. 1030800418





**INSTITUT TEKNOLOGI NASIONAL MALANG**  
Fakultas Teknologi Industri  
Program Studi Teknik Informatika S1  
Jl. Raya Karanglo Km. 2 Malang

## FORMULIR BIMBINGAN SKRIPSI

Nama : Kadek Suarjuna Batu Bulan  
Nim : 09.18.117  
Masa Bimbingan : 15 Oktober 2012 s/d 15 April 2013  
Judul Skripsi : Aplikasi Pendeteksian Penyakit Periodontitis Kronis  
Citra Panoramic Pada Gigi Manusia Berbasis Matlab  
Menggunakan Algoritma Line Strenght Dan Line  
Tracking

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	25-12-2012	Perbaikan Tampilan Program	
2	26-12-2012	Proses Penyusunan Program	
3	28-12-2012	Memperbaiki Fitur-fitur Didalam Program	
4	29-12-2012	Memperjelas Kata-Kata Diprogram	
5	03-01-2013	Merevisi Laporan Bab III	
6	04-01-2013	Memperbaiki Paragraf dan Tulisan Di Bab III	
7	07-01-2013	Merevisi Laporan Bab IV	
8	08-01-2013	Memperjelas Tampilan Print Screen Pada Laporan	
9	10-01-2013	Program dan Laporan FIX	
10			

Malang, 16 Januari 2013

Dosen Pembimbing I

**Ir. Yusuf Ismail Nahkoda, MT**  
NIP. 1018800189



**INSTITUT TEKNOLOGI NASIONAL MALANG**  
 Fakultas Teknologi Industri  
 Program Studi Teknik Informatika S1  
 Jl. Raya Karanglo Km. 2 Malang

## FORMULIR BIMBINGAN SKRIPSI

Nama : Kadek Suarjuna Batu Bulan  
 Nim : 09.18.117  
 Masa Bimbingan : 15 Oktober 2012 s/d 15 April 2013  
 Judul Skripsi : Aplikasi Pendeteksian Penyakit Periodontitis Kronis  
 Citra Panoramic Pada Gigi Manusia Berbasis Matlab  
 Menggunakan Algoritma Line Strenght Dan Line  
 Tracking

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	25-12-2012	Memperbaiki Bab III	
2	26-12-2012	Lebih Memperjelas Output Program	
3	28-12-2012	Warna Pada Aplikasi Program Terlalu Kontras	
4	29-12-2013	Memperbaiki Font Dan Ukuran Pada Program	
5	04-01-2013	Merevisi Bab III dan Bab IV	
6	08-01-2013	Hasil Tampilan Gambar Program Diperjelas	
7	10-01-2013	Perbaikan Pada Paragraf Laporan	
8	14-01-2013	Program dan Laporan FIX	

Malang, 16 Januari 2013  
 Dosen Pembimbing II

**Suryo Adi Wibowo, ST**  
 NIP. 1031000438



## FORMULIR PERBAIKAN SKRIPSI

Nama : Kadek Suarjuna Batu Bulan  
NIM : 09.18.117  
Prodi : Teknik Informatika S-1  
Judul : Aplikasi Pendeteksian Penyakit Periodontitis Kronis Citra  
Panoramic Pada Gigi Manusia Berbasis Matlab Menggunakan  
Algoritma Line Strenght Dan Line Tracking

Tanggal	Penguji	Uraian	Paraf
18 Februari 2013	I	<ul style="list-style-type: none"><li>- Bab I, Latar Belakang diperbaiki</li><li>- Tinjauan pustaka dihilangkan</li><li>- Poin 3.5 dihilangkan</li><li>- BAB IV, diperbaiki</li><li>- Bab 5, kesimpulan dan saran</li><li>- Kata pengantar</li></ul>	
18 Februari 2013	II	<ul style="list-style-type: none"><li>- Landasan Teori</li><li>- Penulisan harus menggunakan angka/ abjad tidak boleh button</li><li>- Hapus rencana implementasi</li><li>- Kesimpulan</li></ul>	

### Anggota Penguji :

#### Dosen Penguji I

Karina Auliasari, ST, M.Eng  
NIP.P. 10310000426

#### Dosen Penguji II

Sandy Nataly Mantja, S.Kom  
NIP.P. 1030800418

### Mengetahui

#### Dosen Pembimbing I

Ir. Yusuf Ismail Nahkoda, MT  
NIP. Y. 10118800189

#### Dosen Pembimbing II

Suryo Adi Wibowo, ST  
NIP. P. 1031000438

# LAMPIRAN

## Script Program Aplikasi

```
function varargout = untitled(varargin)
% UNTITLED M-file for untitled.fig
%   UNTITLED, by itself, creates a new UNTITLED or raises the
existing
%   singleton*.
%
%   H = UNTITLED returns the handle to a new UNTITLED or the
handle to
%   the existing singleton*.
%
%   UNTITLED('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in UNTITLED.M with the given input
arguments.
%
%   UNTITLED('Property','Value',...) creates a new UNTITLED or
raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before untitled_OpeningFcn gets called.
An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to untitled_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help untitled

% Last Modified by GUIDE v2.5 07-Jan-2013 22:22:19

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @untitled_OpeningFcn, ...
                  'gui_OutputFcn',  @untitled_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```



```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before untitled is made visible.
function untitled_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to untitled (see VARARGIN)

% Choose default command line output for untitled
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes untitled wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = untitled_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
project=guidata(gcbo);
cla(project.axes10,'reset');
cla(project.axes11,'reset');
cla(project.axes16,'reset');
cla(project.axes17,'reset');
cla(project.axes18,'reset');
cla(project.axes25,'reset');
cla(project.axes26,'reset');
cla(project.axes27,'reset');
cla(project.axes28,'reset');
cla(project.axes29,'reset');
cla(project.axes30,'reset');
cla(project.axes31,'reset');
cla(project.axes32,'reset');
cla(project.axes33,'reset');
cla(project.axes34,'reset');
cla(project.axes35,'reset');
cla(project.axes36,'reset');
cla(project.axes40,'reset');
cla(project.axes45,'reset');
cla(project.axes46,'reset');
[namafile,direktori]=uigetfile({'*.jpg;*.bmp;*.png;*.tif'}, 'Open
Image');
if isequal(namafile,0)
return;
end
eval(['cd '' direktori '';']);
Gambar=imread(namafile);
set(project.figure1,'CurrentAxes',project.axes10);
set(imshow(Gambar));colormap('gray');
set(project.axes10,'Userdata',Gambar);
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```



```

project=guidata(gcbo);
Gambar=get(project.axes10,'Userdata');
GambarCrop=imcrop(Gambar);
set(project.figure1,'CurrentAxes',project.axes11);
set(imshow(GambarCrop));colormap('gray');
set(project.axes11,'Userdata',GambarCrop);
set(project.figure1,'CurrentAxes',project.axes45);
imhist(rgb2gray(GambarCrop));
% -----
----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% -----
----
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% -----
----
function Untitled_3_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of
edit4 as a double
% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
project=guidata(gcbo);
set(project.uipanel20,'Visible','on');
set(project.uipanel12,'Visible','off');
I0 =get(project.axes11,'Userdata');
I1 = impyramid(I0, 'reduce');
set(project.figure1,'CurrentAxes',project.axes16);
set(imshow(I1));colormap('gray');
set(project.axes16,'Userdata',I1);
I2 = impyramid(I1, 'reduce');
set(project.figure1,'CurrentAxes',project.axes17);
set(imshow(I2));colormap('gray');
set(project.axes17,'Userdata',I2);
I3 = impyramid(I2, 'reduce');
set(project.figure1,'CurrentAxes',project.axes18);
set(imshow(I3));colormap('gray');
set(project.axes18,'Userdata',I3);
E1 = impyramid(I1, 'expand');
set(project.figure1,'CurrentAxes',project.axes25);
set(imshow(E1));colormap('gray');
set(project.axes25,'Userdata',E1);
E2 = impyramid(I2, 'expand');
set(project.figure1,'CurrentAxes',project.axes26);
set(imshow(E2));colormap('gray');
set(project.axes26,'Userdata',E2);
E3 = impyramid(I3, 'expand');
set(project.figure1,'CurrentAxes',project.axes27);
set(imshow(E3));colormap('gray');
set(project.axes27,'Userdata',E3);
% GambarCrop=get(project.axes11,'Userdata');
% GambarThreshold=threshold(GambarCrop,128);
% set(project.figure1,'CurrentAxes',project.axes4);
% set(imshow(GambarThreshold));colormap('gray');
% set(project.axes4,'Userdata',GambarThreshold);

```

```

PA1=get(project.axes25, 'Userdata');
% figure, imshow(PA1);
[r1 c1]=size(PA1);
for x1=1:r1
    for y1=1:c1
        PA11(x1,y1) = PA1(x1,y1);
    end
end
HPA1= padarray(PA11, [2 2], 'symmetric', 'both');
% figure, imshow(HPA1);
set(project.figure1, 'CurrentAxes', project.axes28);
set(imshow(HPA1)); colormap('gray');
set(project.axes28, 'Userdata', HPA1);
PA2=get(project.axes26, 'Userdata');
% figure, imshow(PA1);
[r2 c2]=size(PA2);
for x2=1:r2
    for y2=1:c2
        PA22(x2,y2) = PA2(x2,y2);
    end
end
HPA2= padarray(PA22, [2 2], 'symmetric', 'both');
% figure, imshow(HPA1);
set(project.figure1, 'CurrentAxes', project.axes29);
set(imshow(HPA2)); colormap('gray');
set(project.axes29, 'Userdata', HPA2);
PA3=get(project.axes27, 'Userdata');
% figure, imshow(PA1);
[r3 c3]=size(PA3);
for x3=1:r3
    for y3=1:c3
        PA33(x3,y3) = PA3(x3,y3);
    end
end
HPA3= padarray(PA33, [2 2], 'symmetric', 'both');
% figure, imshow(HPA1);
set(project.figure1, 'CurrentAxes', project.axes30);
set(imshow(HPA3)); colormap('gray');
set(project.axes30, 'Userdata', HPA3);
R1=get(project.axes28, 'Userdata');
HR1 = imrotate(R1, 45);
set(project.figure1, 'CurrentAxes', project.axes31);
set(imshow(HR1)); colormap('gray');
set(project.axes31, 'Userdata', HR1);
R2=get(project.axes29, 'Userdata');
HR2 = imrotate(R2, 45);
set(project.figure1, 'CurrentAxes', project.axes32);
set(imshow(HR2)); colormap('gray');
set(project.axes32, 'Userdata', HR2);
R3=get(project.axes30, 'Userdata');
HR3 = imrotate(R3, 45);
set(project.figure1, 'CurrentAxes', project.axes33);
set(imshow(HR3)); colormap('gray');

```



```

[row col]=size(HHTa);
count=0;
total=0;
for x=1:row
    for y=1:col
        total=total+1;
        if(HHTa(x,y)>0)
            count=count+1;
        else

            end
        end
    end
end
% if count<=500
%     set(project.text53,'String','Periodontitis');
% else
%     set(project.text61,'String',total-count);
%     set(project.text58,'String',count);
% end
if count<=500
    set(project.text53,'String','Periodontitis');
else
    set(project.text53,'String','Normal');
end

set(project.figure1,'CurrentAxes',project.axes46);
imhist(HHTa);

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit5 as text
%         str2double(get(hObject,'String')) returns contents of
edit5 as a double
% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
project=guidata(gcbo);
% Parameter Database
    host = 'localhost'; % isi dengan host, misal : localhost
    user = 'root'; % isi dengan user, misal : root
    password = ''; % isi dengan password, misal : iloveyou
    dbName = 'matlab'; % isi dengan nama database, misal :
databaseku
    % Parameters JDBC
    jdbcString = sprintf('jdbc:mysql://%s/%s', host, dbName);
    jdbcDriver = 'com.mysql.jdbc.Driver';
    % Buat Koneksi Database
    dbConn = database(dbName, user , password, jdbcDriver,
jdbcString);
    kd=get(project.kode, 'String');
    usnm=get(project.username, 'String');
    psswd=get(project.password, 'String');
    exdata = {kd, usnm, psswd};
    colnames = {'kd_user', 'username', 'password'};
    fastinsert(dbConn, 'user', colnames, exdata);
% Insert data into Table1.
% exec(dbConn, 'INSERT INTO user(kd_user,username,password)
VALUES("jhjh", "hgjhgusrnm", "psswd")');
close(dbConn);
% --- Executes on button press in pushbutton17.
function pushbutton15_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton17.
function pushbutton17_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
project=guidata(gcbo);
set(project.uipanel12, 'Visible', 'on');
set(project.uipanel10, 'Visible', 'off');
set(project.uipanel17, 'Visible', 'off');
host = 'localhost'; % isi dengan host, misal : localhost
    user = 'root'; % isi dengan user, misal : root
    password = ''; % isi dengan password, misal : iloveyou
    dbName = 'matlab'; % isi dengan nama database, misal :
databaseku
    % Parameters JDBC
    jdbcString = sprintf('jdbc:mysql://%s/%s', host, dbName);
    jdbcDriver = 'com.mysql.jdbc.Driver';

```

```

% Buat Koneksi Database
dbConn = database(dbName, user , password, jdbcDriver,
jdbcString);
usrnme=get(project.edit13,'String');
passwd=get(project.edit16,'String');
% exdata = {kd, usrn,passwd};
% colnames = {'kd_user','username','password'};
% fastinsert(dbConn, 'user', colnames, exdata);
exec(dbConn,'select * from user where username='+usrnme+' AND
password='+passwd+')






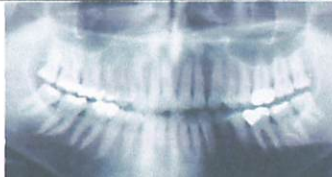

% Insert data into Table1.
% exec(dbConn,'INSERT INTO user(kd_user,username,password)
VALUES("jhjh", "hgjhgsrnm","passwd")');
close(dbConn);
function edit13_Callback(hObject, eventdata, handles)
% hObject handle to edit13 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit13 as text
% str2double(get(hObject,'String')) returns contents of
edit13 as a double
% --- Executes during object creation, after setting all
properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit13 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton16 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
function edit14_Callback(hObject, eventdata, handles)
% hObject handle to edit16 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)



% Hints: get(hObject,'String') returns contents of edit16 as text
% str2double(get(hObject,'String')) returns contents of
edit16 as a double

```



### Nama Citra dan Gambar

NO	Nama Citra	Citra
1	Gambar1	
2	Gambar2	
3	Gambar3	
4	Gambar4	
5	Gambar5	
6	Gambar6	
7	Gambar7	

8	Gambar8	
9	Gambar9	
10	Gambar10	