

**SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG
PADA PCB MENGGUNAKAN SENSOR KAMERA**

SKRIPSI

INSTITUT TEKNOLOGI NASIONAL



Disusun Oleh :

SYAMSUL DESKA WIJAYA

NIM : 11.12.524

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2015**

AMERICAN AIR FORCE
OFFICE OF THE SECRETARY

SECRET

SECRET

AMERICAN AIR FORCE
OFFICE OF THE SECRETARY

SECRET
OFFICE OF THE SECRETARY
AMERICAN AIR FORCE
OFFICE OF THE SECRETARY
SECRET

LEMBAR PERSETUJUAN
SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG
PADA PCB MENGGUNAKAN SENSOR KAMERA

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Persyaratan Guna
Untuk Memperoleh Gelar Sarjana Teknik Strata Satu (S-1)*

Disusun Oleh :

Syamsul Deska Wijaya
11.12.524

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP.Y.1030800417

Dr. Eng. I Komang Somawirata, ST, MT
NIP.P.1030100361

Mengetahui,
Ketua Jurusan Teknik Elektro S-1



M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2015

SURAT PERNYATAAN ORISINALITAS

Yang bertandatangan di bawah ini:

Nama : Syamsul Deska Wijaya
NIM : 1112524
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 14 September 2015

Yang membuat Pernyataan,



Syamsul Deska Wijaya
NIM : 1112524



SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG PADA PCB MENGGUNAKAN SENSOR KAMERA

Syamsul Deska Wijaya, NIM 11.12.524

Dosen Pembimbing: Dr. Eng. Aryuanto Soetedjo, ST, MT dan
Dr. Eng. I Komang Soma Wirata, ST, MT

Konsentrasi Teknik Elektronika, Jurusan Teknik Komputer
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
Email: syambosta@yahoo.com

ABSTRAK

PCB (Printed Circuit Board) merupakan alat wajib jika ingin membangun suatu rangkaian elektronika yang baik. Pengeboran PCB otomatis merupakan suatu sistem pengeboran yang memungkinkan dilakukan pengeboran pada suatu PCB secara otomatis. Dengan *Image Processing* dapat mengimplementasikan pengeboran otomatis ke dalam sebuah Aplikasi DELPHI untuk menentukan titik koordinat lubang PCB. Maka diperlukan teori-teori dasar *Image Processing* yang dapat menunjang agar sistem bor otomatis pada PCB khususnya pengenalan titik titik bor PCB dapat dikenali seperti *Gray scale*, *Invers* (negasi), dan *Thresholding*. Dari teori tersebut nanti akan dideteksi titik titik PCB yang telah ditandai sebagai titik drill dan akan disimulasikan hasil pendeteksian titik drill secara satu persatu sebagai posisi drill yang dilakukan oleh mesin.

Keyword : *PCB, Drill, Image Processing*

KATA PENGANTAR

Puji Syukur Kehadirat Allah SWT atas berkat dan rahmat-Nya, sehingga kami selaku penyusun dapat menyelesaikan Laporan Skripsi ini yang berjudul “Sistem Pendeteksi Untuk Menentukan Lubang Pada PCB Menggunakan Sensor Kamera”.

Sebagai pihak penyusun penulis menyadari tanpa adanya kemauan dan usaha serta bantuan dari berbagai pihak, maka laporan ini tidak dapat diselesaikan dengan baik. Oleh sebab itu, penyusun mengucapkan terima kasih kepada yang terhormat :

1. Dr. Ir. Lalu Mulyadi, MTA selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. M. Ibrahim Ashari, ST, MT selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
4. Dr. Eng. Aryuanto Soetedjo, ST, MT selaku Dosen Pembimbing satu Tugas Skripsi.
5. Dr. Eng. I Komang Soma Wirata, ST, MT selaku Dosen Pembimbing dua Tugas Skripsi.
6. Sahabat-sahabat dan rekan-rekan yang tidak dapat saya sebutkan satu-persatu, kami ucapkan banyak terima kasih atas bantuannya dalam proses pembuatan Skripsi yang telah saya kerjakan, begitu juga dengan penyelesaian laporan ini.

Penyusun menyadari bahwa pembuatan skripsi ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca sangat penulis harapkan demi kesempurnaan dalam pembuatan skripsi ini.

Malang, September 2015

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix

BAB I PENDAHULUAN

1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	3
1.4. Batasan Masalah	3
1.5. Metodologi.....	3
1.6. Hambatan – Hambatan.....	4
1.7. Sistem Penulisan	4

BAB II LANDASAN TEORI

2.1. Image Prossesing	6
2.1.1. Image Prossesing dan Kegunaan	6
2.1.2. Grayscale	7
2.1.3. Invert.....	8
2.1.4. Thresholding	9
2.1.4.1. Singel Threshold.....	10
2.1.4.2. Dual Threshold	10
2.1.5. Representasi Objek	10
2.1.6. Teknik Segmentasi Dalam Citra	12
2.1.7. Ekstrasi Objek.....	13
2.2. Sesnor Kamera	13
2.3. Aplikasi Delphi	14
2.4. Microsoft Access	15

2.5	Format Bitmap	16
-----	---------------------	----

BAB III PERANCANGAN DAN PEMBUATAN APLIKASI

3.1.	Sampel Pengujian	18
3.1.1.	Studi pustaka.....	19
3.1.2.	Perancangan Sistem	19
3.1.3.	Pembentukan Foto PCB.....	20
3.1.4.	Data Uji.....	20
3.1.5.	Pengembangan Sistem Pembacaan PCB	20
3.1.6.	Sistem Pembacaan Titik.....	20
3.1.7.	Pengujian Sistem.....	20
3.1.8.	Hasil Uji.....	20
3.1.9.	Analisa Hasil.....	20
3.1.10.	Hasil	21
3.2.	Flowcart Sistem	21
3.2.1.	Flowcart Capture Webcam	22
3.2.2.	Flowcart Deteksi Titik	22
3.2.3.	Flowcart Penentuan Koordinat	24
3.3.	Tampilan Software.....	25
3.3.1.	Tampilan Utama	25
3.3.2.	Tampilan Capture Gambar.....	27
3.3.3.	Tampilan Zoom	27
3.3.4.	Tampilan Manual	28
3.3.5.	Tampilan Menu proses.....	29
3.4	Manual Proses Sistem.....	30
3.4.1	Manual Proses Sistem Penentuan Titik Lubang PCB.....	30
3.4.2	Manual Proses Sistem Penentuan Koordinat Titik	35

BAB IV PENGUJIAN DAN PEMBAHASAN

4.1.	Pengujian Deteksi	36
4.2.	Pengujian letak koordinat hasil deteksi titik PCB	38
4.3.	Analisa dan Pengujian	40

BAB V PENUTUP

5.1. Kesimpulan	47
5.2. Saran	47

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1.	Image Prossesin.....	7
Gambar 2.2.	Grayscale.....	8
Gambar 2.3.	Invers.....	9
Gambar 2.4.	Thereshold.....	10
Gambar 2.5.	Macam Representasi Objek.....	11
Gambar 2.6.	Kamera Wecam.....	14
Gambar 2.7	Aplikasi Delphi.....	15
Gambar 2.8	Microsoft Access.....	16
Gambar 3.1.	Sekema Metodologi Penelitian.....	18
Gambar 3.2.	Diagram Blog Sistem.....	19
Gambar 3.3.	Flowcart Utama.....	21
Gambar 3.4.	Flowcart Capture.....	22
Gambar 3.5.	Flowcart Deteksi Titik.....	23
Gambar 3.6.	Flowcart Penentuan Koordinat.....	24
Gambar 3.7.	Tampilan Utama Aplikasi.....	26
Gambar 3.8.	Tampilan Capture.....	27
Gambar 3.9.	Tampilan Zoom.....	28
Gambar 3.10.	Tampilan manual Proses.....	29
Gambar 3.11.	Tampilan Menu Proses.....	29
Gambar 3.12.	Pengujian Titik.....	30
Gambar 3.13.	Pengujian Titik X.....	31
Gambar 3.14.	Pengujian Titik Y.....	32
Gambar 3.15.	Pengujian Titik Image.....	33
Gambar 3.16.	Pengujian Image.....	34
Gambar 3.17.	Penentuan Koordinat.....	35
Gambar 4.1.	Capture Proses Pengambilan Titik.....	37
Gambar 4.2.	Prosses Mendapatkan Titik.....	37
Gambar 4.3.	Prosses SimulasiMendapatkan Titik.....	38
Gambar 4.4.	Prosses Mendapatkan Titik.....	38
Gambar 4.5.	Prosses Simulasi Pendeteksi Titik Keseluruhan.....	39

Gambar 4.6. Proses Simulasi Pendeteksi Titik.....	39
Gambar 4.7. Proses Muncul Tampilan Selesai.....	40
Gambar 4.8. Pengujian Blueprint Koordinat a.....	41
Gambar 4.9. Pengujian Blueprint Penentuan Drill a.....	41
Gambar 4.10. Pengujian Blueprint Koordinat b.....	42
Gambar 4.11. Pengujian Blueprint Penentuan Drill b	42
Gambar 4.12. Pengujian Menggunakan Capture a.....	43
Gambar 4.13. Pengujian Menggunakan Capture b	44
Gambar 4.14. Pengujian Menggunakan Manual Capture a	45
Gambar 4.15. Pengujian Menggunakan Manual Capture b	45

DAFTAR TABEL

Tabel 2.1. Hubungan Antara Bit Pada Bitmap.....	17
Tabel 4.1. Hasil Pengujian Blueprint	43
Tabel 4.2. Hasil Pengujian Capture	44
Tabel 4.3. Hasil Pengujian Manual	46
Tabel 4.4. Hasil Pengujian keseluruhan	46

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Pada zaman yang modern ini banyak alat elektronik mulai dari yang manual hingga yang serba otomatis penggunaannya marak berkembang di Indonesia, bukan hanya alat alat berat, alat alat rumah tangga pun di rancang untuk semakin canggih penggunaannya. Alat alat tersebut memang sengaja dirancang untuk memenuhi kebutuhan manusia yang semakin berkembang dan kompleks. Hingga sistem dan alat yang dikembangkan dan digunakan mengandung sistem yang kompleks pula. Dalam sistem dari alat atau mesin elektronik yang banyak mengandung unsur dan penggunaan PCB (Printed Circuit Board).

Thiang, Sherwin R.U. Sompie, 2002. Menuliskan tentang Mesin Bor Otomatis dengan Menggunakan Kamera untuk Mendeteksi Koordinat Bor. PCB (Printed Circuit Board) merupakan salah satu komponen elektronika yang berfungsi sebagai tempat atau wadah dimana untuk menaruh komponen elektronika yang berada diatasnya, papan PCB juga memiliki jalur konduktor untuk menghubungkan satu komponen ke komponen lainnya. Dalam PCB (Printed Circuit Board) mengandung lubang lubang agar komponen-komponennya mempunyai tempat tersendiri dan tidak berantakan.

Agar PCB (Printed Circuit Board) dapat dimasuki komponen komponen elektronika dan digunakan perlu dilakukan pengeboran PCB. Dalam pengeboran lubang titik dari PCB sekarang ini ada 2 cara yaitu otomatis dan manual. Dimana yang manual dalam mengebor titik titik PCB dilakukan menggunakan media alat yang dapat mengebor titik satu persatu dalam waktu yang lama. Sedangkan pengeboran titik PCB secara otomatis merupakan suatu sistem pengeboran yang yang terbaru dan memungkinkan dilakukan pengeboran pada suatu PCB secara cepat dan tidak memakan waktu lama^[13].

Thiang, Sherwin R.U. Sompie, 2002. Menuliskan tentang Mesin Bor Otomatis dengan Menggunakan Kamera untuk Mendeteksi Koordinat Bor. Pengeboran PCB otomatis ini menggunakan suatu teknologi komputasi yang merekam titik koordinat dari tiap titik dan hanya perlu mengulang perintah dari

sistem kepada mesin Drill PCB untuk melakukan pengeboran terhadap PCB. Hal itu membuat proses pengeboran PCB menjadi lebih cepat dan tidak memakan waktu yang lama.

Dari uraian diatas membuat penulis ingin membuat suatu sistem pengeboran PCB secara otomatis dengan menggunakan media kamera. Pengeboran PCB otomatis ini menggunakan kamera sebagai sensor yang dapat mendeteksi koordinat titik lubang PCB dan pada software sebagai pengelolah citra digital[2]. Sistem ini nanti dalam melakukan proses pengeboran terhadap titik titik PCB dengan lebih beragam dan sistem tidak merekam titik koordinat dari satu layout PCB namun menyimpan banyak layout PCB^[14].

Hal itu dapat dilakukan karena sebelum dilakuakn proses pengeboran, sistem melakukan proses *capture* kamera dari objek PCB yang akan di Drill. Sensor yang dimaksud disini adalah mendeteksi suatu bentuk, warna maupun posisi dari suatu objek yang didapat dari kamera. Dimana hasil capture dari layout PCB akan dideteksi koordinat–koordinat dan disimpan dalam database. Data base sendiri berfungsi sebagai penyimpan data (titik koordinat). Kemudian hasil deteksi akan akan disimpan didalam database untuk dijadikan acuan dari mesin drill dalam menentukan koordinat titik yang akan dilubangi.

Terakhir penulis ingin menjadikan sistem sebagai alat untuk menyelesaikan tugas akhir dalam rangka syarat untuk mendapatkan gelar sarjana teknik computer (ST) dan member judul penelitian tugas akhir ini dengan judul “SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG PADA PCB MENGGUNAKAN SENSOR KAMERA”. Diharapkan dari alat ini nanti dapat membuka pemikiran tentang sistem yang bekerja untuk melakukan proses drill PCB dari berbagai cara dan pengolahan hingga kemajuan sistem pengeboran dapat dikembangkan kearah yang lebih canggih lagi dari yang sekarang ada.

1.2 RUMUSAN MASALAH

Mengacu pada permasalahan diatas yang diuraikan pada skripsi ini, maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara sensor kamera mendapatkan gambar untuk diteruskan ke software ?

2. Bagaimana Aplikasi Delphi menampilkan hasil image dari sensor camera ?
3. Bagaimana Aplikasi Delphi memproses image prosesing untuk mendapatkan koordinat pengeboran PCB ?
4. Bagaimana Aplikasi Dhelpi dapat menentukan jumlah dari lubang yang akan di bor pada PCB?

1.3 TUJUAN PENELITIAN

Tujuan dari Skripsi ini adalah mempelajari teori-teori Image Processing kemudian mengimplementasikan ke dalam sebuah Aplikasi DELPHI untuk mempermudah menentukan titik kordinat lubang PCB.

1.4 BATASAN MASALAH

Agar tidak terjadi penyimpangan maksud dan tujuan utama penyusunan skripsi ini maka batasan masalah pada skripsi ini antara lain:

1. Hanya membahas tentang pencitraan gambar, *Image Processing* dan komunikasi antara sensor kamera dan aplikasi.
2. Tidak membahas mekanik Mesin Bor PCB.
3. Pengambilan data image hanya berasal dari pemindaian PCB yg sudah ada titik dan belum dilubangi.

1.5 METODE YANG DIGUNAKAN

Ada beberapa metode yang digunakan untuk proyek tugas akhir ini, yaitu :

1. Studi Literatur

Mencari referensi-referensi yang tentang studi mengenai bentuk-bentuk gambar PCB, pengolahan *Image Processing*, untuk diprosess dari sensor kamera ke Aplikasi Delphi kemudian mengambil koordinat pada setiap titik lobang PCB.

2. Pengumpulan Data

Data-data yang akan digunakan adalah berbagai macam PCB, jumlah titik lobang PCB, dari yang memiliki jumlah yang sedikit sampai yang banyak.

3. Pembuatan Program

Bahasa pemrograman yang akan digunakan adalah Borland Delphi versi 7.0. Program yang akan dibuat berupa program pemrosesan suatu objek yaitu gambar PCB yang hasil proses pemrosesan *image* akan diketahui setiap titik lobang pada PCB

4. Pengujian

aplikasi akan diuji dengan melakukan pengeboran pada PCB yang sudah ada titik lobang namun belum di bor dalam bentuk visual.

5. Kesimpulan

Penarikan kesimpulan akan dilakukan dengan melihat apakah pengeboran yang dilakukan telah tepat pada tiap-tiap titik koordinat yang sudah ditentukan.

1.6 HAMBATAN-HAMBATAN

Pada perencanaan dan pembuatan tugas akhir ini tidak terlepas dari beberapa hambatan antara lain:

1. Pembuatan scrib yang relatif sulit
2. Harga kamera yang relatif tinggi
3. Susahnya mencari PCB pabrikan yang belum di lubang

1.7 SISTEMATIKA PENULISAN

Untuk mempermudah pemahaman naskah tugas akhir, maka penulisannya disusun sebagai berikut:

Bab I : Pendahuluan

Dalam bab ini dibahas tentang : Latar belakang, ruang lingkup, tujuan, metodologi penelitian, hambatan-hambatan dan sistematika pembahasan.

Bab II : Teori Penunjang

Dalam bab ini akan dibahas tentang : Teori *Image Processing*, Sensor Kamera secara singkat.

Bab III : Perencanaan Aplikasi

Dalam bab ini akan dibahas tentang : Perencanaan sistem secara keseluruhan yang menjelaskan bagaimana sistem bekerja, Perencanaan software meliputi teknik pemrosesan gambar kemudian pendeteksian koordinat titik pada PCB.

Bab IV : Pengujian Aplikasi

Bab ini akan membahas menentukan koordinat lubang PCB, banyaknya lubang pada PCB, Warna koordinat PCB, dan beberapa sampel PCB yang akan di pindai untuk di tentukan kesalahan menentukan koordinat pada aplikasi.

Bab V : Kesimpulan dan Saran

Bab ini berisi kesimpulan-kesimpulan dari hasil uji coba.

DAFTAR PUSTAKA**LAMPIRAN – LAMPIRAN**

BAB II

LANDASAN TEORI

Untuk memudahkan dalam memahami sistem ini, maka diperlukan teori-teori dasar yang menunjang dan dapat menjelaskan tentang karakteristik aplikasi yang digunakan maupun masalah yang dibahas, sehingga dapat diperkirakan prinsip dan cara kerja secara umum dari sistem ini. Selain itu dengan dasar teori yang ada dapat menambah pemahaman yang mendukung dalam perancangan dan pembuatan aplikasi ini.

2.1 IMAGE PROCESSING

2.1.1 Image Processing dan Kegunaannya

Image processing adalah memproses suatu gambar sehingga menghasilkan gambar lain yang lebih sesuai dengan keinginan kita. *Image processing* sendiri suatu bentuk pengolahan atau pemrosesan sinyal dengan input berupa gambar (image) dan ditransformasikan menjadi gambar lain sebagai keluarannya dengan teknik tertentu. Image processing dilakukan untuk memperbaiki kesalahan data sinyal gambar yang terjadi akibat transmisi dan selama akuisisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan manusia baik dengan melakukan manipulasi dan juga penganalisisan terhadap gambar^[1]. Pemakaian image processing dalam dunia sehari-hari diantaranya ialah sebagai berikut:

- 1. Image Processing dalam Fotografi dan Perfilman**

Pada dunia fotografi dan perfilman image processing memegang peranan yang sangat penting, antara lain pemrosesan foto agar hasil yang diinginkan lebih baik maupun gambar-gambar animasi pada film

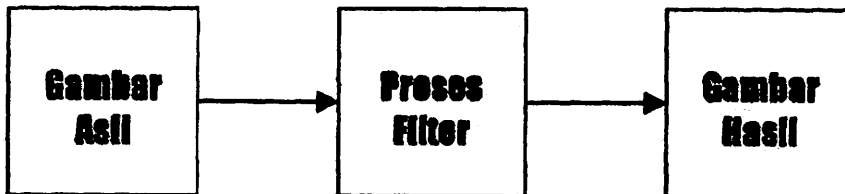
- 2. Image Processing dalam Kedokteran**

Dalam dunia kedokteran, image processing digunakan terutama untuk memperjelas hasil x-ray (ctscan) organ tubuh manusia. Gambar yang didapat dari x-ray umumnya kabur sehingga sulit bagi para dokter untuk menganalisis kelainan-kelainan yang terdapat pada organ tubuh tersebut. Dengan image processing gambar tersebut dapat diperjelas.

3. Image Processing dalam dunia game

Dalam pembuatan game image processing digunakan untuk menciptakan efek-efek seperti bayangan diatas permukaan air, tampilan yang kabur karena terkena angin, transparansi, pencahayaan dan lain-lainnya.

Image processing dapat digambarkan dengan diagram berikut ini :



Gambar 2.1 Image Processing pada gambar^[1]

Filter adalah jenis Image processing yang dilakukan pada gambar asli.

2.1.2 Grayscale

Gray scale bukanlah suatu filter pada Image processing. Dia merupakan suatu format gambar pada sistim gambar digital (image). Pada suatu image format RGB (Red, Green, Blue) mewakili gambar pada sistem digital untuk setiap titiknya, dimana masing - masing komponen warna diwakili dengan satu byte, jadi variasi yang ada pada masing - masing komponen R, G dan B adalah variasi 0 sampai 255. Total variasi yang dihasilkan untuk sistem warna digital ini adalah $256 * 256 * 256$ atau 16.777216 jenis warna karena setiap komponen warna diwakili dengan satu byte. Kalkulasi pemrosesan gambar dengan sistem RGB ini sangatlah memboroskan memory dan juga waktu. Untuk itu diperlukan reduksi warna, dalam pemrosesan gambar, terutama pengenalan objek, sistem RGB sendiri tidaklah memberikan respon yang baik, sehingga digunakan sistem format gray scale atau gray level, dimana format gambar abu - abu. Sistem gambar gray scale memerlukan satu byte (delapan bit) untuk penyimpanan data, mempunyai kemungkinan warna dan 0 (hitam) sampai 255 (putih), dalam hal ini nilai RGB untuk sistim gambar gray scale adalah sama^[2]. Secara sederhana rumusan fungsinya didefinisikan sebagai berikut :

$$Gray(i, j) = \left(\frac{f^{RED}(i, j) + f^{GREEN}(i, j) + f^{BLUE}(i, j)}{3} \right)$$

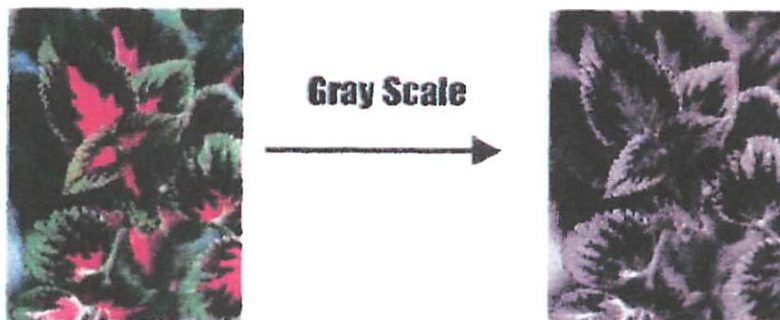
Dimana :

Gray(I,j) = Pixel Grayscale ke I,j

I = Baris Image

J = Kolom Pixel

F^{RedGreenBlue} = Nilai Pixel R,G,B



Gambar 2.2 Contoh Konversi ke Sistim Gray Scale^[2]

2.1.3 Invers

Invers (negasi), operasi yang digunakan untuk mendapatkan citra negatif, seperti film (negatif) dalam dunia cetak foto. Sehingga untuk mencari nilai negatifnya harus dikurangkan dengan nilaimaksimumnya, baik true color dan gray level^[3].

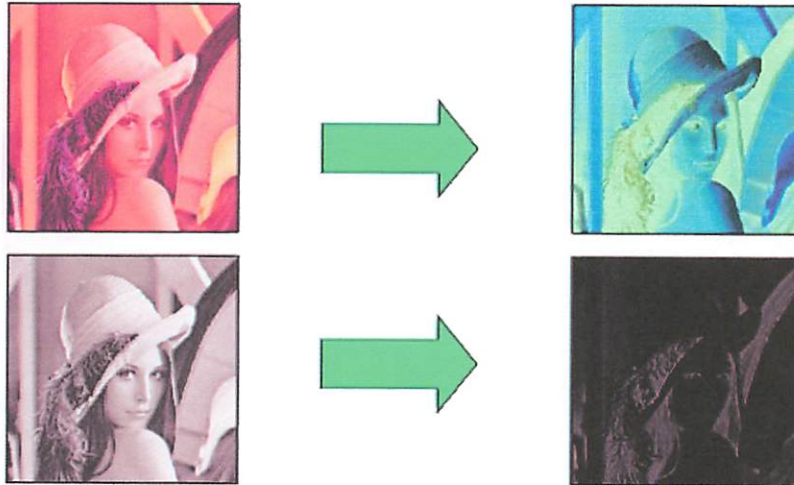
Rumusan fungsi kontras dapat didefinisikan sebagai berikut :

$$f(x, y) = f_{Maksimum} - f(x, y)$$

Dimana

f_{Maksimum}: bilacitra true color=255, bilagray level=128

Contoh hasil transformasi citra grayscale dapat dilihat pada gambar dibawah ini:



Gambar 2.3 Contoh Konversi ke Sistem invers^[3]

2.1.4 Thresholding

Thresholding bertujuan untuk mengurangi warna yang dipakai oleh suatu image, menggeser" posisi pita-pita intensitas pada image histogram ke tempat lain yang diinginkan dan mengubah resolusi intensitas warna dari gambar. Threshold akan bekerja dengan sempurna pada image yang telah di grayscale terlebih dahulu, meskipun tidak menutup kemungkinan dilakukan threshold pada image yang bukan gray scale. Dasar dari threshold ini dilakukan dengan cara mengelompokkan warna-warna yang memiliki intensitas yang hampir sama sehingga jumlah warna yang diperlukan oleh gambar tersebut menjadi berkurang. pada Thresholding operasi pemetaan suatu nilai piksel berdasarkan syarat nilai ambang menjadi nilai baru yang dikehendaki

$$f_0(x, y) = \begin{cases} 0, & f_1(x, y) < 128 \\ 255, & f_1(x, y) \geq 128 \end{cases}$$

Dimana

$f_0(x, y)$: adalah citra hasil threshold

Piksel-piksel yang nilainya intensitasnya dibawah 128 diubah menjadi hitam(nilai intensitas 0), sedangkan piksel-piksel yang nilai intensitany diatas 128 diubah menjadi warna putih(nilai intensitas = 255)^[4].



Figure 1. Comparison of the two images.

The Problem

The problem of finding the best way to represent a set of data is a common one. In this case, the data is represented by two images. The first image is a low-resolution, high-contrast representation of the data. The second image is a high-resolution, low-contrast representation of the data. The goal is to find a way to represent the data that is both compact and informative.

One way to do this is to use a technique called "principal component analysis" (PCA). This technique finds the directions in which the data varies the most, and then uses only the most important directions to represent the data. This can result in a much more compact representation of the data, while still retaining most of the information.

Proses threshold ada dua macam yaitu:

2.1.4.1 Single Threshold

Single threshold adalah threshold yang akan mengurangi suatu warna berdasarkan satu variabel. Variabel disini adalah nilai R,G,B dari suatu image. Nilai variabel akan menentukan batas warna yang akan diubah. yaitu nilai diatas variabel dan dibawahnya. Nilai diatasnya maupun dibawahnya akan berubah sesuai dengan wama yang diinginkan.

2.1.4.2 Dual Threshold

Dual threshold memiliki dua buah variabel, yaitu nilai minimum dan maximum. Wama akan berubah jika nilai variabel ada antara nilai minimum dan maximum. Diluar nilai itu juga akan berubah sesuai dengan wama yang diinginkan.



Gambar 2.4 Contoh Thresholding pada image^[4]

2.1.5 Representasi Objek

Untuk kebutuhan pencarian, objek dapat direpresentasikan berdasarkan bentuk dan penampilan luarnya^[5]. Berdasarkan bentuknya, objek dapat direpresentasikan sebagai berikut :

1. Titik, baik satu titik pada pusat objek maupun beberapa titik pada beberapa posisi objek. Representasi titik ini sesuai untuk objek yang hanya menempati sebagian kecil saja dari area citra yang diamati, atau dengan kata lain objek tersebut cukup kecil ukurannya (Gambar (a) dan (b)).
2. Bentuk geometrik yang sederhana, seperti segi empat, elips, lingkaran dan lain sebagainya. Biasanya representasi ini digunakan untuk objek yang bentuknya tidak mudah berubah (*rigid*) dan sederhana, meskipun

... dan ...

2.1.1.1. ...

... dan ...

2.1.1.2. ...

... dan ...

2.1.2. ...

2.1.2.1. ...

... dan ...

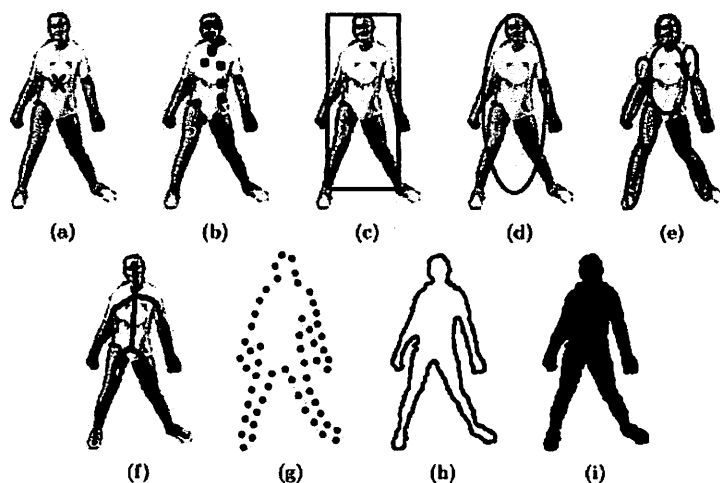
... dan ...

(b)

... dan ...

demikian, pada beberapa kasus representasi ini juga digunakan untuk objek yang bentuknya cukup kompleks (Gambar (c) dan (d)).

3. Model artikulasi bentuk, objek-objek yang merupakan artikulasi bentuk tersusun dari beberapa bagian yang disatukan oleh sebuah penghubung. Sebagai contoh, badan manusia merupakan objek artikulasi bentuk dengan lengan, tangan, kepala, dan kaki yang dihubungkan oleh sendi-sendi. Hubungan antar bagian dibangun oleh model pergerakan kinematik. Untuk memodelkan objek yang merupakan artikulasi bentuk, setiap bagian yang berhubungan dapat direpresentasikan dengan bentuk silinder atau elips (Gambar (e)).
4. Model kerangka, kerangka dari sebuah objek dapat dibentuk dengan menerapkan transformasi *medial axis* pada siluet dari objek. Representasi ini biasa digunakan untuk tujuan pengenalan objek (Gambar (f)).
5. Kontur dan siluet, kontur merupakan garis yang membatasi daerah dari sebuah objek dan memisahkannya dari objek-objek lainnya sedangkan siluet merupakan daerah yang ada didalam kontur tersebut. Representasi ini sesuai untuk objek yang bentuknya kompleks dan mudah berubah (Gambar (h) dan (i)).



Gambar 2.5 Berbagai macam representasi objek. (a) Satu titik pada pusat, (b)Beberapa titik, (c) Segi empat, (d) Elips, (e) Artikulasi bentuk,(f) Kerangka, (g) Titik-titik pengendali pada kontur, (h) Kontur lengkap, (i) Siluet^[5]

2.1.6 Teknik Segmentasi dalam Citra Digital

Segmentasi ditujukan untuk memisahkan objek dari latar belakangnya, hasil akhir dari teknik ini adalah citra yang didalamnya terdapat partisi-partisi sebagai penanda bagi objek yang dicari. Penanda tersebut biasanya berupa warna-warna yang kontras antara objek dengan latar belakangnya. Pada citra biner, objek ditandai dengan warna putih sedangkan latar belakangnya ditandai dengan warna hitam (ini bisa berlaku sebaliknya).

Salah satu pendekatan untuk mempartisi citra kedalam beberapa daerah adalah segmentasi berdasarkan daerah itu sendiri yang biasanya mempunyai nilai intensitas yang berbeda dengan latar belakang.

Pada segmentasi berdasarkan daerah, semua piksel yang berkorespondensi dengan sebuah objek dikelompokkan dan diberi tanda yang sama agar mereka dikenali sebagai satu daerah.

Piksel-piksel ini dikelompokkan ke dalam daerah menggunakan kriteria yang membedakan mereka dari bagian citra lainnya. Dua kriteria yang penting dalam segmentasi adalah kemiripan nilai intensitas dan kedekatan posisi.

Kemiripan nilai intensitas lebih banyak digunakan karena memang lebih dominan, sedangkan kedekatan posisi digunakan bila sebagian dari objek dan latar belakang mempunyai kemiripan nilai sehingga perlu diperiksa dengan syarat yang kedua.

Kemiripan nilai intensitas dua piksel dapat diukur berdasarkan perbedaan nilai intensitas keduanya, sedangkan kedekatan posisi dua piksel dapat diukur melalui jaraknya (biasanya menggunakan jarak *euclidean*).

Dalam situasi yang sederhana dimana citra hanya mengandung beberapa objek, segmentasi dapat dilakukan dengan binerisasi melalui proses *thresholding* dan citra yang lebih kompleks diperlukan teknik yang lebih canggih untuk menandai piksel milik suatu daerah yang berkorespondensi dengan bagian objek. Segmentasi juga dapat dilakukan dengan menemukan piksel-piksel yang terletak pada perbatasan daerah. Piksel-piksel ini disebut juga dengan piksel tepi, dapat ditemukan dengan memeriksa piksel-piksel tetangganya^[6].

2.1.7 Ekstraksi Objek

Setelah melalui proses segmentasi, objek yang menjadi pusat perhatian dapat dengan mudah dipisahkan dari latar belakangnya karena telah memiliki tanda yang spesifik. Proses ekstraksi objek adalah menemukan tanda dari objek tersebut untuk kemudian diproses sesuai dengan kebutuhan, seperti misalnya dilakukan pengukuran maupun modifikasi terhadapnya^[7].

2.2 Sensor Kamera

Sensor kamera adalah sensor penangkap gambar yang dikenal juga sebagai CCD (Charged Coupled Device) dan CMOS (Complementary Metal Oxide Semiconductor) dan yang terbaru BSI-CMOS (Back Side Illumination CMOS) yang terdiri dari jutaan piksel (MP-mega pixel) lebih. Pada tahun 2013, hampir semua kamera telah menggunakan BSI-CMOS yang lebih irit daya, tetapi dapat menghasilkan gambar yang lebih baik.

Sensor ini berbentuk chip yang terletak tepat di belakang lensa. Semakin banyak pixel yang ditangkap, semakin memungkinkan pencetakan gambar besar dengan detail gambar yang cukup (dibatasi kapasitas kertas cetak foto yang biasanya 300dpi (dot per inch), sedangkan menggunakan printer dapat dihasilkan dpi yang lebih tinggi). Untuk menghasilkan foto seukuran postcard dengan 300dpi cukup menggunakan kamera 3MP. Kualitas foto sama sekali tidak ditentukan oleh besarnya MP, tetapi oleh kualitas sensor, prosesor kamera dan yang terakhir baru lensa, berbeda dengan kamera film dimana kualitas lensa lebih menentukan^[8].

Webcam adalah kamera digital yang dikoneksikan ke komputer, digunakan untuk telekonferensi video atau tujuan lain. Webcam dapat menangkap gambar video gerak-penuh, dan beberapa model termasuk mikrofon dan kemampuan zoom^[9].



Gambar 2.6 Kamera Webcam^[9]

2.3 Aplikasi Delphi

Delphi adalah Suatu bahasa pemrograman yang menggunakan visualisasi sama seperti bahasa pemrograman *Visual Basic* (VB). Namun Delphi menggunakan bahasa yang hampir sama dengan pascal (sering disebut object pascal). Sehingga lebih mudah untuk digunakan. Bahasa pemrograman Delphi dikembangkan oleh Code Gear sebagai divisi pengembangan perangkat lunak milik *embarcadero*. Divisi tersebut awalnya milik borland, sehingga bahasa ini memiliki versi Borland Delphi.

Delphi juga menggunakan konsep yang berorientasi objek (OOP), maksudnya pemrograman dengan membantu sebuah aplikasi yang mendekati keadaan dunia yang sesungguhnya. Hal itu bisa dilakukan dengan cara mendesign objek untuk menyelesaikan masalah. OOP ini memiliki beberapa unsur yaitu : *Encapsulation* (pemodelan), *Inheritance* (Penurunan), *Polymorphism* (Polimorfisme).

Umumnya Delphi hanya digunakan untuk pengembangan aplikasi dekstop, enterprise berbasis database dan program - program kecil. Namun karena pengembangan delphi yang semakin pesat dan bersifat general purpose bahasa pemrograman ini mampu digunakan untuk berbagai jenis pengembangan software. Dan Delphi juga disebut sebagai pelopor perkembangan RadTool (*Rapid Application Development*) tahun 1995. Sehingga banyak orang yang mulai

mengenal dan menyukai bahasa pemrograman yang bersifat VCL (Visual Component Library) ini^[10].



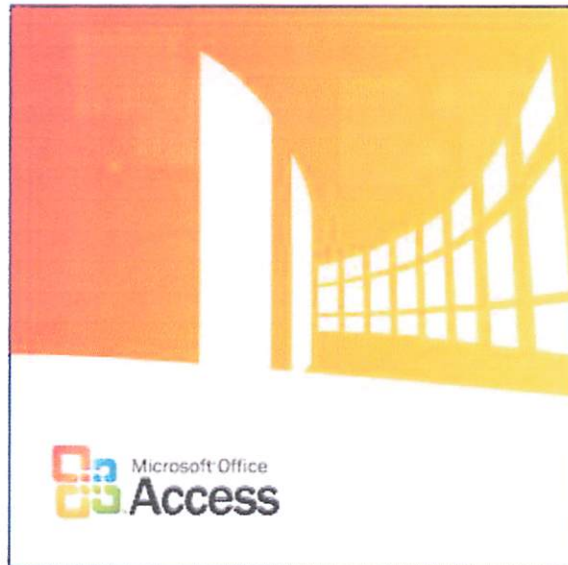
Gambar 2.7 Aplikasi Borland Delphi^[10]

2.4 Microsoft Access

Microsoft Access adalah suatu program aplikasi basis data komputer relasional yang digunakan untuk merancang, membuat dan mengolah berbagai jenis data dengan kapasitas yang besar. Pada aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna untuk membuat database.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/programmer yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik

pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek^[11].



Gambar 2.8 Aplikasi Microsoft Access^[11]

2.5 Format Bitmap

Format *bitmap*, pada citra disimpan sebagai suatu matriks di mana masing-masing elemennya digunakan untuk menyimpan informasi warna untuk setiap *pixel*. Untuk jumlah warna yang dapat disimpan ditentukan dengan satuan *bit-per-pixel*. Semakin besar ukuran *bit-per-pixel* dari suatu *bitmap*, semakin banyak pula jumlah warna yang dapat disimpan.

Format *bitmap* ini cocok digunakan untuk menyimpan citra digital yang memiliki banyak variasi dalam bentuknya maupun warnanya, seperti foto, lukisan, dan *frame* video. Format *file* yang menggunakan format *bitmap* ini antara lain adalah BMP, DIB, PCX, GIF, dan JPG. Format yang menjadi standar dalam sistem operasi Microsoft Windows adalah format *bitmap* BMP atau DIB.

Citra dalam format BMP lebih bagus dari pada citra dalam format JPEG, Terjemahan bebas dari *bitmap* adalah pemetaan bit. Artinya, nilai intensitas *pixel* di dalam citra dipetakan ke sejumlah bit tertentu. Peta bit yang umum adalah 8, artinya setiap *pixel* panjangnya 8 bit. Delapan bit ini

merepresentasikan nilai intensitas *pixel*. Dengan demikian ada sebanyak $2^8 = 256$ derajat keabuan, mulai dari 0 sampai 255.

Citra dalam format BMP ada tiga macam : citra biner, citra berwarna, dan citra hitam-putih (*grayscale*). Citra biner hanya mempunyai dua nilai keabuan, yaitu 0 dan 1. Oleh karena itu, 1 bit sudah cukup untuk merepresentasikan nilai *pixel*. Citra berwarna adalah citra yang lebih umum. Warna yang terlihat pada citra *bitmap* merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru. Setiap *pixel* disusun oleh tiga komponen warna : R (*red*), G (*green*), dan B (*blue*). Kombinasi dari tiga warna RGB tersebut menghasilkan warna yang khas untuk *pixel* yang bersangkutan.

Karakteristik lain dari *bitmap* yang juga penting adalah jumlah warna yang dapat disimpan dalam *bitmap* tersebut. Ini ditentukan oleh banyaknya bit yang digunakan untuk menyimpan setiap titik dari *bitmap* yang menggunakan satuan *bpp* (*bit per pixel*). Dalam Windows dikenal *bitmap* dengan 1, 4, 8, 16, dan 24 *bit per pixel*. Jumlah warna maksimum yang dapat disimpan dalam suatu *bitmap* adalah sebanyak 2^n , dimana n adalah banyaknya bit yang digunakan untuk menyimpan satu titik dari *bitmap*^[12].

Berikut ini tabel yang menunjukkan hubungan antara banyaknya *bit per pixel* dengan jumlah warna maksimum yang dapat disimpan dalam *bitmap*.

Tabel 2.1 Hubungan Antara *Bit Per Pixel* dengan Jumlah Warna Maksimum Pada *Bitmap*

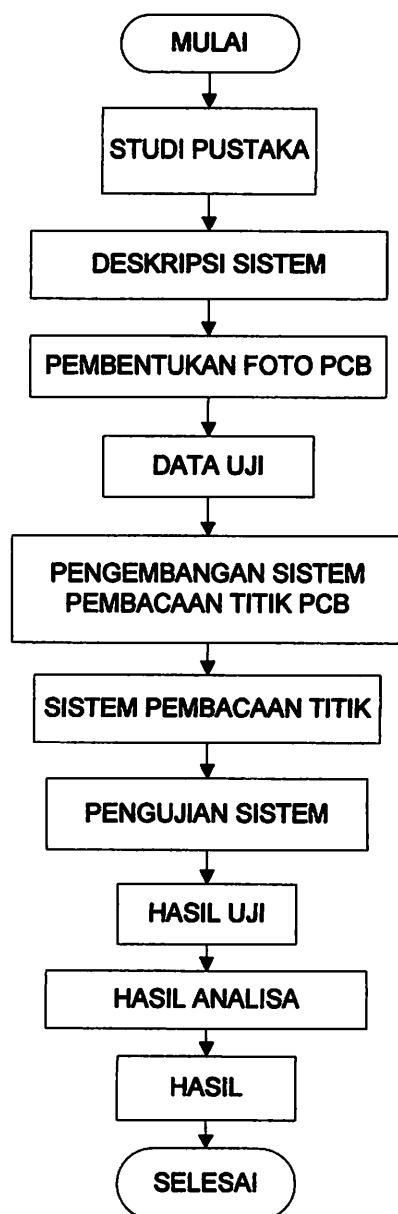
No	Jumlah bit per pixel	Jumlah warna maksimum
1	1	2
2	4	16
3	8	256
4	16	65536
5	24	16777216

BAB III

PERANCANGAN DAN PEMBUATAN APLIKASI

3.1 Tahapan Penelitian

Bab ini berisi penjelasan tentang desain sistem keseluruhan yang akan dibuat. Tahap-tahap dalam pengerjaan penelitian sesuai dengan metode penelitian yang dapat dilihat pada gambar dibawah ini:



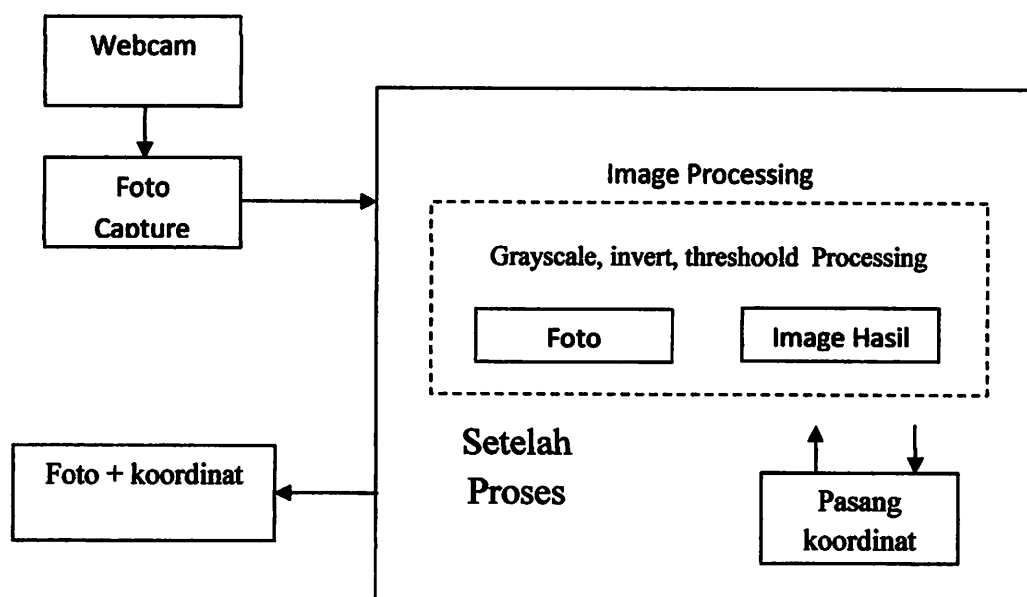
Gambar 3.1 Skema metodologi penelitian

3.1.1 Studi Pustaka

Pada tahap ini, dilakukan proses pengumpulan informasi dari buku, jurnal, dan artikel mengenai *Deteksi titik PCB* dalam penyelesaian penjadwalan perkuliahan. Mulai dari mempelajari teori tentang citra sampai pada mengimplementasi *image processing*.

3.1.2 Perancangan Sistem

Berikut ini merupakan diagram blok proses yang terjadi dalam sistem pendeteksian titik pada PCB menggunakan *image processing*.



Gambar 3.2 Diagram Blok Sistem

Pada diagram blok sistem Gambar 3.2 dapat dilihat bahwa proses pengenalan titik pada pcb dimulai ketika webcam melakukan capture PCB dan menghasilkan foto *capture*. Kemudian dilakukan proses *image processing* berupa grayscale, brightness untuk memroses foto capture menjadi image hasil.

Setelah image hasil didapat, maka sistem memasang letak koordinat tiap titik. Dan akhirnya didapatlah sebuah foto PCB + koordinat dari tiap titik yang terdeteksi.

3.1.3 Pembentukan foto PCB

Pada tahap ini, dilakukan proses pembentukan foto PCB yaitu dengan mengcapture dari PCB menggunakan webcam. Dan hasil dari proses capture akan di jadikan image BMP untuk dijadikan image uji.

3.1.4 Pembentukan data uji

Pada tahap ini, dilakukan proses pembentukan data uji berupa nilai piksel dari foto hasil proses capture yang telah di jadikan image BMP.

3.1.5 Pengembangan Sistem Pembaca titik PCB

Pada tahap ini, dilakukan proses pengembangan sistem pembaca titik PCB yang berupa rancangan dan kode program untuk dibuat sebuah program pembaca titik lubang PCB.

3.1.6 Sistem Pembaca titik

Dari pengembangan sistem pembaca titik PCB, maka dapat dibentuk sebuah aplikasi berupa sistem pembaca titik lubang PCB.

3.1.7 Pengujian Sistem

Pada tahap pengujian sistem dilakukan suatu pengujian terhadap sistem yang telah dirancang dan dibuat sebelumnya dengan menggunakan data uji berupa piksel dari foto PCB hasil proses capture webcam.

3.1.8 Hasil Uji

Pada tahap ini output dari pengujian sistem yang telah dilakukan didapatkan yaitu berupa foto hasil capture ditambah dengan nilai koordinat dari pendeteksian tiap titik lubang dari PCB.

3.1.9 Analisis Hasil

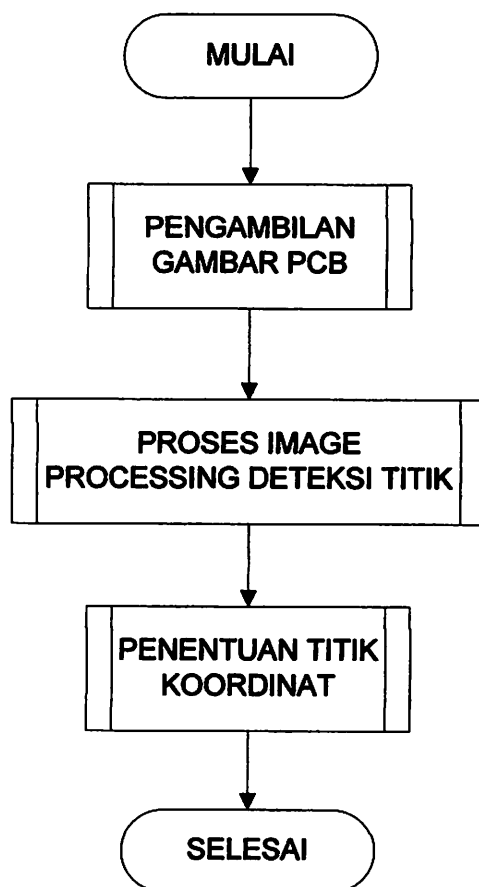
Pada tahap ini hasil pengujian, akan dianalisa berapa persen kebenaran dari sistem, yang nantinya akan dijadikan bahan tulisan.

3.1.10 Hasil

Pada tahap ini hasil analisa akan di dokumentasikan menjadi sebuah buku karya ilmiah yaitu Skripsi. Dimana dalam hasil dokumentasi ini, akan dijadikan alat dalam mendapatkan gelar sarjana.

3.2 Flowcart Sistem

Pemodelan alur kerja dari sebuah proses sistem dan sebuah urutan alur kerja dari sebuah proses akan digambarkan dari sebuah alur diagram. alur yang digambarkan mulai dari capture webcam, deteksi titik sampai dengan menentukan koordinat. Berikut ini adalah alur diagram secara keseluruhan pada sistem.

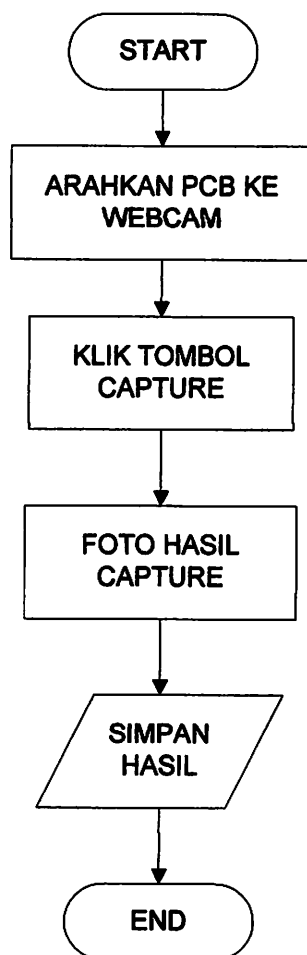


Gambar 3.3 Flowchar Utama Sistem

3.2.1 Flowcart Sistem Capture Webcam

Alur diagram capture webcam ini menjelaskan proses saat pengambilan gambar melakukan capture terhadap PCB sampai menjadi foto hasil *capture* webcam.

Alur diagram capture webcam ditunjukkan pada gambar dibawah ini :

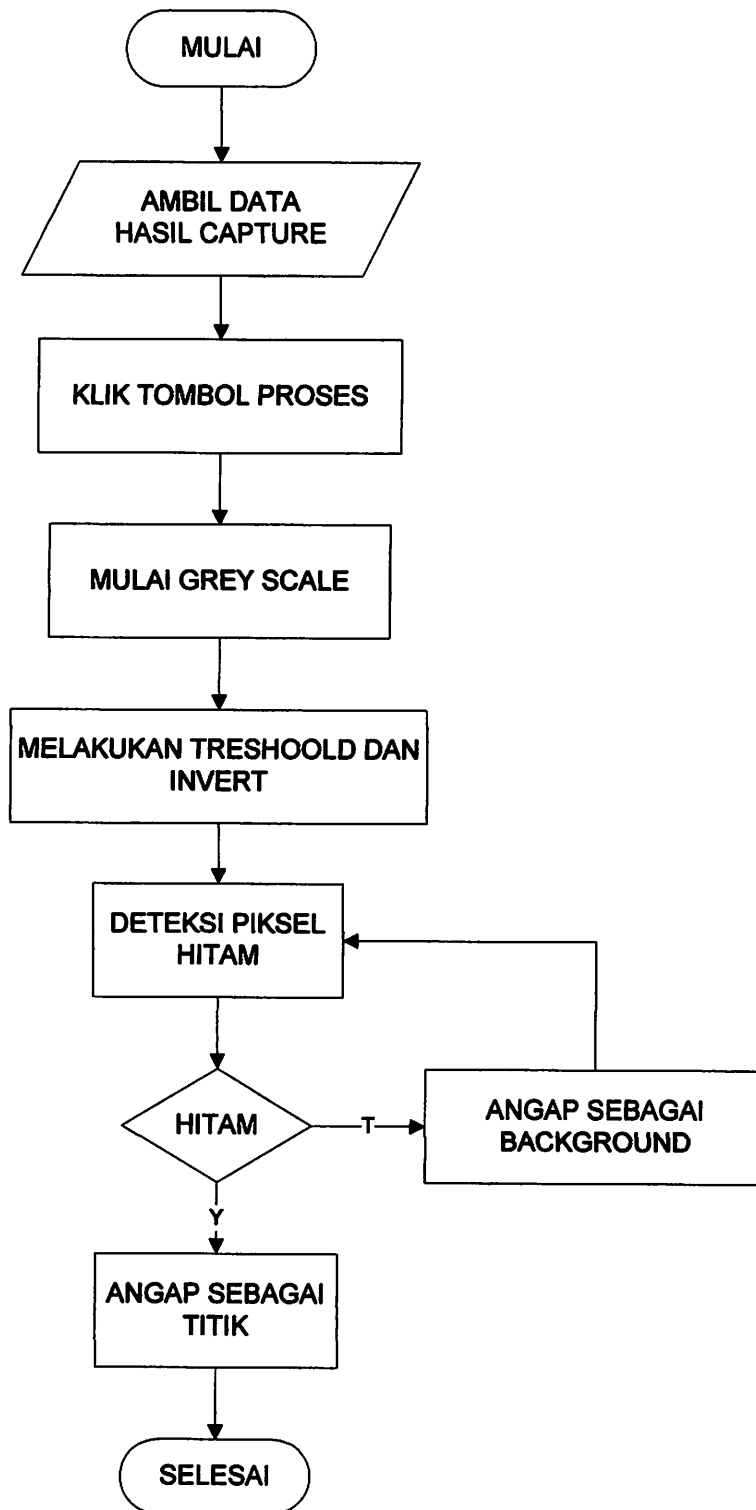


Gambar 3.4 Flowchart capture webcam

3.2.2 Flowcart Sistem Deteksi Titik

Alur diagram deteksi titik ini menjelaskan proses saat pengguna memerintah sistem untuk melakukan image processing berupa proses pendeteksian titik terhadap foto hasil capture dari PCB.

Alur diagram deteksi titik ditunjukkan pada gambar dibawah ini :

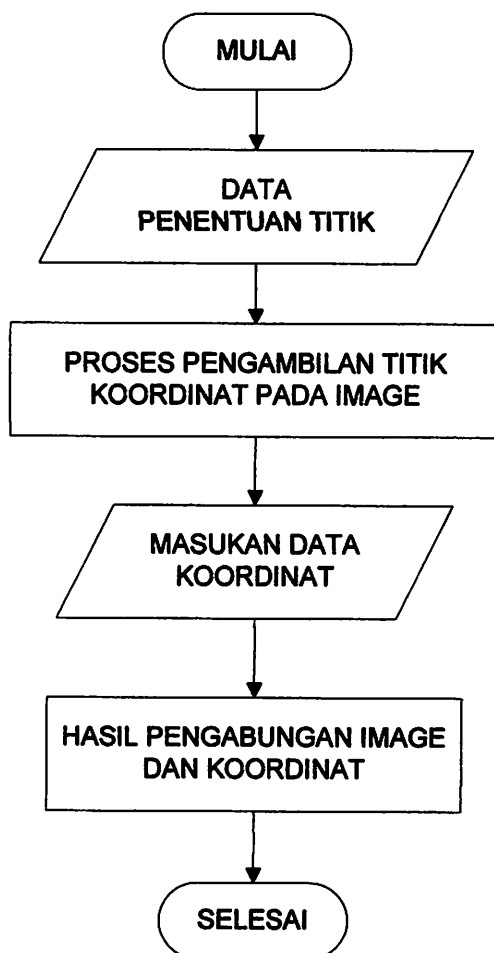


Gambar 3.5 Flowchart Deteksi Titik

3.2.3 Flowcar Sistem Penentuan Koordinat

Alur diagram penentuan koordinat menjelaskan proses saat *sistem* melakukan proses pemberian teks koordinat pada gambar hasil pengolahan deteksi titik sesuai dengan data titik yang telah didapat.

Alur diagram Penentuan koordinat ditunjukkan pada gambar dibawah ini :



Gambar 3.6 Flowchart penentuan koordinat

3.3 Tampilan Software

Graphical User Interface atau yang sering disebut sebagai GUI adalah tampilan dari program yang bisa dinikmati oleh pengguna. Perancangan tampilan *software* harus dibuat semenarik dan seindah mungkin dengan tetap mengutamakan kenyamanan dalam mengoperasikan program (*user friendly*).

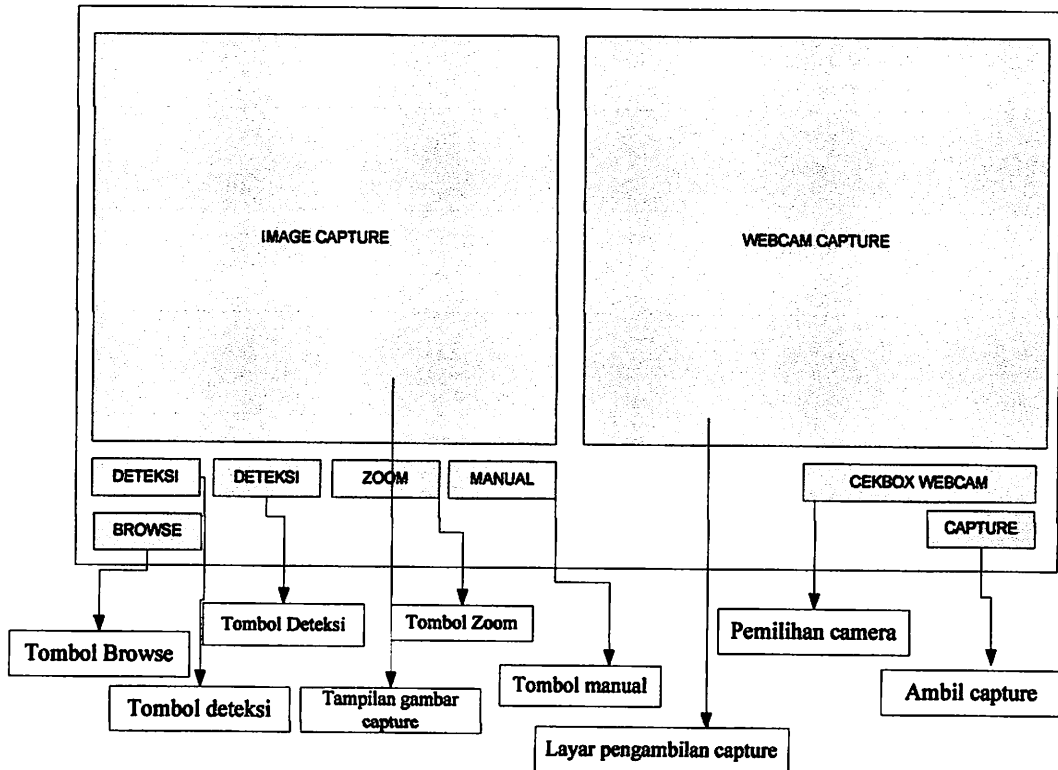
Tampilan *software* yang ada dalam aplikasi ini yang meliputi tampilan *form* aplikasi dibuat dengan menggunakan bahasa pemrograman Pascal yang terinclude dalam aplikasi editor *Delphi 7*.

Berikut ini desain tampilan dari *form* yang digunakan dalam pengembangan aplikasi pendeteksian titik PCB.

3.3.1 Perancangan Tampilan Utama

Perancangan Tampilan Menu Utama yang merupakan *form* yang berisi form menu menu yang ada dalam aplikasi.

Berikut adalah gambar dari perancangan tampilan menu utama dari aplikasi proses deteksi titik PCB.



Gambar 3.7 Tampilan Halaman Menu Utama

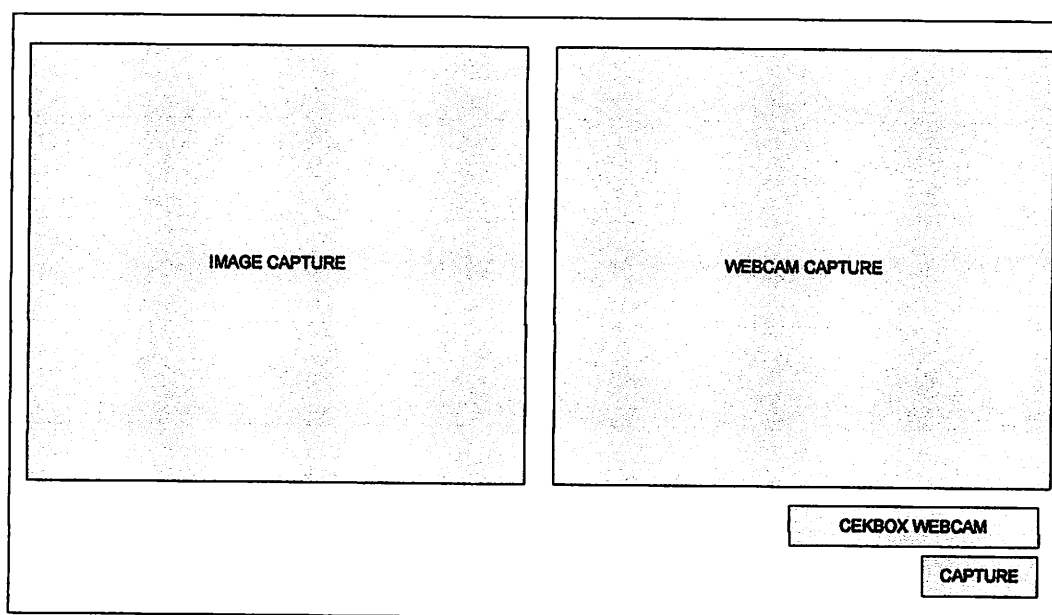
Brikut ini adalah beberapa penjelasan dari tampilan :

1. Image capture : Berfungsi untuk menampilkan gambar setelah di capture.
2. Webcam capture: Berfungsi untuk menampilkan gambar dari sensor camera ke aplikasi untuk di capture.
3. Deteksi 1 : Berfungsi sebagai proses penentuan koordinat.
4. Deteksi 2 : Berfungsi sebagai proses penentuan koordinat.
5. Zoom : Berfungsi untuk memperbesar gambar.
6. Manual : Berfungsi sebagai proses untuk mengolah gambar secara manual untuk menen tukan titik.
7. Browse : Berfungsi untuk mengambil file gambar yang udah tersedia kemudian di inport dalam aplikasi.

3.3.2 Perancangan Tampilan Menu Capture

Perancangan Tampilan Menu Capture merupakan *form* yang berisi aplikasi proses pengcapturan PCB untuk dijadikan image uji. dalam perancangan ini gambar akan diambil untuk dipindahkan ke aplikasi untuk di proses.

Berikut adalah gambar dari perancangan tampilan menu capture dari aplikasi proses pendeteksian titik PCB.

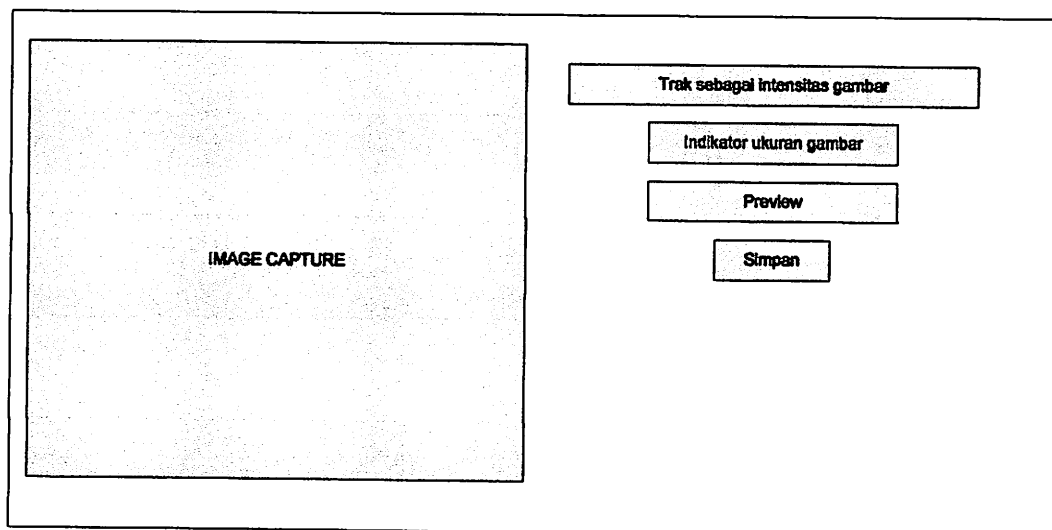


Gambar 3.8 Tampilan Halaman Menu Capture

3.3.3 Perancangan Tampilan Zoom

Perancangan Tampilan Menu zoom merupakan *form* yang berisi aplikasi proses pengolahan gambar untuk menjadikan image menjadi kecil apa besar sesuai keinginan. Dalam perancangan tampilan ini terdapat beberapa fungsi, pertama fungsi trek pada gambar untuk mengubah ukuran gambar menjadi besar maupun kecil, kemudian pada indikator akan mengikuti besar kecil gambar. setelah gambar di perbesar bisa klik tombol preview untuk melihat hasilnya, kemudian tombol simpan akan menyimpan gambar kemudian kembali ke menu utama untuk di deteksi.

Berikut adalah gambar dari perancangan tampilan menu zoom dari aplikasi proses pendeteksian titik PCB.

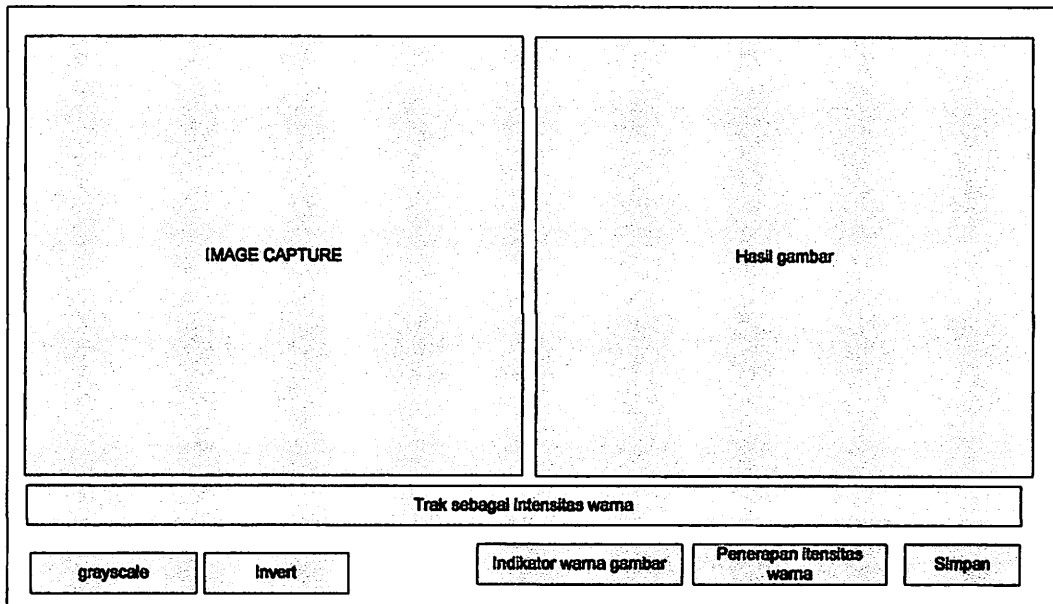


Gambar 3.9 Tampilan Halaman Menu Zoom

3.3.4 Perancangan tampilan Manual Proses

Perancangan Tampilan Menu Manual merupakan *form* yang berisi aplikasi proses pengolahan gambar untuk mendapat kan hasil yang maksimal untuk penentuan titik. Dalam hal ini proses manual berfungsi untuk menuncak penentuan titik dalam proses pencitraan. untuk itu terdapat tombol greyscale dan invert berfungsi untuk mengubah gambar menjadi hitam putih kemudian pada indikator warna itu berfungsi sebagai treshold untuk mendapatkan titik(ambang batas warna)yang hitam semakin hitam yang putih semakin putih.

Berikut adalah gambar dari perancangan tampilan menu manual dari aplikasi proses pendeteksian titik PCB.

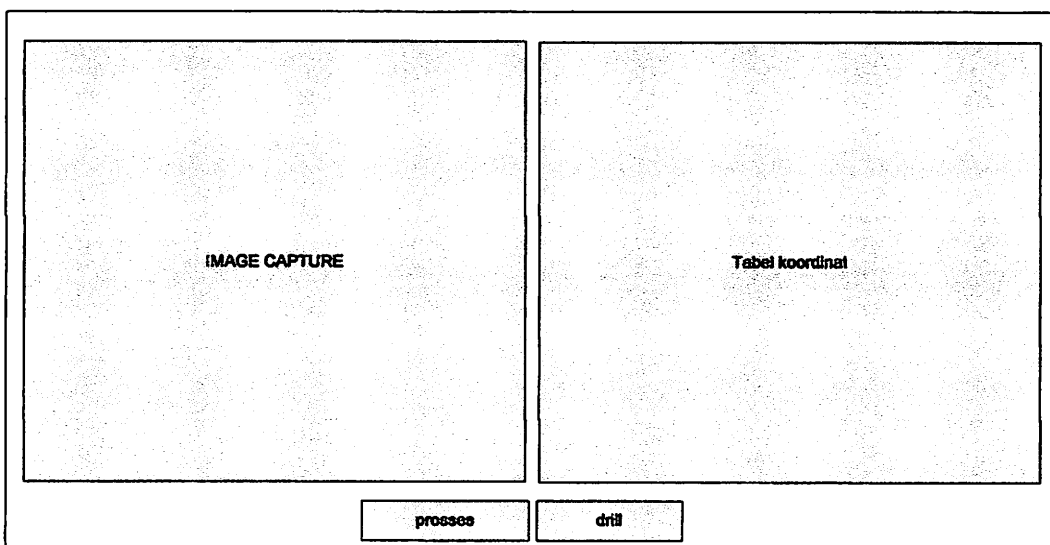


Gambar 3.10 Tampilan Halaman Menu Manual

3.3.5 Perancangan Tampilan Menu Proses

Perancangan Tampilan Menu Proses merupakan *form* yang berisi form pemrosesan image processing sistem mulai dari pendeteksian titik pada gambar PCB sampai add titik ke gambar hasil capture. pada tampilan ini aplikasi akan memproses penentuan koordinat pada gambar.

Berikut adalah gambar dari perancangan tampilan menu proses dari aplikasi proses deteksi titik PCB.

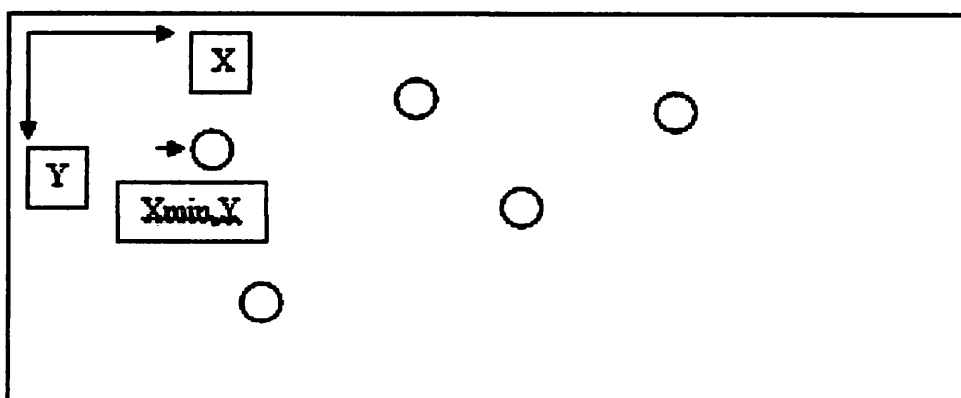


Gambar 3.11 Interface Halaman Menu Proses

3.4 Manual Proses Sistem

3.4.1 Manual Proses Sistem Penentuan Titik Lubang PCB

Proses penentuan titik lubang PCB yaitu proses dimana sistem mendeteksi tiap lubang PCB hingga dapat ditentukan letak tiap lubang yang ada dalam image PCB. proses manual penentuan titik pada image PCB dilakukan dengan mencari titik putih pertama yang dijadikan titik koordinat x_{minimal} pada image yang diduga titik drill PCB. Langkah untuk mencari titik koordinat dapat disimulasikan seperti gambar dibawah ini :

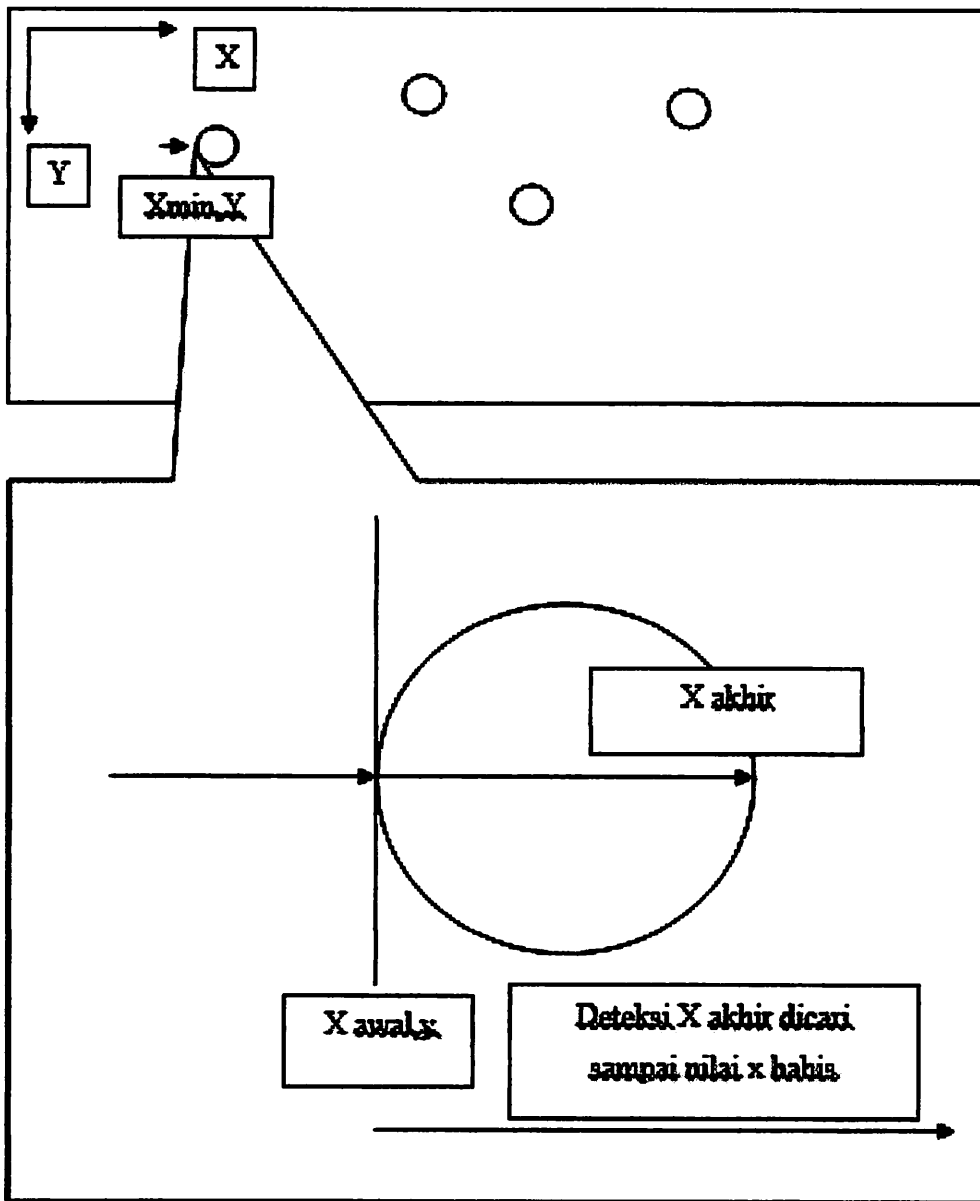


Gambar 3.12 Pengujian Titik

Dalam mencari titik pertama piksel dicari mulai dari titik koordinat x ujung atas kiri layar yang merupakan titik koordinat x minimal dari image ($x=0$) sampai ke kanan layar yang merupakan titik koordinat y minimal dari image ($y=0$), dan titik koordinat y di ujung atas kiri layar sampai ujung kanan bawah kanan layar.

Jika ada terdeteksi warna putih, maka itu dianggap koordinat x_{min} untuk titik suatu lubang drill PCB. Sedangkan koordinat y yang didapat dijadikan koordinat y awal pendeteksian koordinat Y_{min} dan koordinat Y_{maks} titik.

Selanjutnya untuk melakukan pengujian terhadap pencarian titik koordinat X_{maks} pada image PCB dilakukan dengan mencari titik putih koordinat x terakhir dari koordinat y dari koordinat x_{min} . Langkah untuk mencari titik koordinat X_{maks} dapat disimulasikan seperti gambar dibawah ini :

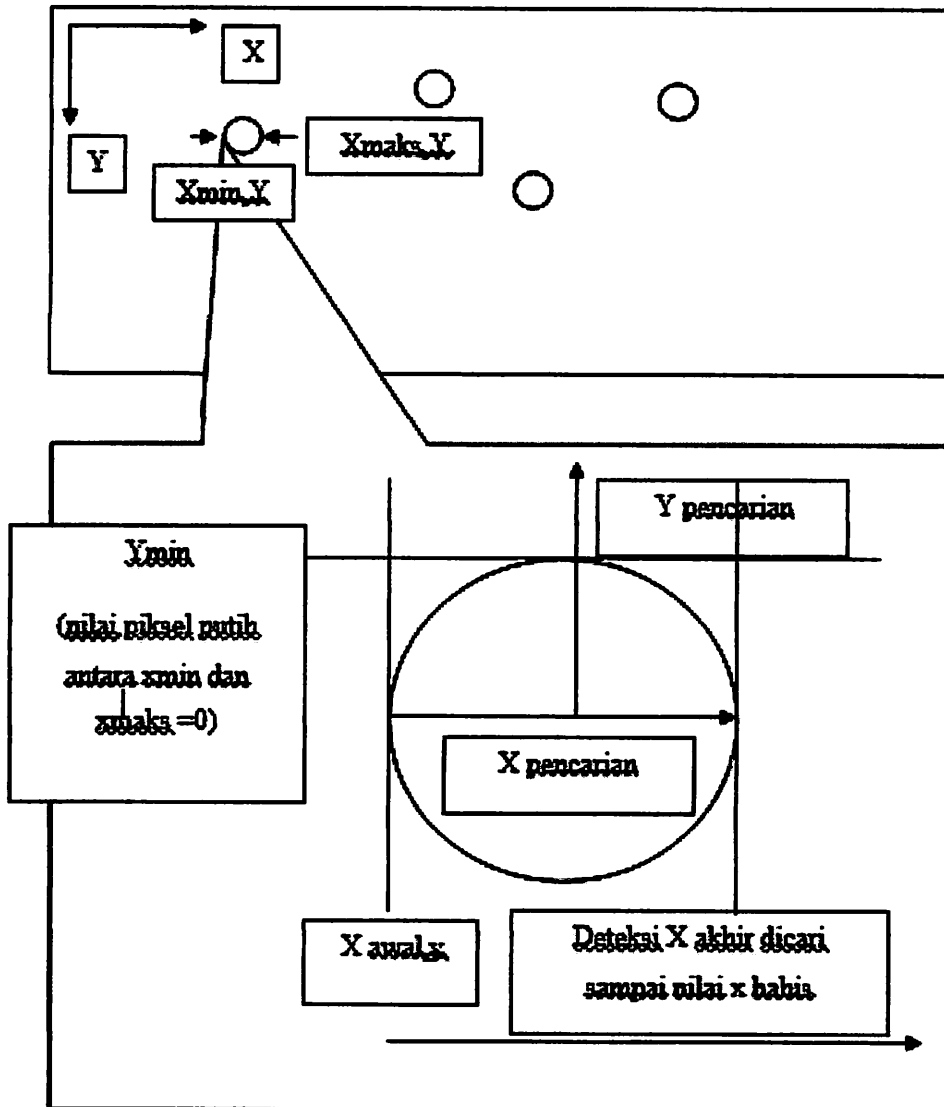


Gambar 3.13 Pengujian Titik X

Dalam mencari titik kedua adalah titik koordinat x maksimal dimana piksel ini dicari mulai dari titik koordinat x dari koordinat xmin dan koordinat y dari pencarian koordinat xmin, sampai koordinat x image yang merupakan panjang dari image dengan menggunakan koordinat y dari pencarian xmin yang telah didapat sebelumnya.

Sementara untuk melakukan pengujian terhadap pencarian titik Ymin titik pada image PCB dilakukan dengan mencari total titik putih yang bernilai 0 untuk

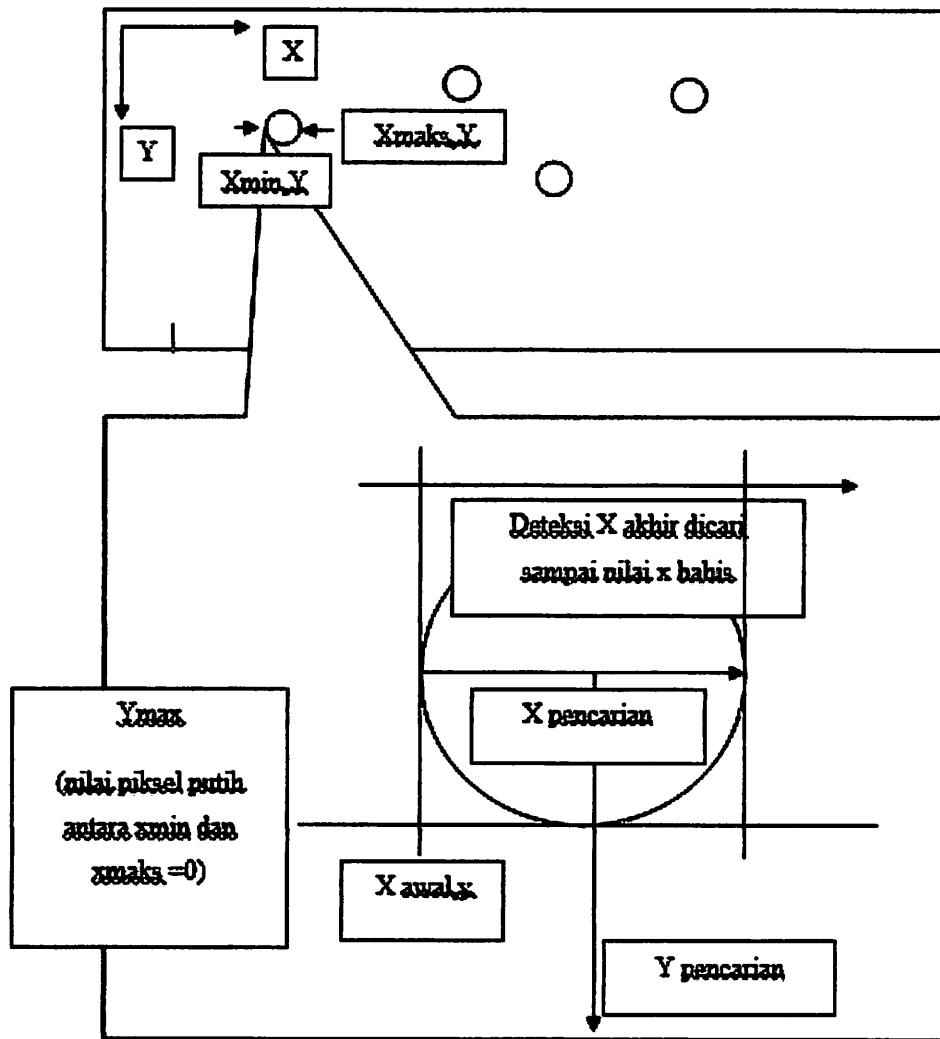
tiap piksel antara koordinat x_{min} dan koordinat x_{maks} dan koordinat y dari koordinat x_{min} menuju koordinat $y = 0$. Langkah untuk mencari titik koordinat Y_{min} dapat disimulasikan seperti gambar dibawah ini :



Gambar 3.14 Pengujian Titik Y

Dalam mencari titik kedua adalah titik $y_{minimal}$ dimana piksel ini dicari mulai dari titik koordinat x dari koordinat x_{min} dan koordinat y dari pencarian koordinat x_{min} , pencarian dimulai dari titik koordinat x_{min} sampai koordinat x image dengan menggunakan koordinat y dari pencarian koordinat x_{min} menuju koordinat y awal atau sama dengan 0.

Sementara untuk melakukan pengujian terhadap pencarian titik koordinat Y_{max} pada image PCB dilakukan dengan mencari total titik putih yang bernilai 0 untuk tiap piksel antara koordinat x_{min} dan koordinat x_{maks} dan koordinat y dari koordinat x_{min} menuju koordinat y image. Langkah untuk mencari titik koordinat Y_{max} dapat disimulasikan seperti gambar dibawah ini :

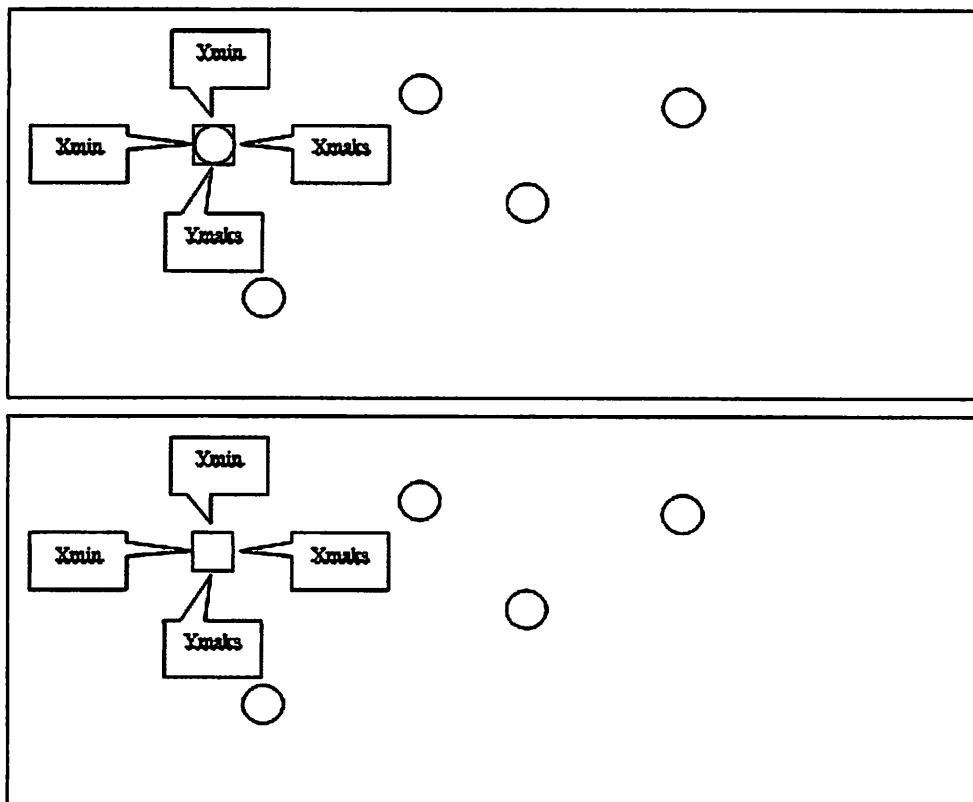


Gambar 3.15 Pengujian Titik Image

Dalam mencari titik kedua adalah titik koordinat y maksimal dimana piksel ini dicari mulai dari titik koordinat x dari koordinat x_{min} dan koordinat y dari pencarian koordinat x_{min} , pencarian dimulai dari titik koordinat x_{min} sampai

koordinat x image dengan menggunakan koordinat y dari pencarian xmin menuju koordinat y image atau sama dengan tinggi image.

Selanjutnya agar titik PCB yang telah terdeteksi koordinat xmin, koordinat xmax, koordinat ymin dan koordinat ymax tidak terdeteksi kembali, maka dihilangkan dari image dan hasil koordinat disimpan dalam database. Langkah untuk menghapus titik koordinat yang telah dideteksi dapat disimulasikan seperti gambar dibawah ini :



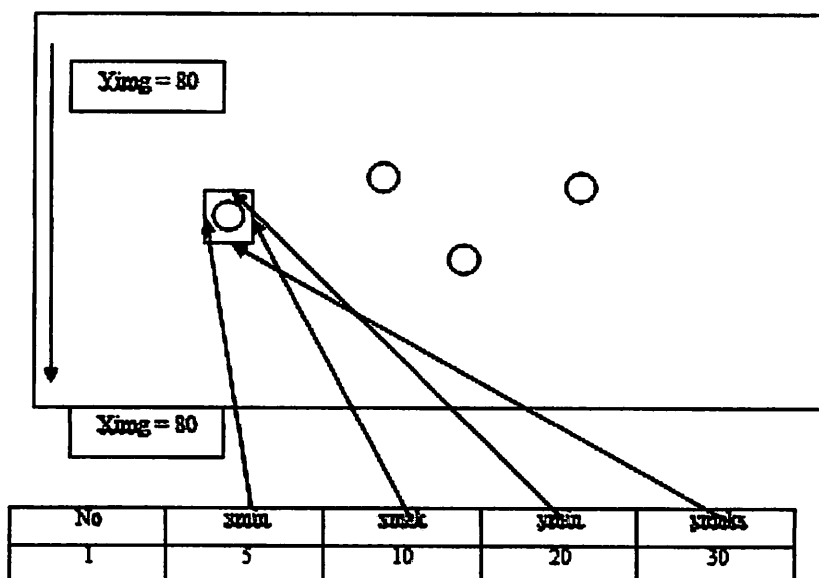
Gambar 3.16 Pengujian Pengambilan Titik

Dalam menghapus titik yang telah dideteksi, maka akan diganti piksel yang berada dalam area koordinat xmin, koordinat xmax serta koordinat ymin, koordinat ymax menjadi piksel berwarna hitam yang berarti diganti dengan warna background.

3.4.2 Manual Proses Sistem penentuan koordinat titik

Sistem penentuan koordinat titik adalah menentukan titik PCB yang telah disimpan pada database, maka data yang telah tersimpan diload ke form simulasi DRILL, untuk mengetahui letak tiap titik yang terdeteksi.

Proses manual dari letak koordinat hasil deteksi titik PCB ini dapat digambarkan pada gambar dibawah ini :



Gambar 3.17 Penentuan Koordinat

Penjelasan dari gambar proses manual diatas yaitu pada saat data koordinat diload, maka sistem mengambil 4 field data yaitu xmin, xmax, ymin dan ymax keempat data itu dimanfaatkan untuk membuat gambar kotak yang mengelilingi titik lubang yang telah ditentukan.

BAB IV

HASIL DAN PEMBAHASAN

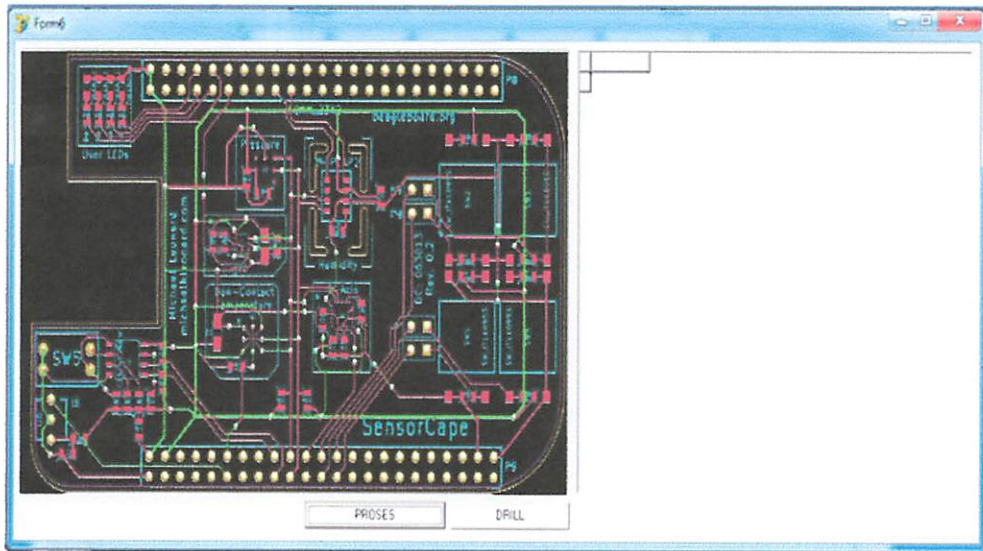
Bab ini menjelaskan tentang hasil yang diperoleh berdasarkan pengujian yang telah dilakukan. Tujuan dari dilakukannya pengujian adalah mengetahui apakah pendeteksian titik PCB pada image hasil capture menggunakan webcam memberikan hasil yang sesuai dengan perencanaan atau tidak. Berikut adalah 3 tahap pengujian yang dilakukan:

1. Pengujian deteksi titik PCB.
2. Pengujian letak koordinat hasil deteksi titik PCB

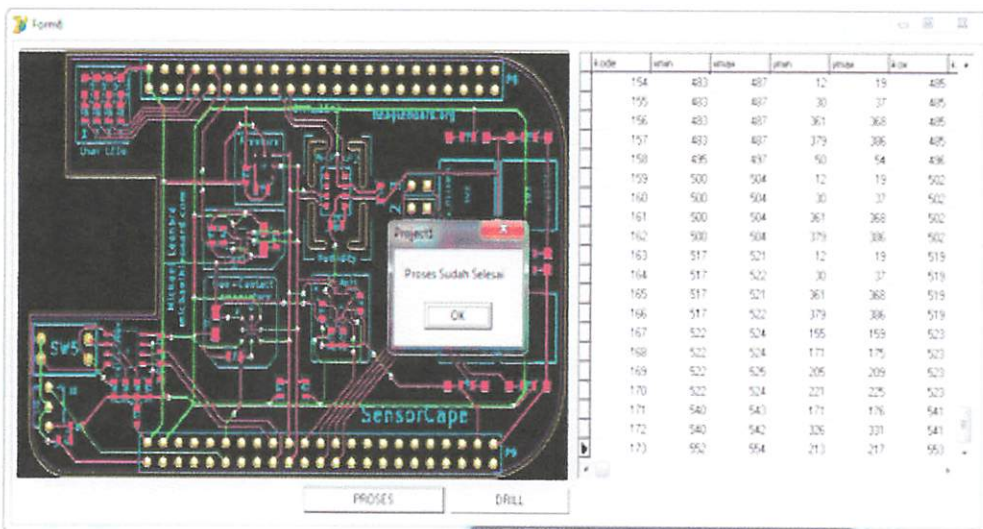
4.1 Pengujian deteksi titik PCB

Pengujian deteksi titik PCB yaitu pengujian dari proses saat sistem mencari dan menentukan tiap lubang pada image PCB, hingga nanti didapat koordinat titik dari tiap lubang yang ada pada image untuk disimpan didalam database.

Tampilan interface dari proses pengujian pencarian titik drill pada image PCB akan ditampilkan pada gambar dibawah ini :



Gambar 4.1 diatas merupakan capture aplikasi ketika sebelum dilakukan proses pendeteksian titik PCB.

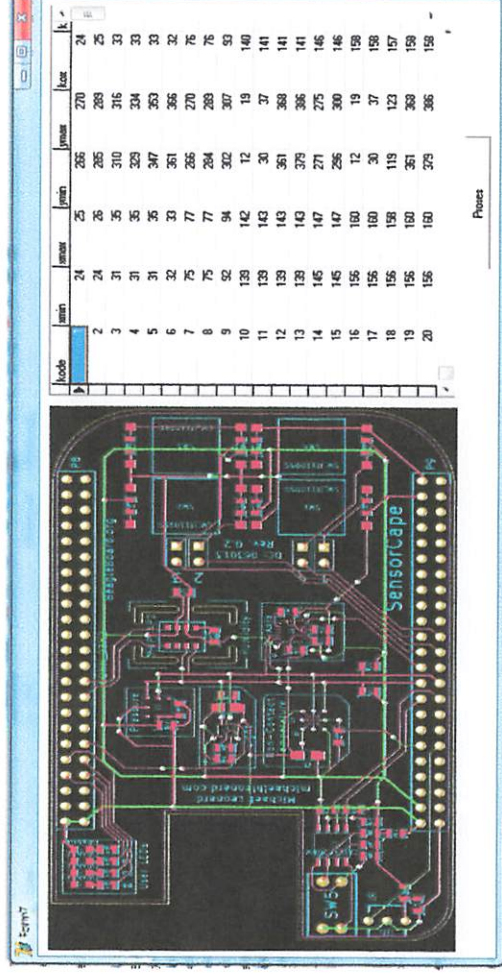


Gambar 4.2 diatas merupakan capture aplikasi ketika setelah dilakukan proses pendeteksian titik PCB dan posisi koordinat PCB telah diketahui.

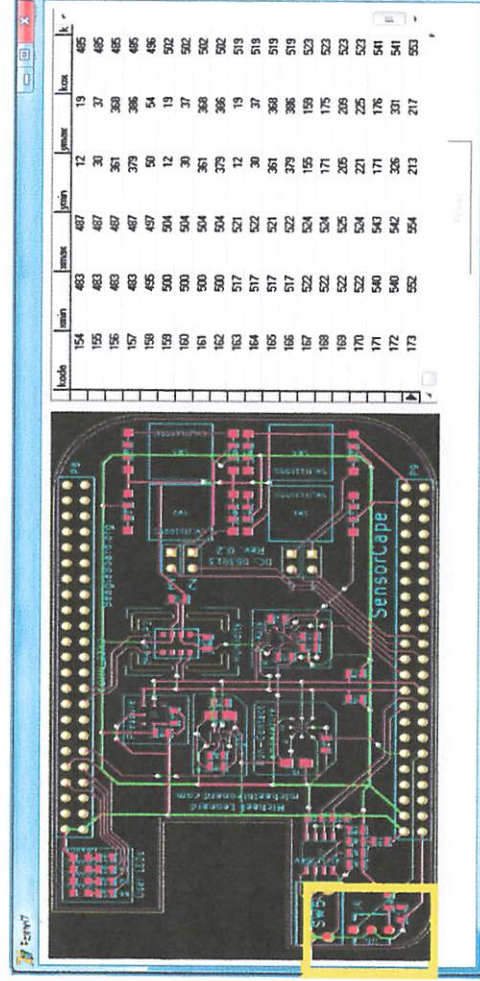
Dari hasil pengujian pada pendeteksian titik pada image PCB pada aplikasi, deteksi titik pada PCB pada aplikasi yang dibuat telah berhasil dan sesuai dengan perancangan jika yang dideteksi gambar blue print dari PCB.

4.2 Pengujian letak koordinat hasil deteksi titik PCB

Interface untuk Pengujian letak koordinat hasil deteksi titik PCB Akan dipelihatkan pada gambar dibawah ini :



Gambar 4.3 diatas merupakan capture aplikasi ketika sebelum dilakukan proses simulasi pendeteksian titik PCB.

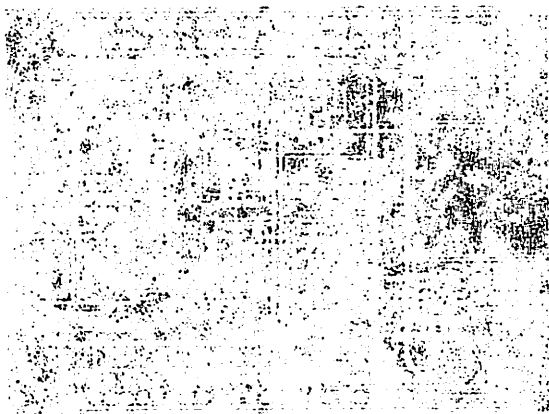


Gambar 4.4 diatas merupakan capture aplikasi ketika setelah 4 kali tahap pendeteksian titik drill PCB yang dilakukan pada proses simulasi pendeteksian titik PCB selesai dilakukan.

... .. 2.4

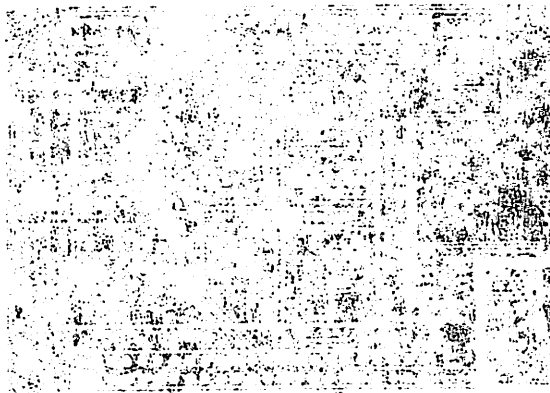
... ..

... ..



... ..

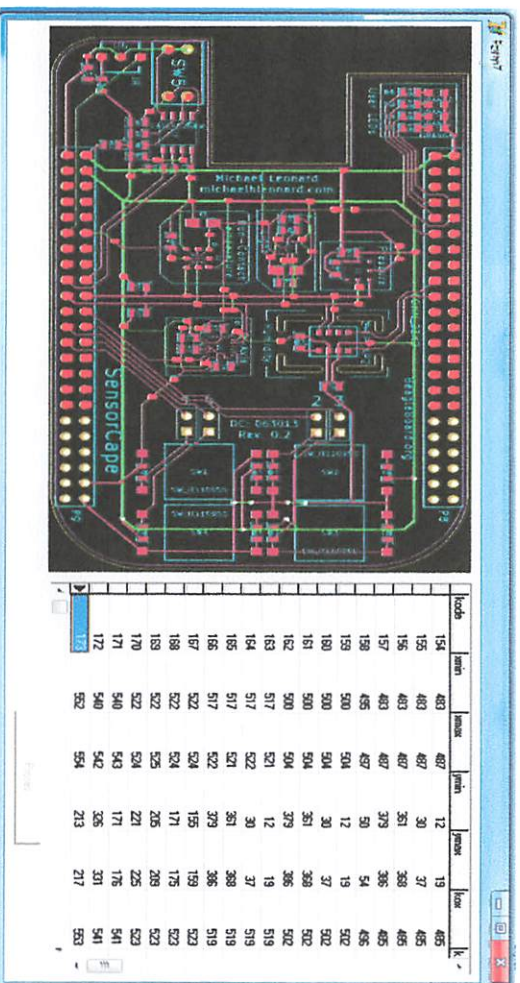
... ..



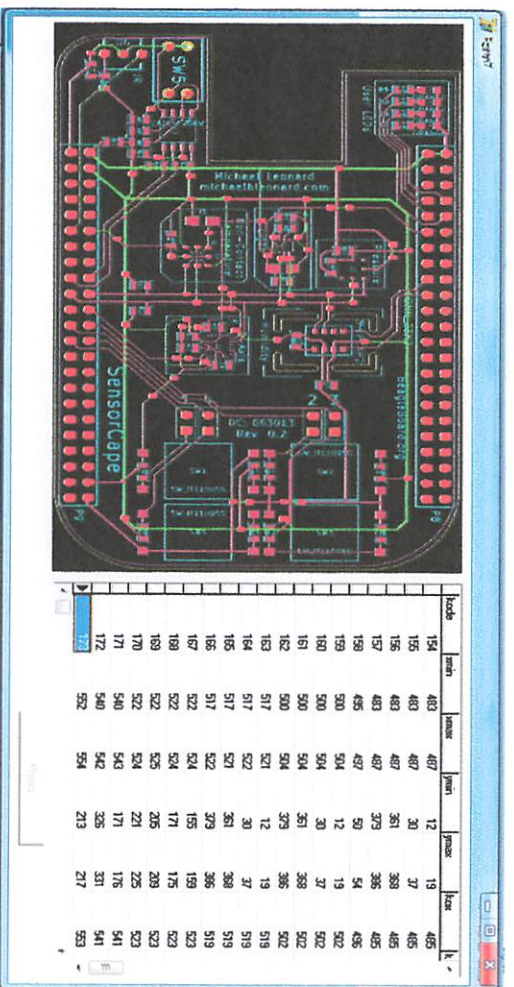
... ..

... ..

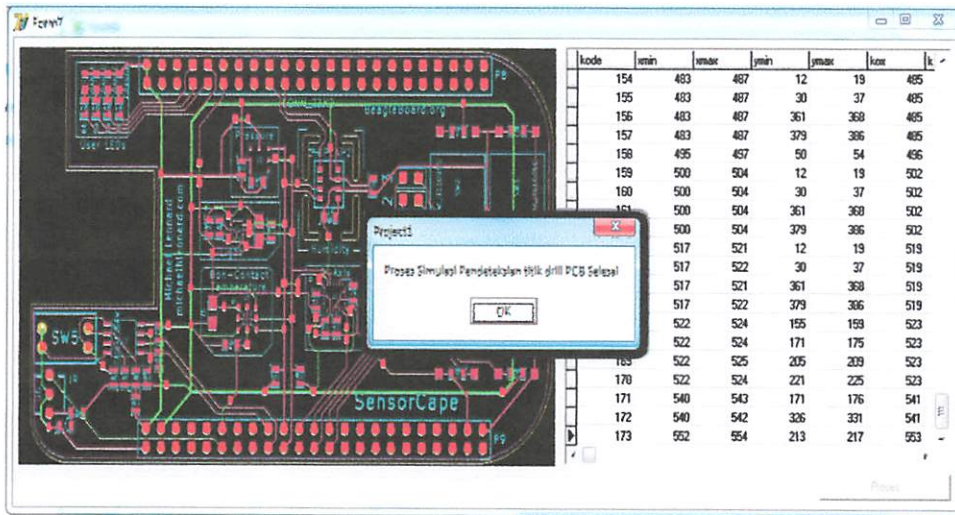
... ..



Gambar 4.5 diatas merupakan capture aplikasi ketika setelah lebih dari 3/4 bagian pada PCB tahap pendeteksian titik drill PCB yang dilakukan pada proses simulasi pendeteksian titik PCB selesai dilakukan.



Gambar 4.6 diatas merupakan capture aplikasi ketika setelah semua bagian pada PCB tahap pendeteksian titik drill PCB yang dilakukan selesai dilakukan pada proses simulasi pendeteksian titik PCB.



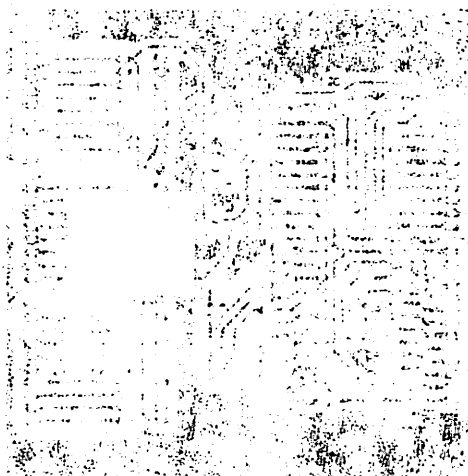
Gambar 4.7 diatas merupakan capture aplikasi ketika setelah semua bagian pada PCB tahap pendeteksian titik drill PCB yang dilakukan selesai dilakukan dan muncul pesan selesai pada proses simulasi pendeteksian titik PCB.

Dari hasil Pengujian letak koordinat hasil deteksi titik PCB data yang sudah tersimpan didatabase, ternyata dari setiap simulasi penentuan titik drill PCB telah mendekati sempurna atau telah berjalan dengan baik pada aplikasi yang telah dibuat. Hingga bisa dikatakan sesuai dengan perancanganyang telah dibuat sebelumnya.

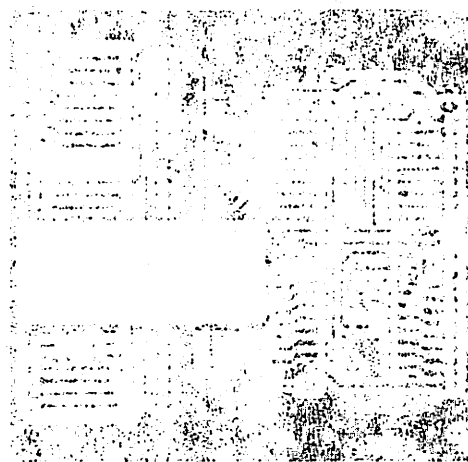
4.3 Analisa Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan terhadap 3 jenis data image yaitu image hasil capture, capture dengan sedikit operasi citra secara manual, serta image blueprint dari PCB maka dapat dianalisa hasilnya dan dapat dijelaskan hasil pengujiannya seperti dibawah ini.

Dari Data image Blue Print PCB yang telah di buka dan dideteksi titik serta koordinatnya melalui sistem browse, dapat diperlihatkan pada gambar dibawah ini :



1. The following information is for your information only.



2. The following information is for your information only.

Kode	Xmm	Ymm	Xmil	Ymil	Xmm	Ymm	Xmil	Ymil	Xmm	Ymm	Xmil	Ymil
174	402	405	402	405	351	368	351	368	404	404	404	404
175	405	405	405	405	362	405	362	405	405	405	405	405
176	405	405	405	405	368	368	368	368	105	105	105	105
177	417	417	422	422	66	66	66	66	72	72	72	72
178	417	417	422	422	108	108	108	108	141	141	141	141
179	417	417	422	422	168	168	168	168	196	196	196	196
180	417	417	422	422	208	208	208	208	232	232	232	232
181	417	417	422	422	238	238	238	238	264	264	264	264
182	417	417	422	422	268	268	268	268	296	296	296	296
183	417	417	422	422	298	298	298	298	328	328	328	328
184	417	417	422	422	328	328	328	328	360	360	360	360
185	413	421	421	421	328	328	328	328	392	392	392	392
186	433	433	433	433	362	362	362	362	424	424	424	424
187	433	433	433	433	392	392	392	392	456	456	456	456
188	433	433	433	433	424	424	424	424	488	488	488	488
189	433	433	433	433	456	456	456	456	520	520	520	520
190	433	433	433	433	488	488	488	488	552	552	552	552
191	433	433	433	433	520	520	520	520	584	584	584	584
192	433	433	433	433	552	552	552	552	616	616	616	616
193	433	433	433	433	584	584	584	584	648	648	648	648
194	433	433	433	433	616	616	616	616	680	680	680	680
195	433	433	433	433	648	648	648	648	712	712	712	712
196	433	433	433	433	680	680	680	680	744	744	744	744
197	433	433	433	433	712	712	712	712	776	776	776	776
198	433	433	433	433	744	744	744	744	808	808	808	808
199	433	433	433	433	776	776	776	776	840	840	840	840
200	433	433	433	433	808	808	808	808	872	872	872	872
201	433	433	433	433	840	840	840	840	880	880	880	880
202	433	433	433	433	880	880	880	880	912	912	912	912
203	433	433	433	433	912	912	912	912	944	944	944	944
204	433	433	433	433	944	944	944	944	976	976	976	976
205	433	433	433	433	976	976	976	976	1008	1008	1008	1008
206	433	433	433	433	1008	1008	1008	1008	1040	1040	1040	1040

Gambar 4.10. Pengujian Blueprint Koordinat b

Kode	Xmm	Ymm	Xmil	Ymil	Xmm	Ymm	Xmil	Ymil	Xmm	Ymm	Xmil	Ymil
178	417	417	422	422	189	189	189	189	141	141	141	141
179	417	417	422	422	209	209	209	209	156	156	156	156
180	417	417	422	422	228	228	228	228	171	171	171	171
181	417	417	422	422	248	248	248	248	186	186	186	186
182	417	417	422	422	267	267	267	267	199	199	199	199
183	417	417	422	422	287	287	287	287	213	213	213	213
184	417	417	421	421	298	298	298	298	224	224	224	224
185	413	421	421	421	308	308	308	308	235	235	235	235
186	433	433	433	433	318	318	318	318	244	244	244	244
187	433	433	433	433	328	328	328	328	254	254	254	254
188	433	433	433	433	338	338	338	338	264	264	264	264
189	433	433	433	433	348	348	348	348	274	274	274	274
190	433	433	433	433	358	358	358	358	284	284	284	284
191	433	433	433	433	367	367	367	367	294	294	294	294
192	433	433	433	433	377	377	377	377	304	304	304	304
193	433	433	433	433	387	387	387	387	314	314	314	314
194	433	433	433	433	397	397	397	397	324	324	324	324
195	433	433	433	433	407	407	407	407	334	334	334	334
196	433	433	433	433	417	417	417	417	344	344	344	344
197	433	433	433	433	427	427	427	427	354	354	354	354
198	433	433	433	433	437	437	437	437	364	364	364	364
199	433	433	433	433	447	447	447	447	374	374	374	374
200	433	433	433	433	457	457	457	457	384	384	384	384
201	433	433	433	433	467	467	467	467	394	394	394	394
202	433	433	433	433	477	477	477	477	404	404	404	404
203	433	433	433	433	487	487	487	487	414	414	414	414
204	433	433	433	433	497	497	497	497	424	424	424	424
205	433	433	433	433	507	507	507	507	434	434	434	434
206	433	433	433	433	517	517	517	517	444	444	444	444

Gambar 4.11. Pengujian Blueprint penentuan drill b

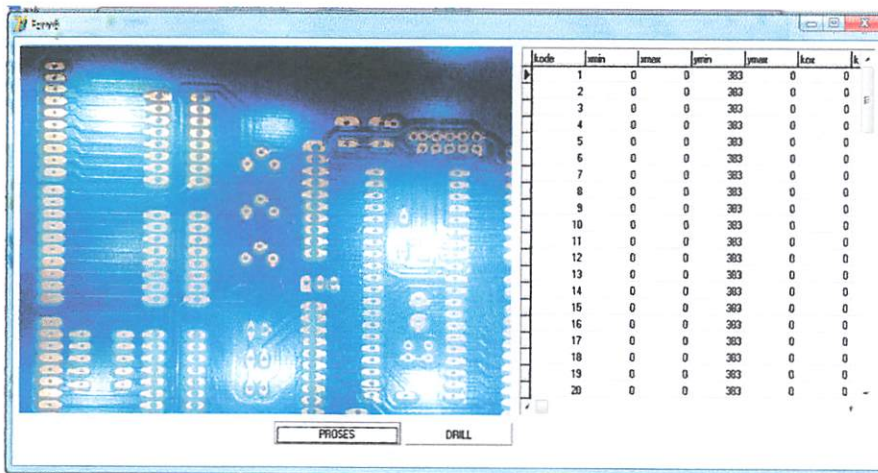
Dari hasil pengujian terhadap image blueprint PCB data hasil hasil pengujian dari data image yang telah dicapture dan dideteksi dapat diperlihatkan pada gambar dibawah ini:

Tabel 4.1 Hasil Pengujian Image Blueprint

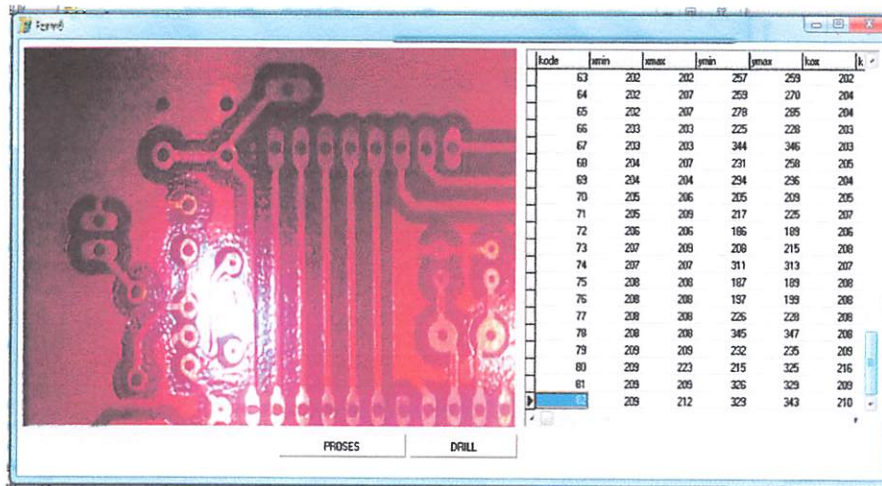
Percobaan	Jumlah Titik Drill	Terdeteksi	Drill Benar	Error (%)
1	245	245	224	0,85
2	206	206	195	0.53
Rata	225.5	225.5	209.5	0.69

Berdasarkan perbandingan nilai Jumlah titik yang terdeteksi dan drill yang benar maka didapat hasil error untuk image PCB 1 sebesar 0.85 %, image PCB 2 sebesar 0.53 %. Dengan rata rata error yang didapat sebesar 0.69 % dan jumlah rata rata titik drill 225.5 dan yang terdeteksi 225.5 dengan ketepatan lubang drill yang terdeteksi sebanyak 209.5.

Kemudian untuk data image PCB yang telah di capture dan dideteksi titik serta koordinatnya melalui webcam secara langsung, dapat diperlihatkan pada gambar dibawah ini :



Gambar 4.12. Pengujian Menggunakan Capture a



Gambar 4.13. Pengujian Menggunakan Capture b

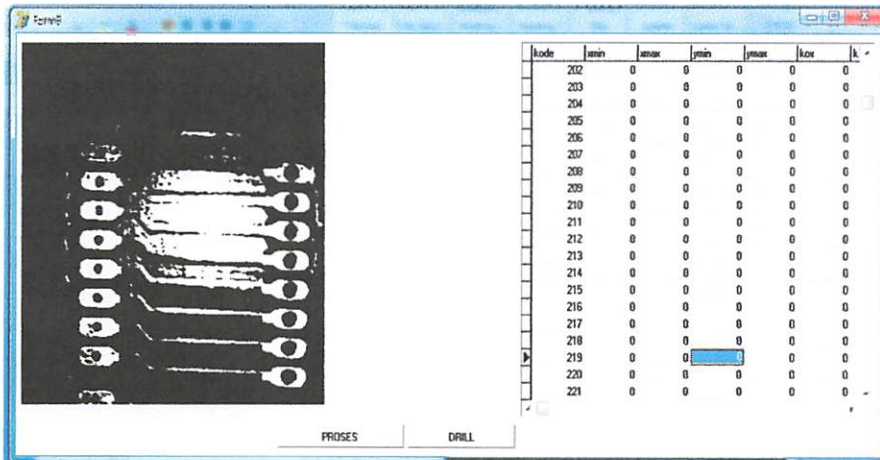
Sementara itu untuk data hasil hasil pengujian dari data image yang telah dicapture dan dideteksi dapat diperlihatkan pada gambar dibawah ini:

Tabel 4.2 Pengujian Image Capture Webcam

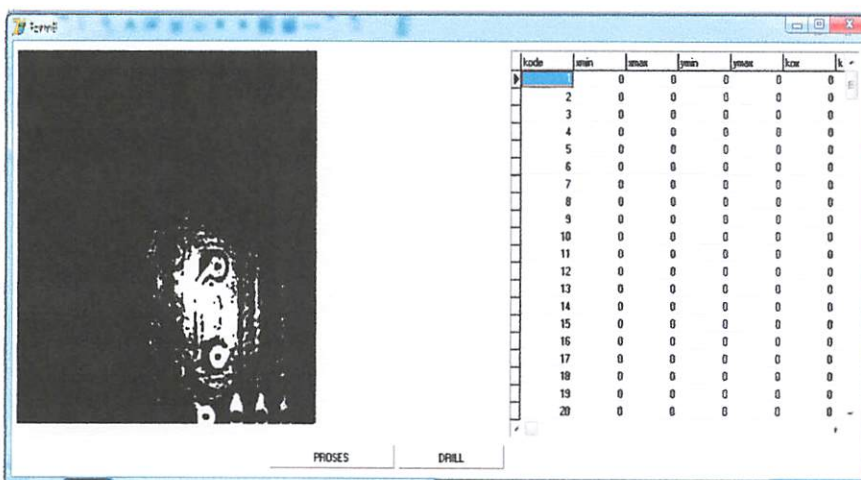
Percobaan	Jumlah Titik Drill	<i>Terdeteksi</i>	<i>Drill Benar</i>	<i>Error (%)</i>
1	150	150	15	90
2	180	180	1	99.4
Rata	165	165	8	95.1

Berdasarkan perbandingan nilai Jumlah titik yang terdeteksi dan drill yang benar maka didapat hasil error untuk image PCB 1 sebesar 90 %, image PCB 2 sebesar 99.4 %. Dengan rata rata error yang didapat sebesar 95.1 % dan jumlah rata rata titik drill 152 dan yang terdeteksi 152 dengan ketepatan lubang drill yang terdeteksi sebanyak 8.

Kemudian untuk data image PCB yang telah di capture menggunakan webcam dan diubah dulu posisi warna secara manual serta dideteksi titik serta koordinatnya, dapat diperlihatkan pada gambar dibawah ini :



Gambar 4.14. Pengujian Menggunakan Manual Capture a



Gambar 4.15. Pengujian Menggunakan Manual Capture b

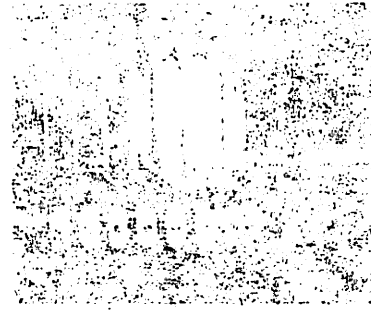
Dan untuk data hasil hasil pengujian dari data image yang telah dicapture dan di ubah dideteksi dapat diperlihatkan pada gambar dibawah ini:

THE UNIVERSITY OF CHICAGO PRESS
50 EAST LAKE STREET, CHICAGO, ILL. 60607

THE UNIVERSITY OF CHICAGO PRESS
50 EAST LAKE STREET, CHICAGO, ILL. 60607



THE UNIVERSITY OF CHICAGO PRESS
50 EAST LAKE STREET, CHICAGO, ILL. 60607



Tabel 4.3 Hasil Pengujian Pengujian Image Capture dengan Operasi Manual

Percobaan	Jumlah Titik Drill	<i>Terdeteksi</i>	<i>Drill Benar</i>	<i>Error (%)</i>
1	150	221	5	96.3
2	180	194	3	98.7
Rata	165	207.5	4	97.5

Berdasarkan perbandingan nilai Jumlah titik yang terdeteksi dan drill yang benar maka didapat hasil error untuk image PCB 1 sebesar 96.3 %, image PCB 2 sebesar 98.7 %. Dengan rata rata error yang didapat sebesar 97.5 % dan jumlah rata rata titik drill 165 dan yang terdeteksi 207.5 dengan ketepatan lubang drill yang terdeteksi sebanyak 4.

Jadi hasil komparasi jumlah persentase error dari ketiga jenis pengujian dapat dilihat pada tabel berikut ini :

Tabel 4.4 Hasil Pengujian keseluruhan

Blueprint(%)	Capture(%)	Manual(%)
0.69	95.1	97.5

Berdasarkan hasil komparasi diatas dapat dipastikan bahwa sistem pendeteksian titik PCB yang paling bagus menggunakan layout dari blueprint PCB dengan error rata-rata sebesar 0.69% atau nyaris semua titik terdeteksi sebagai titik drill PCB.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan perancangan dan pembuatan aplikasi penentuan titik pada PCB, serta menguji aplikasi. Kesimpulan yang dapat diambil adalah:

1. Perancangan dan pembuatan aplikasi penentuan titik pada PCB sudah dapat dioperasikan sesuai rancangan.
2. Dalam memproses gambar susah menentukan titik pengeboran jika titik (warna) sama dengan Background.
3. Deteksi webcam dan deteksi titik menggunakan image processing sudah berjalan sesuai perancangan.
4. Pada PCB intensitas cahaya juga berpengaruh pada pengambilan gambar.
5. Deteksi titik memakai image blueprint telah berhasil dengan error hanya 0.69%, namun jika memakai image webcam dianggap gagal gagal karena mempunyai error 95.1%, begitupun dengan memakai deteksi manual yaitu mengubah warna hasil capture web camera yang mempunyai eror lebih dari 97.5%.
6. Bisa dikatakan pendeteksian penentuan titik PBC yang paling bagus menggunakan blueprint dari layout PCB.

5.2 Saran

Dari mulai pembuatan sampai penyelesaian skripsi ini ada beberapa hal penulis yang ingin di sampaikan untuk pengembangan Aplikasi ini. Meskipun Aplikasi ini sudah dapat bekerja masih ada hal-hal yang perlu di optimalkan, baik dari segi hardware maupun software. Beberapa saran yang penulis berikan:

1. Untuk pengembangan sebaiknya diterapkan sistem blue print karena titik koordinat dari blueprint dapat terdeteksi secara jelas.

2. Perlu penambahan fitur lainnya seperti memakai teknik scan dengan memakai sistem scan terbaik agar hasil image blueprint lebih terlihat detail dan jelas.
3. Perlu menggunakan metode yang lebih kompleks agar hasil capture bisa lebih jernih lagi.
4. perlu penambahan kamera yang lebih berkualitas untuk mendapatkan hasil yang lebih bagus.

DAFTAR PUSTAKA

- [1] T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, "Teori Pengolahan Citra Digital", Andi Yogyakarta dan UDINUS Semarang, 2009.
- [2] Purnomo Mauridhi Hery, Muntasa Arif, " Konsep Pengolahan Citra Digital dan Ekstraksi Fitur ", Graha Ilmu Yogyakarta, 2010.
- [3] Yusronrijal. 2012., "Transformasi Intensitas Citra". <https://yusronrijal.wordpress.com/>. Diakses pada tanggal 31 Agustus 2015.
- [4] Rahman, S. 2013., "THRESHOLDING". <http://saycommunity.blogspot.com/2013/12/thresholding.html>. Diakses pada tanggal 31 Agustus 2015.
- [5] Yilmaz, Javed, Shah. 2006., "Representasi Objek". <http://library.binus.ac.id/eColls/eThesisdDoc/Bab2/TSA-2014-0070%202.pdf>. Diakses pada tanggal 31 Agustus 2015.
- [6] Bradski, Kaehler. 2008., "Segmentasi". <http://library.binus.ac.id/eColls/eThesisdDoc/Bab2/TSA-2014-0070%202.pdf>. Diakses pada tanggal 31 Agustus 2015.
- [7] Lu, 1999., "Ekstrasi Fiture". <http://library.binus.ac.id/eColls/eThesisdDoc/Bab2/TSA-2014-0070%202.pdf>. Diakses pada tanggal 31 Agustus 2015.
- [8] Wikipedia. 2015., "Sensor kamera". https://id.wikipedia.org/wiki/Kamera_digital. Diakses pada tanggal 31 Agustus 2015.
- [9] Wikipedia. 2015., "Webcam". https://id.wikipedia.org/wiki/Kamera_digital. Diakses pada tanggal 31 Agustus 2015.
- [10] Sayhellow. 2011., "Pengertian Delphi". <https://linawidia.wordpress.com/2011/09/18/pengertian-delphi/>. Diakses pada tanggal 31 Agustus 2015.

- [11] Saputro, T. 2012., "Pengertian Microsoft Access". <http://trisaputro1996.blogspot.com/2012/11/pengertian-microsoft-access.html>. Diakses pada tanggal 31 Agustus 2015.
- [12] Rinaldi, M. 2004., " TEORI CITRA BITMAP". <http://sipangenauli.blogspot.com/2012/05/teori-citra-bitmap.html>. Diakses pada tanggal 31 Agustus 2015.
- [13] Thiang, Sherwin R.U. Sompie, " Mesin Bor Otomatis dengan Menggunakan Kamera untuk Mendeteksi Koordinat Bor", Jurusan Teknik Mesin, Fakultas Teknologi Industri, Universitas Kristen Petra, Oktober 2002. <http://puslit.petra.ac.id/journals/mechanical/>
- [14] Thiang, Sherwin R.U. Sompie, "Penggunaan Kamera sebagai Sensor Pendeteksi Koordinat Bor pada Mesin Bor Otomatis", Prosiding Industrial Electronics Seminar, ITS-Surabaya, Oktober 2002.

LAMPIRAN



PERMOHONAN PERSETUJUAN SKRIPSI

Yang Bertanda Tangan Dibawah Ini:

Nama : Syamsul Deska Wijaya
 NIM : 1112524
 Semester : VIII (delapan)
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-I
 Konsentrasi : TEKNIK ENERGI LISTRIK
TEKNIK ELEKTRONIKA
TEKNIK KOMPUTER DAN INFORMATIKA
TEKNIK KOMPUTER
TEKNIK TELEKOMUNIKASI
 Alamat : Perum Permatas Regency 2 ngiso Blok A.11 No 19 Malang

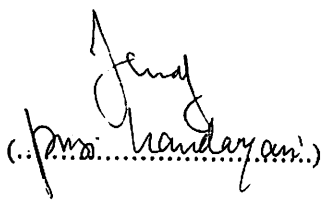
Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat SKRIPSI Tingkat Sarjana. Untuk melengkapi permohonan tersenut, bersama ini kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan- persyaratan pengambilan SKRIPSI adalah sebagai berikut:

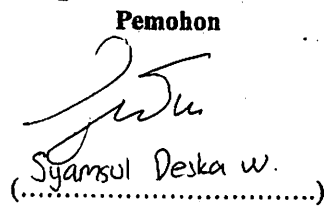
1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh matakuliah > 134 sks dengan IPK > 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar Skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

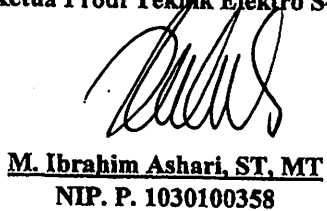
Telah diteliti kebenarannya data tersebut diatas
 Recording Teknik Elektro S-I

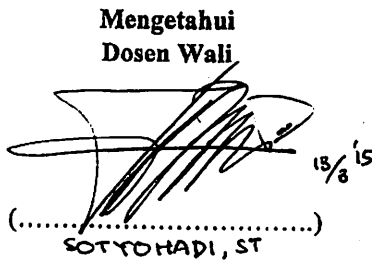

 (.....)

Malang, 13 - Maret - 2015

Pemohon

 (Syamsul Deska w.)

Disetujui
 Ketua Prodi Teknik Elektro S-I


M. Ibrahim Ashari, ST, MT
 NIP. P. 1030100358

Mengetahui
 Dosen Wali

 (.....)
 SOTTOHADI, ST

Catatan:


Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Prodi T. elektro S-I

1. Total sks x Bobot : 457,5
2. Total sks Pembagi : 138
3. IPK : 3.32
4. Skripsi di program semester genap 14/15

BERITA ACARA RAPAT PERSETUJUAN JUDUL/PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer S1



Tanggal :

1.	NIM	11.12.524
2.	Nama	Syamsul Deska Wijaya.
3.	Judul yang diajukan	sistem pendeteksi untuk menentukan lubang pada PCB menggunakan sensor kamera.
4.	Disetujui/Ditolak	
5.	Catatan:	
6.	Pembimbing yang diusulkan:	
	1.	Aryanto ✓
	2.	Korng ✓
Menyetujui		
1. Koordinator Dosen Kelompok Keahlian		
		
2. Dosen Kelompok Keahlian (Terlampir)		

* : Coret yang tidak perlu



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-187/EL-FTI/2015 Tanggal, 28 Mei 2015
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu **Dr. Eng. Aryunto Soetedjo, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **SYAMSUL DESKA WIJAYA**
Nim : **1112524**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

“ Semester Genap Tahun Akademik Genap 2014 - 2015 “

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui
Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-187/EL-FTI/2015 Tanggal, 28 Mei 2015
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu **Dr. Eng. I Komang Soma Wirata, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **SYAMSUL DESKA WIJAYA**
Nim : **1112524**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

“ Semester Genap Tahun Akademik Genap 2014 - 2015 “

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

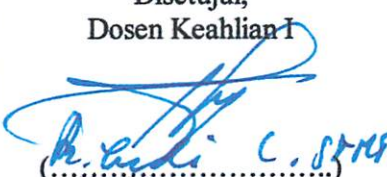
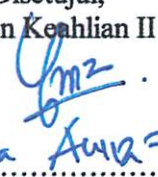

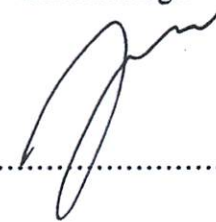


Mengetahui
Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358



BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer

1.	Nim	: 1112524		
2.	Nama	: SYAMSUL DESKA WIJAYA		
3.	Konsentrasi Jurusan	: Teknik Komputer		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
	13 April 2015	09:00	III.1.5	
5.	Judul proposal yang diseminarkan Mahasiswa	SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG PADA PCB MENGGUNAKAN SENSOR KAMERA		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan :	- Fokus ke Image Processing - Pelajari lebih detail		
8.	Catatan :			
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	 (.....)	 (.....)	(.....)	
Mengetahui, Ketua Program Studi Teknik Elektro S-1	Disetujui, Calon Dosen Pembimbing ybs			
	Pembimbing I	Pembimbing II		
 M. Ibrahim Ashari, ST, MT NIP. P 1030100358	 (.....)	(.....)		



Formulir Perbaikan Ujian Skripsi

Dalam Pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T.Energi Listrik,
T. Elektronika, (T. Komputer) / T. Telekomunikasi, Maka Perlu Adanya Perbaikan Skripsi Untuk Mahasiswa:

Nama : Sekarsih
NIM : 1112524
Perbaikan Melalui :

- 1) Penulisan nama pada bab II perlu diperbaiki
dengan kalimat yg diambil dari buku referensi
yang ada dalam
- 2) Konsentrasi teknik (Energi Listrik) ?
pada lembar persetujuan.
- 3) Bahas Format Bab (Bab I) di bab II

Malang.....10 - 10 -2015

[Signature]
Dr. F. Edi (S.S)



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karangla, Km. 2 MALANG

Formulir Perbaikan Ujian Skripsi

Dalam Pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T.Energi Listrik, /
T. Elektronika, /T. Komputer, / T.Telekomunikasi, Maka Perlu Adanya Perbaikan Skripsi Untuk Mahasiswa:

Nama : Syamsul Dasuq Wijaya
NIM : 11 12 524
Perbaikan Meliputi :

Revisi label penulisan

Malang, 10 Agustus 2015


(Yuli Wahyuni ISTMT)






PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi Program studi Teknik Elektro jenjang strata satu (S-1) yang diselenggarakan pada :

Hari : Selasa
Tanggal : 18 Agustus 2015

Telah dilakukan perbaikan skripsi oleh :

Nama : Syamsul Deska Wijaya
NIM : 1112524
Perogram Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Judul Skripsi : **SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG PADA PCB MENGGUNAKAN SENSOR KAMERA**

No	Materi Perbaikan	Keterangan
1.	Penulisan acuan diperbaiki pada BAB II	
2	Penulisan kosentrasi pada lembar persetujuan	
3	Pembuatan format BMP(Bitmap) pada BAB II	

Dosen Penguji I



Dr. F. Yudi Limpraptono, ST, MT
NIP. Y. 1039500274

Dosen Pembimbing I



Dr. Eng. Arynanto Soetedjo, ST, MT
NIP. Y. 1030800417

Dosen Pembimbing II



Dr. Eng. I Komang Somawirata, ST, MT
NIP. P.1030100361




PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi Program studi Teknik Elektro jenjang strata satu (S-1) yang diselenggarakan pada :

Hari : Selasa
Tanggal : 18 Agustus 2015

Telah dilakukan perbaikan skripsi oleh :

Nama : Syamsul Deska Wijaya
NIM : 1112524
Perogram Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Judul Skripsi : **SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG PADA PCB MENGGUNAKAN SENSOR KAMERA**

No	Materi Perbaikan	Keterangan
1.	Revisi Total Penulisan	

Dosen Penguji II



Yuli Wahyuni, ST, MT
NIP. P. 1031200456

Dosen Pembimbing I



Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP. Y. 1030800417

Dosen Pembimbing II



Dr. Eng. I Komang Somawirata, ST, MT
NIP. P.1030100361



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : SYAMSUL DESKA WIJAYA
NIM : 11.12.524
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Masa Bimbingan : Semester Genap 2014-2015
Judul : SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG
PADA PCB MENGGUNAKAN SENSOR KAMERA

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 18 Agustus 2015
Nilai : 76.5 (B+)

Panitia Ujian Skripsi:

Majelis Ketua Penguji

M. Ibrahim Ashari, ST, MT
NIP. P. 1030100358

Sekretaris Majelis Penguji

Dr. Eng. I Komang Somawirata, ST, MT
NIP. P. 1030100361

Anggota Penguji:

Penguji Pertama

Dr. F. Yudi Limpraptono, ST, MT
NIP. Y. 1039500274

Penguji Kedua

Yuli Wahyuni, ST, MT
NIP. P. 1031200456



MONITORING BIMBINGAN SKRIPSI SEMESTER GANJIL TAHUN AKADEMIK 2014-2015

Nama Mahasiswa : Syamsul Deska Wijaya
NIM : 11.12.524
Nama Pembimbing : Dr. Eng. I Komang Soma Wirata, ST, MT
Judul Skripsi : Sistem Pendeteksi Untuk Menentukan Lubang Pada PCB Menggunakan Sensor Kamera

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1			Konsultasi permasalahan pengambilan Gbr	
2	6/6 ¹⁵		Konsultasi Bab I, II & III	
3				
4				
5				
6				
7				





Malang,

Dr. Eng. I Komang Soma Wirata, ST, MT
NIP.Y.1030100361

MONITORING BIMBINGAN SKRIPSI

SEMESTER GANJIL TAHUN AKADEMIK 2014-2015

Nama Mahasiswa : Syamsul Deska Wijaya
 NIM : 11.12.524
 Nama Pembimbing : Dr. Eng. Aryuanto Soetedjo, ST, MT
 Judul Skripsi : Sistem Pendeteksi Untuk Menentukan Lubang Pada PCB Menggunakan Sensor Kamera

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1			- Bahasa perwajahan dipada / C	
2			- Batasan dr aspek lubang	
3			- Image dr kamera	
4			- Pengambilan data dr kamera	
5				
6				
7				

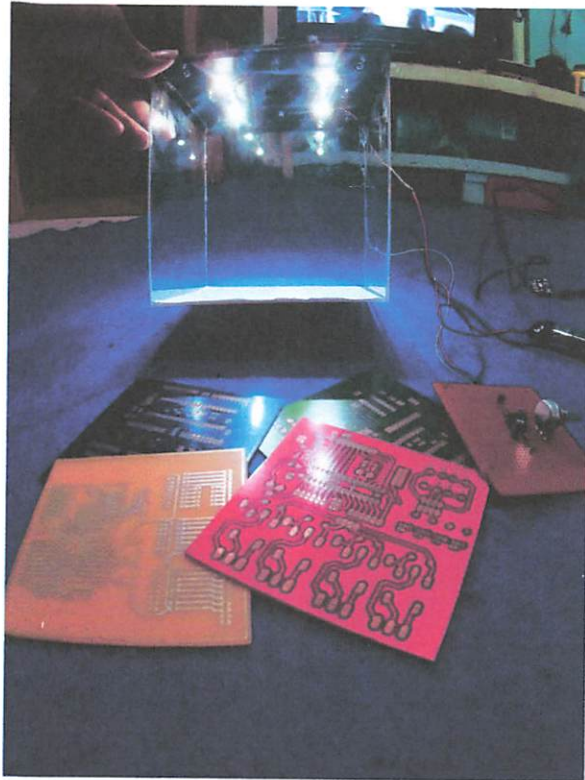
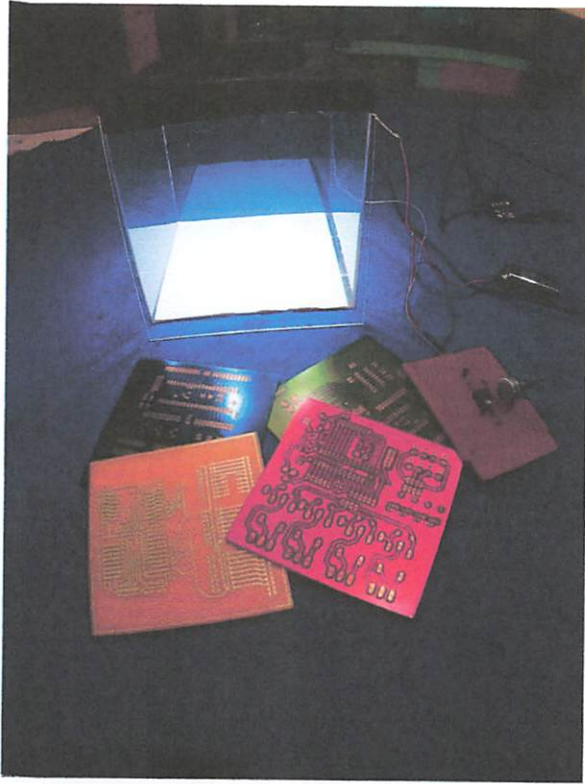
Malang,

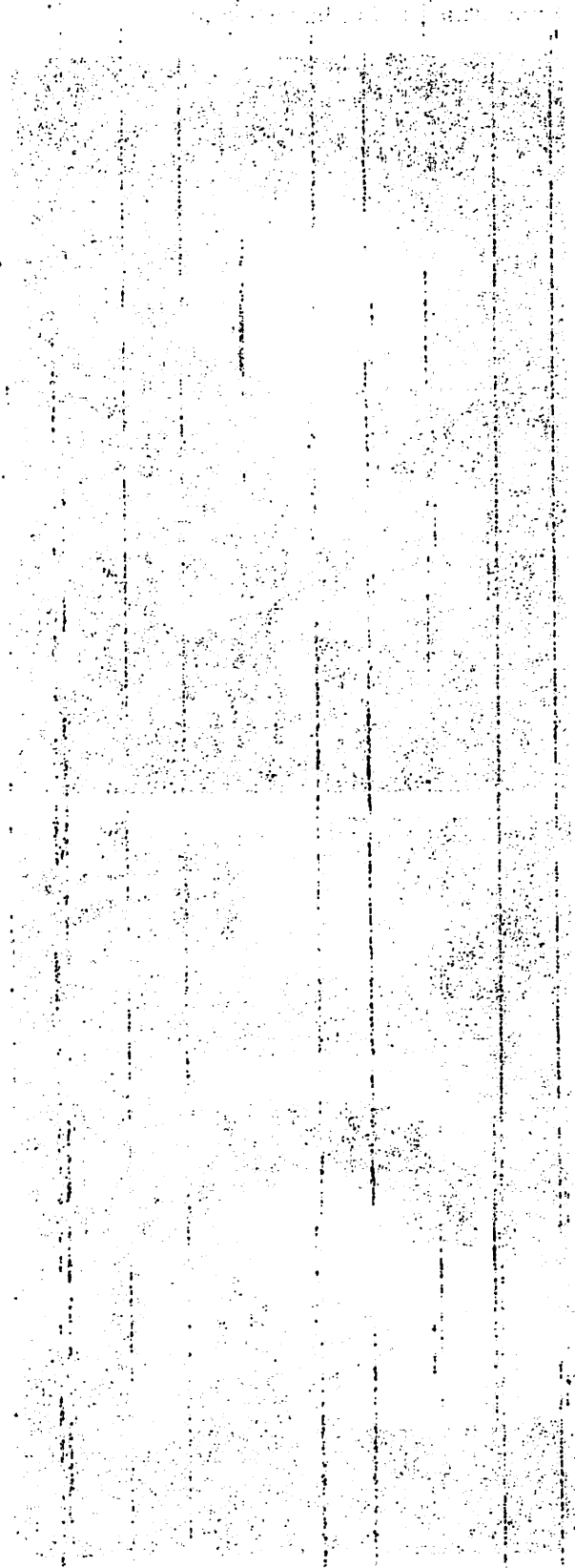
Pembimbing

Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP.Y.1030800417

LAMPIRAN PROGRAM

Lampiran 1 : Gambar Mekanik





Lampiran 2 : Listing Program (unit1.pass)

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, ComCtrls;
type
  TForm1 = class(TForm)
    ProgressBar1: TProgressBar;
    Label1: TLabel;
    Timer1: TTimer;
  procedure Timer1Timer(Sender: TObject);
  private
    angka:Integer;
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
uses Unit2;

{$R *.dfm}

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if(angka<=100) then
  begin
    angka:=angka+1;
  end
  else
```

```
begin
    timer1.Enabled:=false;
    angka:=100;
    Form2.Show;
    //Form1.Visible:=false;
end;
ProgressBar1.Position:=angka;
end;
end.

unit Unit2;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DSPack, ExtDlgs, StdCtrls, ExtCtrls, DirectShow9, DsUtil;
type
    TForm2 = class(TForm)
        Shape1: TShape;
        FilterGraph1: TFilterGraph;
        SampleGrabber1: TSampleGrabber;
        Filter1: TFilter;
        Button3: TButton;
        VideoWindow1: TDSVideoWindowEx2;
        Panel1: TPanel;
        ComboBox1: TComboBox;
        Button1: TButton;
        Image1: TImage;
        Button2: TButton;
        Button4: TButton;
        Button5: TButton;
        Button6: TButton;
        OpenPictureDialog1: TOpenPictureDialog;
```

```

procedure listingDevice;
procedure FormCreate(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure ComboBox1Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button5Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
private
  Path : String;
  { Private declarations }
public
  { Public declarations }
end;
var
  Form2: TForm2;
  CapEnum: TSysDevEnum;
implementation
uses Unit1, Unit3, Unit5, Unit6, Unit8;
{ $R *.dfm }
procedure TForm2.listingDevice;
var i: integer;
begin
  combobox1.Clear;
  CapEnum:=TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
  For i:=0 to CapEnum.CountFilters-1 do
    begin
      combobox1.Items.Add(CapEnum.Filters[i].FriendlyName);
    end;

```



```

end;

procedure TForm2.FormCreate(Sender: TObject);
begin
    listingDevice;

    Path:= ExtractFilePath(Application.ExeName);

    //ShowMessage(Path);
end;

procedure TForm2.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    capenum.Free;

    FilterGraph1.ClearGraph;

    FilterGraph1.Active := false;
end;

procedure TForm2.ComboBox1Click(Sender: TObject);
begin
    FilterGraph1.ClearGraph;

    FilterGraph1.Active := false;

    Filter1.BaseFilter.Moniker :=
capenum.GetMoniker(combobox1.ItemIndex);

    FilterGraph1.Active := true;

    with FilterGraph1 as ICaptureGraphBuilder2 do

        RenderStream(@PIN_CATEGORY_PREVIEW,nil, Filter1 as IBaseFilter, SampleGrabber1
as IBaseFilter, VideoWindow1 as IbaseFilter);

        FilterGraph1.Play;
end;

procedure TForm2.Button1Click(Sender: TObject);
begin
    samplegrabber1.GetBitmap(Image1.Picture.Bitmap);

    Image1.Picture.SaveToFile(Path+'pcb.bmp');

    ShowMessage('Gambar Berhasil Di Capture');
end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin

```

```
//Form2.Close;

end;

procedure TForm2.Button5Click(Sender: TObject);
begin
    Form3.Timer1.Enabled:=true;
    Form3.show;
end;

procedure TForm2.Button4Click(Sender: TObject);
begin
    Form5.Timer1.Enabled:=true;
    Form5.show;
end;

procedure TForm2.Button3Click(Sender: TObject);
begin
    Form6.Timer2.Enabled:=true;
    Form6.show;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
    Form8.Timer2.Enabled:=true;
    Form8.show;
end;

procedure TForm2.Button6Click(Sender: TObject);
begin
    with TOpenPictureDialog.Create(self) do
        try
            Caption := 'Open Image';
            Options := [ofPathMustExist, ofFileMustExist];
            if Execute then
                Image1.Picture.LoadFromFile(FileName);
        finally
            Free;
```

```

    end;

    Image1.Picture.SaveToFile(Path+'pcb.bmp');
end;
end.
var
    Form3: TForm3;
implementation
uses Unit4, Unit2;
{$R *.dfm}
procedure TForm3.Button1Click(Sender: TObject);
var src, dest: TBitmap;
begin
    src := TBitmap.Create;
    dest := TBitmap.Create;
    try
        src.LoadFromFile(Path+'pcb.bmp');
        dest.Width := Round(panjang);
        dest.Height := Round(lebar);
        dest.Canvas.StretchDraw(Rect(0,0,dest.Width,dest.Height),src);
        dest.SaveToFile(Path+'pcb.bmp');
        dest.SaveToFile(Path+'result.bmp');
        ShowMessage('Proses Simpan Gambar Berhasil');
    finally
        src.Free;
        dest.Free;
    end;
    Form2.Button2.Enabled:=false;
    Form2.Button3.Enabled:=true;
    Form2.show;
end;
procedure TForm3.TrackBar1Change(Sender: TObject);
var Zoom : Double;

```

```

    pj,lb: Integer;
begin
    Zoom := TrackBar1.Position;
    panjang := Zoom*64;
    lebar := Zoom*48;
    pj:= Round(panjang);
    lb:= Round(lebar);
    Edit1.Text:=inttostr(pj)+'x'+inttostr(lb);
end;
procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    //Form3.Close;
end;
procedure TForm3.Button2Click(Sender: TObject);
begin
    Form4.Timer1.Enabled:=true;
    Form4.show;
end;
procedure TForm3.Timer1Timer(Sender: TObject);
var Zoom : Double;
    pj,lb: Integer;
begin
    Path:= ExtractFilePath(Application.ExeName);
    Image1.Picture.LoadFromFile(Path+'pcb.bmp');
    Zoom := TrackBar1.Position;
    panjang := Zoom*64;
    lebar := Zoom*48;
    pj:= Round(panjang);
    lb:= Round(lebar);
    Edit1.Text:=inttostr(pj)+'x'+inttostr(lb);
    Timer1.Enabled:=false;
end;

```

```

procedure TForm3.Button3Click(Sender: TObject);
begin
    Form2.Show;
    Form3.Close;
end;
end.
unit Unit4;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls;
type
    TForm4 = class(TForm)
        Image1: TImage;
        Timer1: TTimer;
        procedure Timer1Timer(Sender: TObject);
    private
        Path : String;
        { Private declarations }
    public
        { Public declarations }
    end;
var
    Form4: TForm4;\
implementation
    {$R *.dfm}
    procedure TForm4.Timer1Timer(Sender: TObject);
    var src : TBitmap;
begin
        src := TBitmap.Create;
        Path:= ExtractFilePath(Application.ExeName);
    try

```

```

    src.LoadFromFile(Path+'result.bmp');
    Image1.Picture.Assign(src);
    Image1.Width:=src.Width;
    Image1.Height:=src.Height;
    Form4.Width:=src.Width+18;
    Form4.Height:=src.Height+39;
  finally
    src.Free;
  end;
  Timer1.Enabled:=false;
end;

end.

unit Unit4;

interface

uses

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls;

type

  TForm4 = class(TForm)
    Image1: TImage;
    Timer1: TTimer;
    procedure Timer1Timer(Sender: TObject);
  private
    Path : String;
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation

```

{SR *.dfm}

procedure TForm4.Timer1Timer(Sender: TObject);

var src : TBitmap;

begin

src := TBitmap.Create;

Path:= ExtractFilePath(Application.ExeName);

try

src.LoadFromFile(Path+'result.bmp');

Image1.Picture.Assign(src);

Image1.Width:=src.Width;

Image1.Height:=src.Height;

Form4.Width:=src.Width+18;

Form4.Height:=src.Height+39;

finally

src.Free;

end;

Timer1.Enabled:=false;

end;

end.

unit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ComCtrls, StdCtrls, ExtCtrls;

type

TForm5 = class(TForm)

Image1: TImage;

Image2: TImage;

Panel1: TPanel;

Label1: TLabel;

Label4: TLabel;

Edit1: TEdit;

```

Button1: TButton;
Button2: TButton;
Button3: TButton;
TrackBar1: TTrackBar;
Panel2: TPanel;
Button4: TButton;
Timer1: TTimer;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Button5Click(Sender: TObject);
private
  Path : String;
  { Private declarations }
public
  { Public declarations }
end;
var
  Form5: TForm5;
implementation
uses Unit2;
{ $R *.dfm }
function ConvertBitmapToGrayscale(const Bitmap: TBitmap): TBitmap;
var
  i, j: Integer;
  Grayshade, Red, Green, Blue: Byte;
  PixelColor: Longint;
begin

```



```

with Bitmap do
  for i := 0 to Width - 1 do
    for j := 0 to Height - 1 do
      begin
        PixelColor := ColorToRGB(Canvas.Pixels[i, j]);
        Red      := PixelColor;
        Green    := PixelColor shr 8;
        Blue     := PixelColor shr 16;
        Grayshade := Round(0.3 * Red + 0.6 * Green + 0.1 * Blue);
        Canvas.Pixels[i, j] := RGB(Grayshade, Grayshade, Grayshade);
      end;
    end;
  end;
  Result := Bitmap;

```

```
end;
```

```
function InvertBmp(SourceBmp: TBitmap): TBitMap;
```

```
var
```

```
  i, j: Longint;
```

```
  tmp: TBitMap;
```

```
  red, green, blue: Byte;
```

```
  PixelColor: Longint;
```

```
begin
```

```
  tmp := TBitmap.Create;
```

```
  tmp.Width := SourceBmp.Width;
```

```
  tmp.Height := SourceBmp.Height;
```

```
  for i := 0 to SourceBmp.Width - 1 do
```

```
    begin
```

```
      for j := 0 to SourceBmp.Height - 1 do
```

```
        begin
```

```
          PixelColor := ColorToRGB(SourceBmp.Canvas.Pixels[i, j]);
```

```
          red := PixelColor;
```

```
          green := PixelColor shr 8;
```

```
          blue := PixelColor shr 16;
```

```
          red := 255 - red;
```

```

    green := 255 - green;
    blue := 255 - blue;
    tmp.Canvas.pixels[i, j] := (red shl 8 + green) shl 8 + blue;
end;
end;
Result := tmp;
end;
function BAWBmp(SourceBmp: TBitmap; var eR,Ge,Be : Integer): TBitmap;
var
    i, j,RR,GG,BB: Longint;
    tmp: TBitmap;
    red, green, blue: Byte;
    PixelColor: Longint;
begin
    tmp := TBitmap.Create;
    tmp.Width := SourceBmp.Width;
    tmp.Height := SourceBmp.Height;
    for i := 0 to SourceBmp.Width - 1 do
        begin
            for j := 0 to SourceBmp.Height - 1 do
                begin
                    PixelColor := ColorToRGB(SourceBmp.Canvas.Pixels[i, j]);
                    red := PixelColor;
                    green := PixelColor shr 8;
                    blue := PixelColor shr 16;
                    if(red<eR)then
                        begin
                            RR:=0;
                        end
                    else
                        begin
                            RR:=255;

```

```

    end;
  if(green<Ge)then
    begin
      GG:=0;
    end
  else
    begin
      GG:=255;
    end;
  if(blue<Be) then
    begin
      BB:=0;
    end
  else
    begin
      BB:=255;
    end;
  tmp.Canvas.pixels[i, j] := RGB(RR,GG,BB);
  end;
end;
Result := tmp;
end;
procedure TForm5.FormCreate(Sender: TObject);
begin
  //Path:= ExtractFilePath(Application.ExeName);
  //Image1.Picture.LoadFromFile(Path+'result.bmp');
end;
procedure TForm5.Button1Click(Sender: TObject);
var
  bmp:TBitmap;
  w, h, i, j: Integer;
  colors : TColor;

```

```

begin
  bmp := TBitmap.Create;
  try
    bmp.PixelFormat := pf24bit;
    bmp.Width := Image1.width;
    bmp.Height := Image1.height;
    for h := 0 to Image1.height-1 do
      begin
        for w := 0 to Image1.width-1 do
          begin
            colors:=Image1.Canvas.pixels[w,h];
            bmp.Canvas.Pixels[w,h] := colors;
          end;
        end;
      bmp:=ConvertBitmapToGrayscale(bmp);
      bmp.SaveToFile(Path+'Test.bmp');
      bmp.SaveToFile(Path+'result.bmp');
    finally
      bmp.Free;
    end;
    Image1.Picture.LoadFromFile(Path+'Test.bmp');
    Image1.Stretch:=true;
    Image1.Proportional:=true;
  end;
  procedure TForm5.Button2Click(Sender: TObject);
  var
    bmp:TBitmap;
    w, h, i, j: Integer;
    colors : TColor;
  begin
    bmp := TBitmap.Create;
    try

```

```

    bmp.PixelFormat := pf24bit;
    bmp.Width := Image1.width;
    bmp.Height := Image1.height;
    for h := 0 to Image1.height-1 do
        begin
            for w := 0 to Image1.width-1 do
                begin
                    colors:=Image1.Canvas.pixels[w,h];
                    bmp.Canvas.Pixels[w,h] := colors;
                end;
            end;
        bmp:=InvertBmp(bmp);
        bmp.SaveToFile(Path+'Test.bmp');
        bmp.SaveToFile(Path+'result.bmp');
        //Image1.Picture.Free;
        Image1.Stretch:=true;
        Image1.Picture.LoadFromFile(Path+'Test.bmp');
        Image1.Stretch:=true;
    finally
        bmp.Free;
    end;
end;

```

```

procedure TForm5.Button3Click(Sender: TObject);

```

```

var

```

```

    bmp:TBitmap;

```

```

    w, h, i, j,ergb: Integer;

```

```

    colors : TColor;

```

```

begin

```

```

    ergb:=strtoint(Edit1.text);

```

```

    bmp := TBitmap.Create;

```

```

try

```

```

bmp.PixelFormat := pf24bit;
bmp.Width := Image1.width;
bmp.Height := Image1.height;
for h := 0 to Image1.height-1 do
  begin
    for w := 0 to Image1.width-1 do
      begin
        colors:=Image1.Canvas.pixels[w,h];
        bmp.Canvas.Pixels[w,h] := colors;
      end;
    end;
    bmp:=BAWBmp(bmp,ergb,ergb,ergb);
    bmp.SaveToFile(Path+'result.bmp');
    //Image1.Picture.Free;
    Image1.Stretch:=true;
    Image1.Picture.LoadFromFile(Path+'Test.bmp');
    Image1.Stretch:=true;
    Image2.Stretch:=true;
    Image2.Picture.LoadFromFile(Path+'result.bmp');
    Image2.Stretch:=true;
  finally
    bmp.Free;
  end;
end;

procedure TForm5.Button4Click(Sender: TObject);
begin
  Image2.Picture.SaveToFile(Path+'result.bmp');
  Form2.Button2.Enabled:=true;
  Form2.Button3.Enabled:=false;
  Form2.Show;
  Form5.Close;
end;

```

```

end;
procedure TForm5.TrackBar1Change(Sender: TObject);
begin
    Edit1.Text := inttostr(TrackBar1.Position);
end;
procedure TForm5.Timer1Timer(Sender: TObject);
begin
    Path:= ExtractFilePath(Application.ExeName);
    Image1.Stretch:=true;
    Image1.Picture.LoadFromFile(Path+'pcb.bmp');
    Image1.Stretch:=true;
    Timer1.Enabled:=false;
end;

procedure TForm5.Button5Click(Sender: TObject);
begin
    Form2.Show;
    Form5.Close;
end;
end.
unit Unit6;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, StdCtrls, Grids, DBGrids, DB, ADODB;
type
    TForm6 = class(TForm)
        ADOConnection1: TADOConnection;
        ADOTable1: TADOTable;
        DataSource1: TDataSource;
        Image1: TImage;
        Button1: TButton;

```

```

DBGrid1: TDBGrid;
Timer1: TTimer;
Timer2: TTimer;
Button3: TButton;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
  Path : String;
  { Private declarations }
public
  detik : Integer;
  banyak : Integer;
  EXMIN : array of Integer;
  EXMAX : array of Integer;
  YEMIN : array of Integer;
  YEMAX : array of Integer;
  { Public declarations }
end;
var
  Form6: TForm6;
implementation
uses Unit7;
{ $R *.dfm }
function ConvertBitmapToGrayscale(const Bitmap: TBitmap): TBitmap;
var
  i, j: Integer;
  Grayshade, Red, Green, Blue: Byte;
  PixelColor: Longint;
begin

```



```

with Bitmap do
  for i := 0 to Width - 1 do
    for j := 0 to Height - 1 do
      begin
        PixelColor := ColorToRGB(Canvas.Pixels[i, j]);
        Red      := PixelColor;
        Green    := PixelColor shr 8;
        Blue     := PixelColor shr 16;
        Grayshade := Round(0.3 * Red + 0.6 * Green + 0.1 * Blue);
        Canvas.Pixels[i, j] := RGB(Grayshade, Grayshade, Grayshade);
      end;
    end;
  end;
  Result := Bitmap;
end;

function InvertBmp(SourceBmp: TBitmap): TBitmap;
var
  i, j: Longint;
  tmp: TBitmap;
  red, green, blue: Byte;
  PixelColor: Longint;
begin
  tmp := TBitmap.Create;
  tmp.Width := SourceBmp.Width;
  tmp.Height := SourceBmp.Height;
  for i := 0 to SourceBmp.Width - 1 do
    begin
      for j := 0 to SourceBmp.Height - 1 do
        begin
          PixelColor := ColorToRGB(SourceBmp.Canvas.Pixels[i, j]);
          red := PixelColor;
          green := PixelColor shr 8;
          blue := PixelColor shr 16;
          red := 255 - red;

```

```

    green := 255 - green;
    blue := 255 - blue;
    tmp.Canvas.pixels[i, j] := (red shl 8 + green) shl 8 + blue;
end;
end;
Result := tmp;
end;
function BAWBmp(SourceBmp: TBitmap): TBitMap;
var
    i, j, RR, GG, BB: Longint;
    tmp: TBitMap;
    red, green, blue: Byte;
    PixelColor: Longint;
begin
    tmp := TBitmap.Create;
    tmp.Width := SourceBmp.Width;
    tmp.Height := SourceBmp.Height;
    for i := 0 to SourceBmp.Width - 1 do
        begin
            for j := 0 to SourceBmp.Height - 1 do
                begin
                    PixelColor := ColorToRGB(SourceBmp.Canvas.Pixels[i, j]);
                    red := PixelColor;
                    green := PixelColor shr 8;
                    blue := PixelColor shr 16;
                    if(red<2)then
                        begin
                            RR:=0;
                        end
                    else
                        begin
                            RR:=255;

```

```
    end;
  if(green<2)then
    begin
      GG:=0;
    end
  else
    begin
      GG:=255;
    end;
  if(blue<2) then
    begin
      BB:=0;
    end
  else
    begin
      BB:=255;
    end;
  tmp.Canvas.pixels[i, j] := RGB(RR,GG,BB);
  end;
end;
Result := tmp;
end;
procedure TForm6.FormCreate(Sender: TObject);
begin
  Path:= ExtractFilePath(Application.ExeName);
  Image1.Picture.LoadFromFile(Path+'pcb.bmp');
end;

procedure TForm6.Button1Click(Sender: TObject);
var
  bmp:TBitmap;
  w, h, i, j: Integer;
```

```

colorRGB,colorRT:longint;

colorRGBs : Integer;

colors : TColor;

xko,yko:Integer;

xmin,xmax,ymin,ymax,tot,yt,total,tat,nomor: Integer;

xminb,xmaxb,yminb,ymaxb,sudah:Boolean;

red, green, blue: Byte;

PixelColor: Longint;

begin
  AdoTable1.Close;
  AdoTable1.Open;
  while ADOTable1.recordcount <> 0 do
    ADOTable1.Delete;
  bmp := TBitmap.Create;
  nomor:=0;

  xmin:=0; xmax:=0; ymin:=0; ymax:=0; yt:=0; tot:=0; total:=0;
  tat:=Image1.width*Image1.height;

  xminb:=false; xmaxb:=false; yminb:=false; ymaxb:=false;
  sudah:=false;

  try
    bmp.PixelFormat := pf24bit;
    bmp.Width := Image1.width;
    bmp.Height := Image1.height;
    for h := 0 to Image1.height-1 do
      begin
        for w := 0 to Image1.width-1 do
          begin
            colors:=Image1.Canvas.pixels[w,h];
            bmp.Canvas.Pixels[w,h] := colors;
          end;
        end;
      end;
    bmp:=ConvertBitmapToGrayscale(bmp);
    bmp.SaveToFile(Path+'gray.bmp');
  
```

```
    bmp:=InvertBmp(bmp);  
    //bmp.SaveToFile(Path+'invert.bmp');  
    //bmp:=BAWBmp(bmp);  
    bmp.SaveToFile(Path+'result.bmp');
```

While sudah = false do

begin

nomor:=nomor+1;

xmin:=0; xmax:=0; ymin:=0; ymax:=0; yt:=0; tot:=0; total:=0;

xminb:=false; xmaxb:=false; yminb:=false; ymaxb:=false;

for i := 0 to bmp.Width - 1 do

begin

for j := 0 to bmp.Height - 1 do

begin

PixelColor := ColorToRGB(bmp.Canvas.Pixels[i, j]);

red := PixelColor;

green := PixelColor shr 8;

blue := PixelColor shr 16;

if(red<5) and (green<5) and (blue<5) then

begin

if(xminb=false) then

begin

xmin:=i;

yt:=j;

xminb:=true;

end

else

begin

xminb:=xminb;

xmin:=xmin;

end;

end

else

```

begin
  xminb:=xminb;
  xmin:=xmin;
end;
end;
end;
for i := xmin to bmp.Width - 1 do
begin
  PixelColor := ColorToRGB(bmp.Canvas.Pixels[i,yt]);
  red := PixelColor;
  green := PixelColor shr 8;
  blue := PixelColor shr 16;
  if(red>5) and (green>5) and (blue>5) then
begin
  if(xmaxb=false) then
begin
  xmax:=i-1;
  xmaxb:=true;
end
else
begin
  xmaxb:=xmaxb;
  xmax:=xmax;
end;
end
else
begin
  xmaxb:=xmaxb;
  xmax:=xmax;
end;
end;
end;
for j := yt downto 0 do

```

```
begin
tot:=0;
for i := xmin to xmax do
  begin
    PixelColor := ColorToRGB bmp.Canvas.Pixels[i, j];
    red := PixelColor;
    green := PixelColor shr 8;
    blue := PixelColor shr 16;
    if(red<5) and (green<5) and (blue<5) then
      begin
        tot:=tot+1;
      end
    else
      begin
        tot:=tot;
      end;
    end;
  end;
  if(tot<=0) then
    begin
      if(yminb=false) then
        begin
          ymin:=j-1;
          yminb:=true;
        end
      else
        begin
          yminb:=yminb;
          ymin:=ymin;
        end;
      end;
    end
  else
    begin
```

```

        yminb:=yminb;
        ymin:=ymin;
    end;
end;
for j := yt to bmp.Height - 1 do
    begin
        tot:=0;
        for i := xmin to xmax do
            begin
                PixelColor := ColorToRGB(bmp.Canvas.Pixels[i, j]);
                red := PixelColor;
                green := PixelColor shr 8;
                blue := PixelColor shr 16;
                if(red<5) and (green<5) and (blue<5) then
                    begin
                        tot:=tot+1;
                    end
                else
                    begin
                        tot:=tot;
                    end;
            end;
        end;
        if(tot<=0) then
            begin
                if(ymaxb=false) then
                    begin
                        ymax:=j-1;
                        ymaxb:=true;
                    end
                else
                    begin
                        ymaxb:=ymaxb;
                    end
                end
            end
        end;
    end;
end;

```



```

    end;
    end;
    end;
ADOTable1.Append;
//ADOTable1.FieldByName('ID').AsInteger:=nomor;
ADOTable1.FieldByName('kode').AsInteger:=nomor;
ADOTable1.FieldByName('xmin').AsInteger:=xmin;
ADOTable1.FieldByName('xmax').AsInteger:=xmax;
ADOTable1.FieldByName('ymin').AsInteger:=ymin;
ADOTable1.FieldByName('ymax').AsInteger:=ymax;
xko:=round((xmax-xmin) div 2);
ADOTable1.FieldByName('kox').AsInteger:=xmin+xko;
yko:=round((ymax-ymin) div 2);
ADOTable1.FieldByName('koy').AsInteger:=ymin+yko;
ADOTable1.Post;
////////////////////////////////////
    if(total<=0) then
        begin
            bmp.SaveToFile(Path+'hasil.bmp');
            sudah:=true;
            ShowMessage('Proses Sudah Selesai ');
        end
    else
        begin
            bmp.SaveToFile(Path+'hasil.bmp');
            //bmp.LoadFromFile(Path+'hasil.bmp');
            sudah:=false;
            //ShowMessage('xminimal = '+inttostr(xmin)+'xmaksimal = '+inttostr(xmax)+'yminimal
            = '+inttostr(ymin)+'ymaksimal = '+inttostr(ymax));
        end;
    end;
end;

//ShowMessage('xminimal = '+inttostr(xmin)+'xmaksimal = '+inttostr(xmax)+'yminimal =
'+inttostr(ymin)+'ymaksimal = '+inttostr(ymax));

```

```

finally
    bmp.Free;
    DBGrid1.Refresh;
end;
end;

procedure TForm6.Timer1Timer(Sender: TObject);
var dtk,X0,X1,Y0,Y1 : Integer;
begin
    detik:=detik+1;
    if(detik>banyak) then
        begin
            //ShowMessage('Data Habis Oke');
            Timer1.Enabled:=false;
        end
    else
        begin
            Image1.Canvas.Pen.Color := clRed;
            Image1.Canvas.Pen.Width := 5;
            Image1.Canvas.MoveTo(EXMIN[detik],YEMIN[detik]);
            Image1.Canvas.LineTo(EXMAX[detik],YEMIN[detik]);
            Image1.Canvas.LineTo(EXMAX[detik],YEMAX[detik]);
            Image1.Canvas.LineTo(EXMIN[detik],YEMAX[detik]);
            Image1.Canvas.LineTo(EXMIN[detik],YEMIN[detik]);
            //ShowMessage(inttostr(EXMIN[detik])+'+ '+inttostr(detik)+' ');
        end;
        //
        //ShowMessage(inttostr(detik)+' '+inttostr(banyak)+' ');
        //Timer1.Enabled:=false;
    end;

procedure TForm6.Timer2Timer(Sender: TObject);
begin

```

```

    Path:= ExtractFilePath(Application.ExeName);
    Image1.Picture.LoadFromFile(Path+'pcb.bmp');
    Timer2.Enabled:=false;
end;
procedure TForm6.Button3Click(Sender: TObject);
begin
    Form7.Timer2.Enabled:=true;
    Form7.show;
end;
end.
unit Unit7;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Grids, DBGrids, ExtCtrls, DB, ADODB;
type
    TForm7 = class(TForm)
        Image1: TImage;
        ADOConnection1: TADOConnection;
        ADOTable1: TADOTable;
        DataSource1: TDataSource;
        Timer2: TTimer;
        Timer1: TTimer;
        DBGrid1: TDBGrid;
        Button2: TButton;
        procedure Button2Click(Sender: TObject);
        procedure Timer1Timer(Sender: TObject);
        procedure Timer2Timer(Sender: TObject);
    private
        Path : String;
        { Private declarations }
    public

```

```

detik : Integer;
banyak : Integer;
awal :Integer;
EXMIN : array of Integer;
EXMAX : array of Integer;
YEMIN : array of Integer;
YEMAX : array of Integer;
{ Public declarations }
end;

var
Form7: TForm7;
implementation
{ $R *.dfm }
procedure TForm7.Button2Click(Sender: TObject);
var idy : Integer;
begin
AdoTable1.Close;
Timer1.Enabled:=true;
button2.Enabled:=false;
AdoTable1.open;
banyak := AdoTable1.RecordCount;
SetLength(EXMIN,banyak);
SetLength(EXMAX,banyak);
SetLength(YEMIN,banyak);
SetLength(YEMAX,banyak);
idy := 0;
while not AdoTable1.eof do
begin
EXMIN[idy]:=AdoTable1.Fields[1].asInteger;
EXMAX[idy]:=AdoTable1.Fields[2].asInteger;
YEMIN[idy]:=AdoTable1.Fields[3].asInteger;
YEMAX[idy]:=AdoTable1.Fields[4].asInteger;

```

```

    AdoTable1.next;

    idy := idy + 1;

    //ShowMessage(inttostr(idy)+' '+inttostr(idy)+' ');

end;

detik := 0;

end;

procedure TForm7.Timer1Timer(Sender: TObject);
var dtk,X0,X1,Y0,Y1 : Integer;

begin
    detik:=detik+1;
    if(detik>banyak) then
        begin
            if(awal=0) then
                begin
                    ShowMessage('Proses Simulasi Pendeteksian titik drill PCB Selesai');
                    awal:=1;
                end;
                detik:=0;
                Form7.Timer1.Enabled:=false;
            end
        else
            begin
                Image1.Canvas.Pen.Color := clRed;
                Image1.Canvas.Pen.Width := 5;
                Image1.Canvas.MoveTo(EXMIN[detik],YEMIN[detik]);
                Image1.Canvas.LineTo(EXMAX[detik],YEMIN[detik]);
                Image1.Canvas.LineTo(EXMAX[detik],YEMAX[detik]);
                Image1.Canvas.LineTo(EXMIN[detik],YEMAX[detik]);
                Image1.Canvas.LineTo(EXMIN[detik],YEMIN[detik]);
                //ShowMessage(inttostr(EXMIN[detik])+' ['+inttostr(detik)+'] ');
            end;
        //

```

```

//ShowMessage(inttostr(detik)+' '+inttostr(banyak)+' ');
//Timer1.Enabled:=false;
end;
procedure TForm7.Timer2Timer(Sender: TObject);
begin
    AdoTable1.Close;
    AdoTable1.open;
    awal:=0;
    Path:= ExtractFilePath(Application.ExeName);
    Image1.Picture.LoadFromFile(Path+'pcb.bmp');
    Timer2.Enabled:=false;
end;
end.
unit Unit8;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, ExtCtrls, StdCtrls, Grids, DBGrids;
type
TForm8 = class(TForm)
    Image1: TImage;
    DBGrid1: TDBGrid;
    Button3: TButton;
    Button1: TButton;
    Timer1: TTimer;
    Timer2: TTimer;
    DataSource1: TDataSource;
    ADOTable1: TADOTable;
    ADOConnection1: TADOConnection;
    procedure Button1Click(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure Button3Click(Sender: TObject);

```

```

private
  Path : String;
  { Private declarations }
public
  { Public declarations }
end;
var
  Form8: TForm8;
implementation
uses Unit7;
{$R *.dfm}
procedure TForm8.Button1Click(Sender: TObject);
var
  bmp:TBitmap;
  w, h, i, j: Integer;
  colorRGB,colorRT:longint;
  colorRGBs : Integer;
  colors : TColor;
  xko,yko:Integer;
  xmin,xmax,ymin,ymax,tot,yt,total,tat,nomor: Integer;
  xminb,xmaxb,yminb,ymaxb,sudah:Boolean;
  red, green, blue: Byte;
  PixelColor: Longint;
begin
  AdoTable1.Close;
  AdoTable1.Open;
  while ADOTable1.recordcount < 0 do
    ADOTable1.Delete;
  bmp := TBitmap.Create;
  nomor:=0;
  xmin:=0; xmax:=0; ymin:=0; ymax:=0; yt:=0; tot:=0; total:=0;
  tat:=Image1.width*Image1.height;
  xminb:=false; xmaxb:=false; yminb:=false; ymaxb:=false;

```



```

sudah:=false;
try
  bmp.PixelFormat := pf24bit;
  bmp.Width := Image1.width;
  bmp.Height := Image1.height;
  for h := 0 to Image1.height-1 do
    begin
      for w := 0 to Image1.width-1 do
        begin
          colors:=Image1.Canvas.pixels[w,h];
          bmp.Canvas.Pixels[w,h] := colors;
        end;
      end;
      //bmp:=ConvertBitmapToGrayscale(bmp);
      //bmp:=InvertBmp(bmp);
      //bmp:=BAWBmp(bmp);
      bmp.SaveToFile(Path+'hasil.bmp');

```

While sudah = false do

```

begin
  nomor:=nomor+1;
  xmin:=0; xmax:=0; ymin:=0; ymax:=0; yt:=0; tot:=0; total:=0;
  xminb:=false; xmaxb:=false; yminb:=false; ymaxb:=false;
  for i := 0 to bmp.Width - 1 do
    begin
      for j := 0 to bmp.Height - 1 do
        begin
          PixelColor := ColorToRGB(bmp.Canvas.Pixels[i, j]);
          red := PixelColor;
          green := PixelColor shr 8;
          blue := PixelColor shr 16;
          if(red<5) and (green<5) and (blue<5) then
            begin

```

```

    if(xminb=false) then
        begin
            xmin:=i;
            yt:=j;
            xminb:=true;
        end
    else
        begin
            xminb:=xminb;
            xmin:=xmin;
        end;
    end
else
    begin
        xminb:=xminb;
        xmin:=xmin;
    end;
end;

end;

end;

for i := xmin to bmp.Width - 1 do
    begin
        PixelColor := ColorToRGB(bmp.Canvas.Pixels[i,yt]);
        red := PixelColor;
        green := PixelColor shr 8;
        blue := PixelColor shr 16;
        if(red>5) and (green>5) and (blue>5) then
            begin
                if(xmaxb=false) then
                    begin
                        xmax:=i-1;
                        xmaxb:=true;
                    end
            end
        end
    end
end

```

```

        else
            begin
                xmaxb:=xmaxb;
                xmax:=xmax;
            end;
        end
    else
        begin
            xmaxb:=xmaxb;
            xmax:=xmax;
        end;
    end;
end;
for j := yt downto 0 do
    begin
        tot:=0;
        for i := xmin to xmax do
            begin
                PixelColor := ColorToRGB bmp.Canvas.Pixels[i, j];
                red := PixelColor;
                green := PixelColor shr 8;
                blue := PixelColor shr 16;
                if(red<5) and (green<5) and (blue<5) then
                    begin
                        tot:=tot+1;
                    end
                else
                    begin
                        tot:=tot;
                    end;
            end;
        end;
    end;
    if(tot<=0) then
        begin

```

```

    if(yminb=false) then
        begin
            ymin:=j-1;
            yminb:=true;
        end
    else
        begin
            yminb:=yminb;
            ymin:=ymin;
        end;
    end
else
    begin
        yminb:=yminb;
        ymin:=ymin;
    end;
end;

for j := yt to bmp.Height - 1 do
    begin
        tot:=0;
        for i := xmin to xmax do
            begin
                PixelColor := ColorToRGB(bmp.Canvas.Pixels[i, j]);
                red := PixelColor;
                green := PixelColor shr 8;
                blue := PixelColor shr 16;
                if(red<5) and (green<5) and (blue<5) then
                    begin
                        tot:=tot+1;
                    end
                else
                    begin

```

```

        tot:=tot;
    end;
end;
if(tot<=0) then
    begin
        if(ymaxb=false) then
            begin
                ymax:=j-1;
                ymaxb:=true;
            end
        else
            begin
                ymaxb:=ymaxb;
                ymax:=ymax;
            end;
        end
    else
        begin
            ymaxb:=ymaxb;
            ymax:=ymax;
        end;
    end;
end;
for i := xmin to xmax+1 do
    begin
        for j := ymin to ymax+1 do
            begin
                bmp.Canvas.pixels[i, j] := RGB(255,255,255);
            end;
        end;
    end;
total:=0;
for i := 0 to bmp.Width - 1 do
    begin

```

```

for j := 0 to bmp.Height - 1 do
begin
    PixelColor := ColorToRGB(bmp.Canvas.Pixels[i, j]);
    red := PixelColor;
    green := PixelColor shr 8;
    blue := PixelColor shr 16;
    if(red<5) and (green<5) and (blue<5) then
        begin
            total:=total+1;
        end
    else
        begin
            total:=total;
        end;
    end;
end;

ADOTable1.Append;
//ADOTable1.FieldByName('ID').AsInteger:=nomor;
ADOTable1.FieldByName('kode').AsInteger:=nomor;
ADOTable1.FieldByName('xmin').AsInteger:=xmin;
ADOTable1.FieldByName('xmax').AsInteger:=xmax;
ADOTable1.FieldByName('ymin').AsInteger:=ymin;
ADOTable1.FieldByName('ymax').AsInteger:=ymax;
xko:=round((xmax-xmin) div 2);
ADOTable1.FieldByName('kox').AsInteger:=xmin+xko;
yko:=round((ymax-ymin) div 2);
ADOTable1.FieldByName('koy').AsInteger:=ymin+yko;
ADOTable1.Post;

if(total<=0) then
begin
    bmp.SaveToFile(Path+'hasil.bmp');
    sudah:=true;
end;

```

```

        ShowMessage('Proses Sudah Selesai ');
    end
else
    begin
        bmp.SaveToFile(Path+'hasil.bmp');
        //bmp.LoadFromFile(Path+'hasil.bmp');
        sudah:=false;
        //ShowMessage('xminimal = '+inttostr(xmin)+'xmaksimal = '+inttostr(xmax)+'yminimal
= '+inttostr(ymin)+'ymaksimal = '+inttostr(ymax));
    end;
end;

//ShowMessage('xminimal = '+inttostr(xmin)+'xmaksimal = '+inttostr(xmax)+'yminimal =
'+inttostr(ymin)+'ymaksimal = '+inttostr(ymax));

finally
    bmp.Free;
    DBGrid1.Refresh;
end;
end;
procedure TForm8.Timer2Timer(Sender: TObject);
begin
    Path:= ExtractFilePath(Application.ExeName);
    Image1.Picture.LoadFromFile(Path+'result.bmp');
    Timer2.Enabled:=false;
end;
procedure TForm8.Button3Click(Sender: TObject);
begin
    Form7.Timer2.Enabled:=true;
    Form7.show;
end;
end.

```

Logitech®

HD Webcam C270

HD video calling that's simple.

- HD 720p video calling on most major IM applications and Logitech Vid™ HD
- 3-megapixel pics (software enhanced)
- Logitech® RightLight™ technology
- Logitech® RightSound™ technology



Logitech® HD Webcam C270. HD video calling that's simple. You'll enjoy an HD 720p video call on most major instant messaging applications and Logitech Vid™ HD—the free, fast and easy way to see your loved ones face to face. Send your family and friends a picture when you don't have time to talk or record a video. Take a 3MP photo (software enhanced) with just one click of a button. Your webcam will adjust to poor lighting conditions automatically to produce the best possible image—thanks to Logitech RightLight™ technology. And background noise doesn't have to spoil your video calls. Logitech RightSound™ technology delivers clear conversations—so you can be heard.



CONFIGURATION MINIMALE REQUISE

Windows® XP (SP2 or higher), Windows Vista® or Windows® 7 (32-bit or 64-bit)

Basic requirements:

- 1 GHz
- 512 MB RAM or more
- 200 MB hard drive space
- Internet connection
- USB 1.1 port (2.0 recommended)

For HD 720p video calling and recording:

- 2.4 GHz Intel® Core™2 Duo
- 2 GB RAM
- 200 MB hard drive space
- USB 2.0 port
- 1 Mbps upload speed or higher
- 1280 x 720 screen resolution



Logitech Vid™



Logitech® HD Webcam C270



TECHNICAL SPECIFICATIONS

- HD video calling (1280 x 720 pixels) with recommended system
- Video capture: Up to 1280 x 720 pixels
- Photos: Up to 3.0 megapixels (software enhanced)
- Built-in mic with Logitech RightSound™ technology
- Hi-Speed USB 2.0 certified (recommended)
- Universal clip fits laptops, LCD or CRT monitors

Logitech webcam software:

- Logitech Vid™ HD
- Logitech RightLight™ technology
- Video and photo capture

Works with most instant messaging applications

PACKAGE SPECIFICATIONS

	Primary Pack	Master Shipper Carton
Part # WER	960-000582	n/a
Bar code	5099206023758 (EAN-13)	50992060237517 (SCC-14)
Part # CENTRAL	960-000635	n/a
Bar code	5099206023802 (EAN-13)	50992060238019 (SCC-14)
Part # EER	960-000636	n/a
Bar code	5099206023819 (EAN-13)	50992060238118 (SCC-14)
Weight	226.8g	2.147 kg
Length	20.95cm	31.50cm
Width	7.62 cm	31.06cm
Height/depth	15.24 cm	22.40 cm
Volume	2.433dm ³	0.02230m ³
1 primary pack	1	n/a
1 intermediate pack	0	n/a
1 master shipper carton	8	1
1 pallet EURO	432	54
1 container 20 ft	10080	1260
1 container 40 ft	21280	2660
1 container 40 ft HQ	23408	2926

PACKAGE CONTENT

- Webcam with 5-foot cable
- Headset
- Logitech webcam software with Logitech Vid™ HD
- User documentation

Logitech Vid™

Logitech® Vid™. The free, fast, and easy way to make video calls with your Logitech webcam.

www.logitech.com

©2010 Logitech. Logitech, the Logitech logo and other Logitech marks are owned by Logitech and may be registered. All other trademarks are the property of their respective owners.

BIOGRAFI PENULIS



Penulis lahir di Ampenan pada tanggal 06 Desember 1992 dari ayah Agus Jaya dan ibu Nur Zulaikah. Penulis merupakan putra Pertama dari 2 bersaudara. Penulis memulai pendidikan pada tahun 1999 di SDN 50 Cakranegara Kecamatan Sandubaya Mataram Lombok NTB. Pertengahan tahun 2005 penulis menempuh pendidikan di SMP Negeri 6 Mataram Lombok NTB sampai tahun 2008. Kemudian melanjutkan pendidikan di SMK Negeri 3 Mataram Lombok NTB mulai tahun 2008 dan lulus tahun 2011, kemudian pada tahun yang sama penulis melanjutkan studi di perguruan tinggi Institut Teknologi Nasional Malang. Penulis memilih Program Studi Teknik Elektro S-1, Konsentrasi Teknik Komputer, Fakultas Teknologi Industri dan diwisuda tanggal 26 September 2015, dengan judul Skripsi “SISTEM PENDETEKSI UNTUK MENENTUKAN LUBANG PADA PCB MENGGUNAKAN SENSOR KAMERA”.