

**PENERAPAN METODE *TEMPLATE MATCHING* DALAM
MENGANALISA CACAT PADA KEPING PCB**

TUGAS AKHIR

*Diajukan kepada Tim Penguji Tugas Akhir Jurusan Teknik Elektronika sebagai
salah satu persyaratan Guna memperoleh Gelar Sarjana Pendidikan*



Oleh:

**MUSTIKA SONSANK
NIM. 18657/2010**

**PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2015**

PERSETUJUAN PEMBIMBING

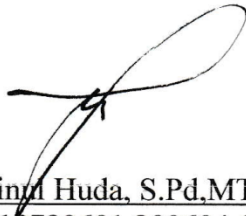
PENERAPAN METODE *TEMPLATE MATCHING* DALAM MENGANALISA CACAT PADA KEPING PCB

Nama : Mustika Sonsank
NIM/TM : 18657/2010
Program Studi : Pendidikan Teknik Elektronika
Jurusan : Teknik Elektronika
Fakultas : Teknik

Padang, April 2015

Disetujui Oleh

Pembimbing I,



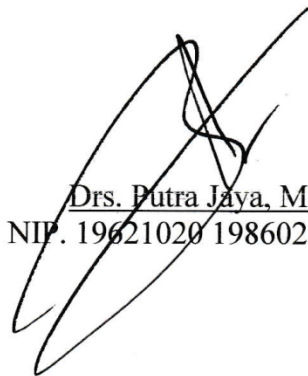
Yasding Huda, S.Pd, MT
NIP. 19790601 200604 1 026

Pembimbing II,



Khairi Budayawan, S.Pd, M.Sc
NIP. 19760801 200312 1 002

Mengetahui,
Ketua Jurusan Teknik Elektronika



Drs. Putra Jaya, MT
NIP. 19621020 198602 1 001

PENGESAHAN

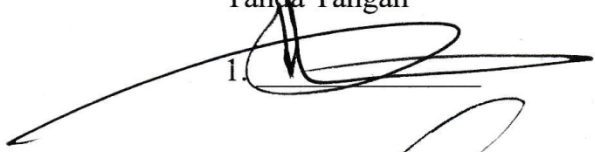
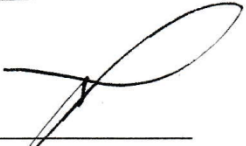
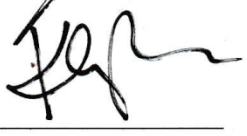
**Dinyatakan lulus setelah dipertahankan di depan Tim Penguji Tugas Akhir
Program Studi Pendidikan Teknik Elektronika Jurusan Teknik Elektronika
Fakultas Teknik Universitas Negeri Padang**

**Judul : Penerapan Metode Template Matching Dalam
Menganalisa Cacat Pada Keping PCB**

Nama : Mustika Sonsank
NIM/TM : 18657/2010
Program Studi : Pendidikan Teknik Elektronika
Jurusan : Teknik Elektronika
Fakultas : Teknik

Padang, April 2015

Tim Penguji

	Nama	Tanda Tangan
1. Ketua	: Drs. Almasri, MT	1. 
2. Sekretaris	: Yasdinul Huda, S.Pd, MT	2. 
3. Anggota	: Khairi Budayawan, S.Pd, M.Sc	3. 
4. Anggota	: Muhammad Anwar, S.Pd, MT	4. 

ABSTRAK

Mustika Sonsank : Penerapan Metode Template Matching Dalam Menganalisa Cacat Pada Keping PCB

Perkembangan teknologi saat ini sangat berarti bagi kehidupan manusia. Teknologi saat ini menjadi salah satu kebutuhan dalam kehidupan sehari-hari. Pemanfaatan teknologi terbukti dapat mempermudah kinerja manusia. Hal inilah yang menyebabkan teknologi diterapkan dalam beragam bidang, tidak terkecuali dalam bidang elektronika. Bidang elektronika tidak lepas dari komponen – komponen dan proses pengerjaan suatu rangkaian elektronika. Proses pengerjaan suatu rangkaian elektronika terutama pembuatan jalur PCB yang dikerjakan secara manual yaitu menggunakan tenaga manusia memungkinkan terjadinya *Human Error*, seperti kecacatan dalam rangkaian/jalur PCB. *Human Error* sering terjadi dalam pembuatan jalur PCB, namun tidak ada cara cepat untuk mengetahui apakah jalur PCB sudah benar atau masih ada cacatnya. Cacatan yang dimaksud adalah putus jalur atau pelebaran/penyempitan jalur tembaga yang mengakibatkan adanya jalur terbuka, jalur hubung singkat sehingga membuat PCB tidak berfungsi. Pengidentifikasian cacat pada keping PCB bisa menggunakan Ohm Meter, namun memakan waktu yang lama, pengidentifikasian menggunakan tenaga manusia pun kurang efektif karena jalur sirkuit PCB sangat kecil. Mengatasi masalah diatas, dimanfaatkan sebuah metode yaitu *Template Matching*, untuk mengetahui letak cacat pada keping PCB, yang menggunakan teknik pengolahan citra dan dibantu bahasa pemrograman Delphi. Dengan teknik pengolahan citra, metode *Template Matching* dan bahasa pemrograman delphi diharapkan dapat menggambarkan letak cacat pada PCB dengan cepat dan tepat.

Kata Kunci : PCB, Metode *Template Matching*, Pengolahan Citra, Delphi

KATA PENGANTAR

Puji syukur Alhamdulillah penulis ucapkan kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, serta dengan izin-Nya penulis dapat menyusun dan menyelesaikan tugas akhir dengan judul “Penerapan Metode *Template Matching* Dalam Menganalisa Cacat Pada Keping PCB”.

Shalawat dan salam tidak lupa pula dicurahkan kepada Nabi Besar Muhammad SAW, sebagai sosok panutan yang telah membawa umat manusia dari zaman jaliah dan kebodohan menuju ke zaman islamiyah serta penuh ilmu pengetahuan.

Penulisan laporan ini dimaksudkan untuk memenuhi salah satu syarat untuk memperoleh gelar Sarjana Pendidikan Strata Satu pada Program Studi Pendidikan Teknik Elektronika Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.

Penulis menyadari banyaknya kekeliruan yang terjadi sehingga tidak sedikit bantuan dan bimbingan yang didapatkan dari berbagai pihak dalam menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Bapak Drs. Syahril, ST, MSCE, Ph.D selaku Dekan Fakultas Teknik Universitas Negeri Padang.
2. Bapak Drs. Putra Jaya, MT selaku Ketua Jurusan Teknik Elektronika dan Penasehat Akademik.
3. Bapak Yasdinul Huda, MT selaku Sekretaris Jurusan Teknik Elektronika dan Pembimbing 1 Tugas Akhir.
4. Bapak Khairi Budayawan, S.Pd, M.Sc selaku Pembimbing II Tugas Akhir.
5. Bapak Drs. Almasri, MT selaku Penguji Tugas Akhir.
6. Bapak Muhammad Anwar, S.Pd, MT selaku Penguji Tugas Akhir.

Semoga semua kebaikan, bimbingan dan kerja sama yang diberikan akan menjadi amal jariyah dan mendapat balasan yang setimpal dari Allah SWT.

Tugas Akhir masih jauh dari sempurna, untuk itu diharapkan kritikan dan saran yang membangun dari para pembaca. Terima kasih.

Padang, Januari 2015

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
SURAT PERNYATAAN	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	4
C. Batasan Masalah	4
D. Rumusan Masalah	5
E. Tujuan	5
F. Manfaat	5
BAB II KAJIAN PUSTAKA	7
A. Printed Circuit Board	7
1. Pengertian PCB	7
2. Cacat pada Keping PCB	9
B. Pengolahan Citra Digital	11
1. Pengertian Citra Digital	11
2. Piksel	13
3. Proses Akuisisi Citra	14
4. Elemen-Elemen Citra Digital	15
5. Elemen sistem Pemrosesan Citra Digital	17
6. Jenis-Jenis Citra Digital	19

7. Format File Citra	21
8. Peningkatan Kualitas Citra	24
C. Metode <i>Template Matching</i>	25
1. Pengertian Metode <i>Template Matching</i>	25
2. Metode <i>Template Matching</i> untuk PCB	28
3. Algoritma Identifikasi Cacat dengan Menggunakan Metode <i>Template Matching</i>	33
D. Flowchart	34
1. Pengertian Flowchart	34
2. Simbol-Simbol Flowchart	37
E. Pemograman Delphi	40
1. Pemograman Delphi 7	40
2. Kebutuhan Sistem minimum	44
F. Penelitian yang Relevan	45
 BAB III METODOLOGI PERANCANGAN	46
A. Metode Uji Coba	46
B. Analisa Kebutuhan Hardware dan Software.....	47
1. Analisa Kebutuhan Hardware	47
2. Analisa Kebutuhan Software	48
C. Flowchart	48
D. Perancangan Program Uji COba.....	51
1. Perancangan Splash	51
2. Perancangan Menu Utama	52
3. Perancangan Proses Prapengolahan	52
 BAB IV IMPLEMENTASI DAN PEMBAHASAN	55
A. Implementasi	55
1. Implementasi Software	55
2. Implementasi Hardware	55
B. Tampilan Program	56

1. Tampilan Splash Program Uji Coba	56
2. Tampilan Menu Utama	57
3. Pembacaan Citra PCB Acuan	62
4. Pengubahan Citra Warna Menjadi Citra Aras Keabuan	65
5. Penapisan Citra (<i>Filter</i>).....	68
6. Pengambangan (<i>Thershold</i>)	71
7. <i>Template Matching</i>	73
8. Deteksi	78
C. Pembahasan	81
1. Pembacaan Citra	81
2. Pengubahan Citra Asli Menjadi Citra Aras Keabuan	81
3. Penapisan Citra	82
4. Pengambangan	83
5. Deteksi	83
BAB V PENUTUP	85
A. Kesimpulan	85
B. Saran	85

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar	Halaman
1. Keping PCB yang Jalur Membentuk Pulau	10
2. Keping PCB Bagus Tanpa Cacat	10
3. Proses Pencuplikan	12
4. Matriks Citra Digital	12
5. Posisi Piksel dalam Sistem Koordinat Citra Berukuran M x N	13
6. Proses Akuisisi Citra Digital	14
7. Elemen Pemrosesan Citra	18
8. Diagram Algoritma <i>Template Matching</i>	28
9. Penapisan dengan Tapis Median pada Matriks 5 x 5	32
10. Contoh Sistem Flowchart.....	35
11. Contoh Conceptual Flowchart	36
12. Contoh Detail Flowchart.....	37
13. Component Palette Delphi	42
14. Object Treeview Delphi	43
15. Object Inspector Delphi	43
16. Blok Diagram Identifikasi Cacat Pada PCB	47
17. Flowchart Program	49
18. Rancangan Splash	51
19. Rancangan Tampilan Utama	52

20. Rancangan Interface Program Simulasi	56
21. Tampilan Splash Program Uji Coba	56
22. Tampilan Program Utama	57
23. Tampilan <i>Form</i> Petunjuk	58
24. Tampilan Menu Prapengolahan	59
25. Tampilan Bingkai Utama Citra PCB Masukan	61
26. Citra Awal PCB Acuan	62
27. PCB Acuan Ditampilkan Dalam Bingkai Tampilan pada Program	63
28. Citra Awal PCB Acuan	63
29. PCB Acuan Ditampilkan Dalam Bingkai Tampilan pada Program.....	64
30. Citra PCB RGB Menjadi Aras Keabuan	66
31. Citra PCB Aras Keabuan Dalam Bingkai Tampilan pada Program	67
32. Citra PCB RGB Menjadi Aras Keabuan	67
33. Citra PCB Aras Keabuan Dalam Bingkai Tampilan pada Program	68
34. Citra PCB melewati Penapisan (<i>Filter</i>)	69
35. Citra PCB melewati Penapisan (<i>Filter</i>)	69
36. Citra PCB Acuan Setelah Dilakukan Pengambangan.....	72
37. Citra PCB Acuan Setelah Dilakukan Pengambangan.....	72
38. Tampilan Bingkai Citra Masukan	74
39. Citra PCB di Bingkai Citra Masukan.....	75
40. Citra PCB di Bingkai Citra Masukan	75
41. PCB Terdapat Jalur Putus	79
42. PCB Terdapat Jalur Putus	79

43. Kondisi PCB Tanpa Cacat	80
44. Kondisi PCB Tanpa Cacat	80

DAFTAR TABEL

Tabel	Halaman
1. Format Citra File Bitmap	21
2. <i>Flow Direction Symbols</i>	38
3. <i>Processing Symbols</i>	38
4. <i>Input / Output Symbols</i>	39

BAB I PENDAHULUAN

A. Latar Belakang Masalah

Perkembangan teknologi sangat berarti bagi kehidupan manusia. Teknologi saat ini menjadi salah satu kebutuhan dalam kehidupan sehari-hari. Pemanfaatan teknologi terbukti dapat mempermudah kinerja manusia. Hal inilah yang menyebabkan teknologi diterapkan dalam beragam bidang, tidak terkecuali dalam bidang elektronika.

Bidang elektronika tidak lepas dari komponen-komponen yang digunakan untuk membuat rangkaian elektronika. Salah satu komponen yang sangat berpengaruh adalah papan rangkaian tercetak (*Printed Circuit Board, PCB*). Menurut Sugianto (2007) “PCB (*Printed Circuit Board*) adalah papan rangkaian yang digunakan sebagai tempat penghubung jalur konduktor dan penyusunan letak komponen-komponen elektronika”.

Proses pengerjaan suatu rangkaian elektronika terutama pembuatan jalur pada PCB yang dikerjakan secara manual yaitu menggunakan tenaga manusia memungkinkan terjadinya *Human Error*, seperti kecacatan dalam rangkaian/jalur PCB. Berfungsi atau tidaknya rangkaian elektronika tergantung pada jalur PCB yang dirancang.

Human Error sering terjadi dalam pembuatan jalur pada PCB, namun tidak ada cara cepat untuk mengetahui apakah jalur PCB sudah benar atau masih ada cacatnya. Cacat yang dimaksud adalah putusya jalur, atau

pelebaran/ penyempitan jalur tembaga yang mengakibatkan adanya jalur terbuka, jalur hubung singkat sehingga membuat PCB tidak berfungsi.

PCB cacat tidak dapat diperbaiki untuk digunakan ulang, PCB akan terbuang sehingga menghabiskan biaya dan banyak waktu saat pengerjaannya. Oleh karena itu pengidentifikasian cacat pada keping/ jalur PCB sangat penting.

Pengidentifikasian cacat pada keping PCB menggunakan beberapa cara salah satunya menggunakan Ohm Meter. Ohm Meter berfungsi untuk mengetahui besaran hambatan atau besaran tahanan pada suatu beban dan juga berfungsi untuk mengetahui ada atau tidak adanya jalur PCB yang putus.

Pengidentifikasian menggunakan Ohm meter memiliki kekurangan yaitu membutuhkan waktu yang lama. Pengidentifikasian dengan mengandalkan tenaga manusia untuk meneliti satu persatu jalur sirkuit PCB apakah terdapat kecacatan tidak efisien, hal ini disebabkan jalur sirkuit yang sangat kecil dan rapat menyulitkan mata manusia untuk meneliti jalur, sehingga menyebabkan tingkat kelelahan yang tinggi dan membutuhkan waktu lama.

Cara lain yang digunakan dalam mengidentifikasi cacat pada keping PCB adalah pengolahan citra digital. Pengolahan citra merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra yang banyak melibatkan persepsi visual dengan bantuan komputer. Menurut Rinaldi Munir (2004 : 3) “Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau mesin”.

Pengolahan citra dapat mengidentifikasi jalur PCB yang putus, dimana oleh mata manusia, jalur tersebut tidak terlihat, namun oleh pengolahan citra akan teridentifikasi. Sistem pengolahan citra digital sangat luas, salah satunya metode *Template Matching*.

Menurut Achmad Hidayatno (2005 :2) “Metode *Template Matching* adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi template (acuan)”. Metode *Template Matching* adalah salah satu metode yang digunakan untuk menjelaskan bagaimana otak manusia mengenali kembali bentuk-bentuk atau pola-pola.

Tuntutan praktek kuliah mengharuskan mahasiswa bekerja cepat, efektif dan efisien baik dalam manajemen waktu maupun penggunaan biaya. Pengidentifikasian cacat pada keping PCB menggunakan pengolahan citra lebih cepat dan memberi hasil yang akurat di banding menggunakan Ohm meter atau pengidentifikasian secara manual. Oleh karena itu perlu dibuat tugas akhir dengan judul “**Penerapan Metode Template Matching dalam Menganalisa Cacat pada Keping PCB.**”

B. Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan diidentifikasi masalah sebagai berikut :

1. PCB yang pembuatan jalur dikerjakan secara manual oleh manusia memiliki kemungkinan adanya *Human Error* seperti, jalur putus, penyempitan/ pelebaran jalur tembaga.
2. PCB yang cacat jalurnya tidak dapat digunakan, sehingga membuang waktu dan pemborosan.
3. Pengidentifikasian jalur PCB menggunakan Ohm Meter tidak efektif, karena akan membutuhkan waktu yang lama.
4. Pengidentifikasian jalur PCB secara manual mengandalkan mata manusia untuk meneliti kecacatan jalur satu-persatu kurang efisien karena membutuhkan waktu lama dan tingkat kelelahan yang tinggi.

C. Batasan Masalah

Mengingat ruang lingkup permasalahan yang luas dan keterbatasan yang ada, maka permasalahan dibatasi pada:

1. Program uji coba dirancang menggunakan pengolahan citra digital dan menggunakan software untuk mengidentifikasi cacat pada PCB, yang membutuhkan waktu lebih cepat dan akurat.
2. Perancangan dan pembuatan program uji coba akan mendeteksi dan menampilkan cacat pada keping PCB dengan penerapan metode *Template Matching*.

D. Rumusan Masalah

Berdasarkan batasan masalah yang telah dikemukakan dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana membuat dan menerapkan software / bahasa pemrograman untuk mengidentifikasi cacat pada keping PCB?
2. Bagaimana menerapkan metode *Template Matching* agar mendapatkan kecocokan citra acuan dengan citra masukan PCB?

E. Tujuan Tugas Akhir

Sesuai dengan latar belakang, batasan masalah serta rumusan masalah, maka penelitian ini bertujuan untuk :

1. Membuat program uji coba yang mampu mendeteksi adanya cacat pada keping PCB secara otomatis dengan menggunakan pendekatan metode *Template Matching*.
2. Mengurangi resiko tidak berfungsinya PCB karena terdapat cacat pada jalur-jalur PCB.

F. Manfaat Tugas Akhir

Adapun manfaat tugas akhir adalah :

1. Menghemat waktu dan biaya dalam proses pengerjaan rangkaian elektronika.
2. Mempersingkat waktu untuk mendeteksi kecacatan pada keping PCB dengan pengolahan citra dan metode *Template Matching*.
3. Hasil penelitian dapat digunakan sebagai bahan referensi perkuliahan citra digital.

4. Sebagai bahan referensi dan perbandingan dalam melakukan penelitian di masa yang akan datang.

BAB II

KAJIAN PUSTAKA

A. Printed Circuit Board

1. Pengertian PCB

Papan rangkaian tercetak (Printed Circuit Board, PCB) dikutip dari Hasimunikom (2012) adalah “Sebuah papan yang penuh dengan jalur sirkuit dari logam yang menghubungkan komponen elektronik satu sama lain tanpa kabel”. Menurut Sugianto (2007) “PCB adalah papan rangkaian yang digunakan sebagai tempat penghubung jalur konduktor dan penyusunan letak komponen-komponen elektronika”.

Pada tahun 1936 papan sirkuit cetak pertama di temukan *Paul Eisler* ilmuan Austria yang memasukkan penggunaan papan sirkuit ini dalam sebuah radio. Setelah tahun 1950 papan sirkuit cetak telah digunakan secara masal di industri elektronik.

Industri elektronik PCB berkembang pesat sehingga ada bermacam-macam tipe PCB dilihat dari berbagai sudut pandang.

Menurut Adhita Wisnu Wardhana (2008 :1)

PCB dilihat dari susunan lapis digolongkan menjadi 3 jenis yaitu, lapis tunggal, lapis ganda dan multi lapis. Dilihat dari bahan baku pembuatan PCB digolongkan menjadi 2, yaitu PCB keras dan PCB lunak. Sedangkan dari teknologi pengelupasan lapisan tembaga, ada dua jenis, yaitu pengelupasan mesin dan pengelupasan dengan larutan kimia.

PCB berfungsi untuk memasang komponen elektronik, serta membuat rapi rakitan rangkaian elektronika. PCB juga berfungsi meminimalisir pemakaian tempat, dan ruang rangkaian elektronika.

Macam-macam bentuk PCB dikutip dari Chogwang (2014) adalah :

a. PCB Matrix Strip Board.

Merupakan jenis PCB yang bentuknya terdiri atas lubang-lubang. Kekurangan PCB ini adalah sulitnya mengatur pengkabelan yang menghubungkan antara komponen satu dengan komponen lain sehingga menyebabkan kabel-kabel yang dihubungkan saling menyilang.

Kesulitan lain adalah saat penyolderan kaki-kaki komponen dengan 2 kabel penghubung atau lebih.

b. PCB Cooper Clad.

PCB jenis Cooper Clad merupakan PCB yang terbuat dari bahan ebonite atau fiber glass yang salah satu atau kedua sisinya dilapisi oleh lapisan tembaga. PCB jenis Cooper Clad dapat dibagi menjadi beberapa bagian, yaitu :

1) PCB Satu Sisi (*Single Side*)

Yaitu papan PCB yang hanya memiliki satu sisi yang dilapisi oleh lempeng tembaga.

2) PCB Dua Sisi (*Double Side*)

Yaitu papan PCB yang mempunyai dua sisi yang dilapisi oleh lempeng tembaga dan lapisan fibernya ada diantara dua lapisan tembaga tersebut.

3) PCB Banyak Sisi (*Multi Layer*).

Yaitu layer yang biasanya hanya dibuat oleh pabrik pembuat peralatan tersebut. Type multi layer ini terdiri dari beberapa lapis tembaga dan fiber yang disusun secara berselingan.

PCB yang akan diidentifikasi dalam tugas akhir adalah PCB dengan jenis Cooper Clad, yang mempunyai satu sisi berbahan keras dan pengelupasannya menggunakan larutan kimia.

2. Cacat pada Keping PCB

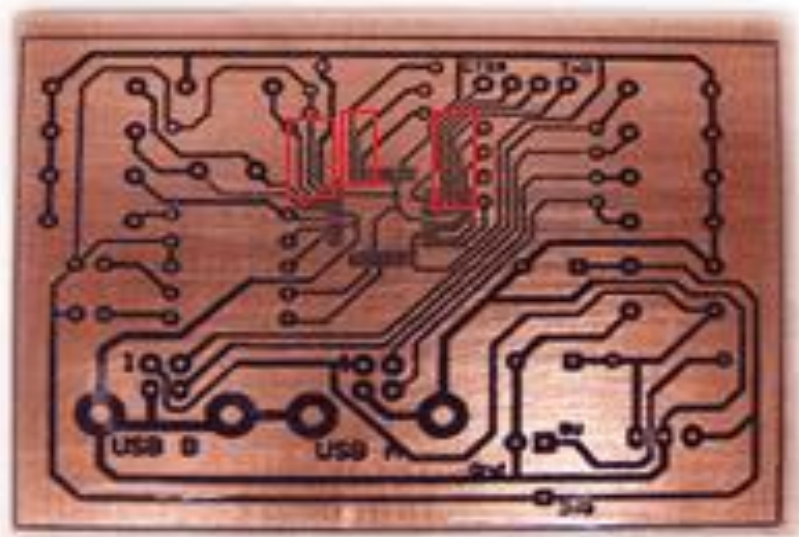
Pembuatan keping PCB dalam dunia industri tidak lepas dari hasil yang tidak sempurna atau cacat. Bagian terpenting adalah memisahkan antara PCB baik dengan PCB cacat, cacat pada keping PCB ada yang cacat sebelum proses pembuatan dan ada cacat yang setelah pengemasan.

Beberapa jenis cacat pada keping PCB menurut Kris (2003 : 2) diantaranya :

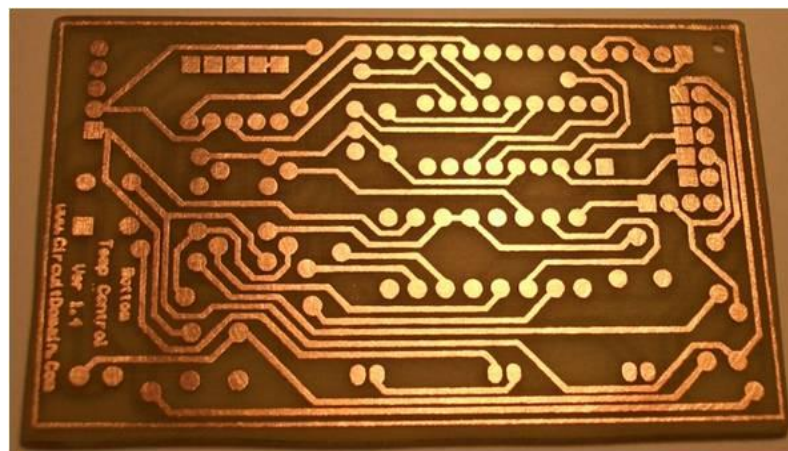
- a. Cacat karena proses pelarutan atau pengelupasan lapisan tembaga yang tidak sempurna sehingga mengakibatkan pelebaran dan penyempitan lapisan tembaga, serta juga adanya bercak pada keping PCB.
- b. Cacat pada saat pensablonan.
- c. Lapisan cat tembaga tidak merata dan rusak.

- d. Penulisan tata letak komponen tidak tepat.
- e. PCB rapuh dan berubah warna.
- f. Ujung siku PCB rusak karena pengemasan salah.

Cacat yang akan diidentifikasi pada tugas akhir ini adalah cacat pelebaran dan penyempitan jalur lapisan tembaga yang mengakibatkan adanya jalur hubung singkat, jalur hubung terbuka sehingga PCB tidak berfungsi.



Gambar 1. Keping PCB yang Jalur Membentuk Pulau



Gambar 2. Keping PCB Bagus Tanpa Cacat.

B. Pengolahan Citra Digital

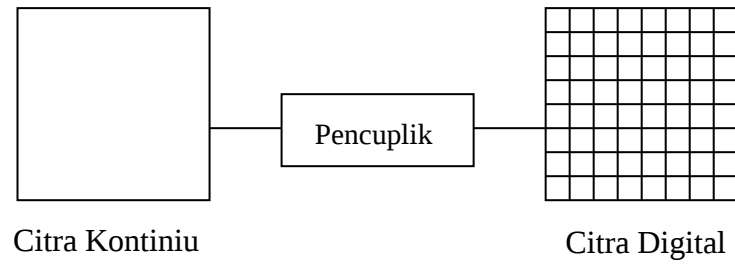
1. Pengertian Citra Digital

Rinaldi Munir (2004 : 15) “Citra adalah fungsi intensitas cahaya dari suatu objek dalam dua dimensi, yang dinotasikan dalam $f(x, y)$, dimana x dan y adalah koordinat titik citra, sedangkan nilai $f(x, y)$ merupakan tingkat intensitas citra pada titik tersebut”. $f(x, y)$ merupakan bentuk energi sehingga memiliki daerah intensitas nol sampai tak hingga.

Citra digital merupakan suatu matriks yang baris dan kolom menunjukkan letak titik di dalam citra, dan nilai elemen matriks menunjukkan tingkat keabuan di titik tersebut. Citra digital tidak selalu hasil langsung data rekaman suatu sistem, kadang-kadang hasil rekaman data bersifat kontiniu seperti gambar monitor televisi, karena itu citra digital diperlukan suatu konversi, sehingga citra dapat diolah komputer.

Sutoyo dkk (2009 :9) “Citra analog tidak dapat di representasikan dalam komputer sehingga tidak bisa diproses di komputer secara langsung”. Proses pengubahan citra analog menjadi citra digital melalui proses konversi yaitu melalui pembentukan kisi .

Proses pembentukan kisi-kisi arah horisontal dan vertikal, sehingga diperoleh gambar dalam bentuk larik (*array*) dua dimensi inilah yang disebut dengan digitasi atau sampling. Setiap elemen larik tersebut dikenal sebagai elemen gambar atau piksel.



Gambar 3. Proses Pencuplikan

Pembagian gambar menjadi sejumlah piksel dengan ukuran tertentu akan menentukan resolusi yang di peroleh, semakin tinggi resolusi yang di peroleh berarti semakin kecil ukuran pikselnya maka semakin halus gambar yang di peroleh karena informasi yang hilang akibat penglompokkan tingkat keabuan pada proses pembuatan kisi-kisi semakin kecil.

Citra digital hasil digitalisasi berbentuk matrik dengan ukuran $M \times N$ menurut Rinaldi Munir (2004 :19) akan berbentuk sebagai berikut :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,n-1) \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ f(m-1,0) & f(m-1,1) & f(m-1,2) & \dots & f(m-1,n-1) \end{bmatrix}$$

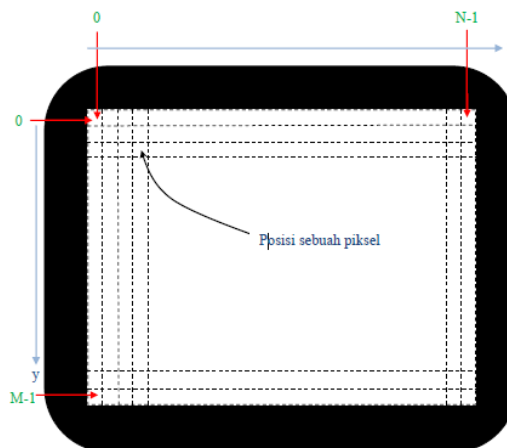
Gambar 4. Matrik Citra Digital

2. Piksel

Setiap elemen pada citra digital (elemen matriks) disebut *image element*, *picture element* atau *piksel*. Piksel adalah titik pada suatu citra yang telah didigitkan secara spasial dan terdiri dari $M \times N$ sample yang terdistribusi secara sama.

Citra hasil pengolahan komputer digital harus disimpan dalam format yang dapat diolah oleh program komputer, dengan cara membagi citra menjadi sekumpulan sel-sel diskret yang disebut piksel.

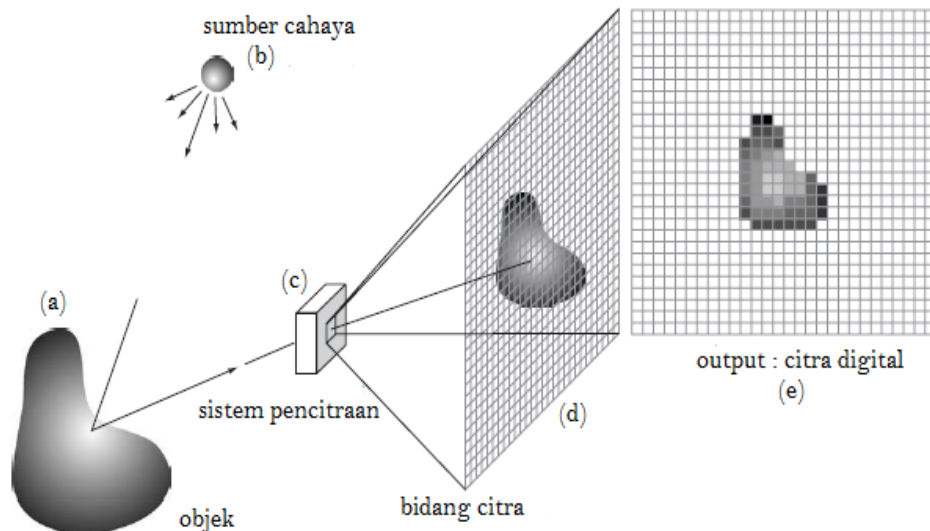
Citra pada umumnya dibagi menjadi kisi-kisi persegi, sehingga piksel sendiri adalah sebuah kisi-kisi persegi yang kecil. Piksel diberi nilai yang menyatakan warna atau tingkat kecerahan piksel, yang disebut intensitas piksel. Setiap piksel mempunyai koordinat posisi, seperti Gambar 5.



Gambar 5. Posisi Piksel dalam Sistem Koordinat Citra Berukuran $M \times N$

Piksel dalam sistem koordinat citra berupa (x,y) dimana, x menyatakan posisi kolom, y menyatakan posisi baris.

3. Proses Akuisisi Citra



Gambar 6. Contoh proses akuisisi citra digital (a) objek yang diambil gambarnya, (b) sumber cahaya, (c) sistem pencitraan (d) bidang citra, (e) citra digital

Sutoyo dkk (2009:11) menjelaskan proses akuisisi sebagai berikut : sebuah objek yang diambil gambarnya untuk dijadikan citra digital. Sumber cahaya diperlukan untuk menerangi objek, yang berarti ada intensitas cahaya (*brightness*) yang diterima objek. Oleh objek intensitas cahaya ini sebagian diserap dan sebagian lagi dipantulkan ke lingkungan sekitar objek secara radial. Sistem pencitraan (*imaging*) menerima sebagian dari intensitas cahaya yang dipantulkan oleh objek tadi.

Sensor optik yang terdapat di dalam sistem pencitraan digunakan untuk mendeteksi intensitas cahaya yang masuk kedalam sistem. Keluaran dari sensor ini berupa arus yang besar sebanding dengan intensitas cahaya yang mengenainya. Arus tersebut kemudian

di konvensi menjadi data digital yang kemudian dikirimkan ke unit penampil atau unit pengolah lainnya. Secara keseluruhan hasil keluaran sistem adalah data digital.

4. Elemen-Elemen Citra Digital

Citra digital mengandung komponen elemen-elemen dasar, dimana elemen tersebut dimanipulasi dalam pengolahan citra digital dan dieksploitasi lebih lanjut dalam computer vision. Menurut Rinaldi Munir (2004 : 25) elemen-elemen dasar yang penting, diantaranya :

a. Kecerahan (*brightness*)

Menurut Rinaldi Munir (2004:25) “Kecerahan adalah kata lain dari intensitas cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan”. Kecerahan pada sebuah titik (*pixel*) di dalam citra merupakan intensitas rata-rata dari area yang melingkupinya.

b. Kontras (*contrast*)

Kontras menerangkan tentang terang (*lightness*) dan gelap (*darkness*) sebuah gambar. Citra yang baik adalah citra yang komposisi gelap dan terangnya tersebar merata.

c. Kontur (*contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada pixel-pixel bertetangga. Karena adanya perubahan intensitas ini, mata kita mampu mendeteksi tepi-tepi objek di dalam citra.

d. Warna (*colour*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan objek. Setiap objek memiliki panjang gelombang yang berbeda.

Warna yang diterima oleh mata manusia merupakan hasil kombinasi cahaya dengan panjang gelombang yang berbeda. Warna merah mempunyai panjang gelombang yang paling tinggi, sedangkan warna ungu (*violet*) mempunyai panjang gelombang yang paling rendah. Kombinasi warna memberikan rentang warna yang paling lebar yaitu merah (*red*), hijau (*green*) dan biru (*blue*).

e. Bentuk (*shape*)

Menurut Rinaldi Munir (2004 : 26) “*Shape* adalah intrinsik dari objek tiga dimensi, dengan pengertian bahwa *shape* merupakan properti intrinsik untuk sistem visual manusia”. Pada umumnya citra yang dibentuk oleh mata manusia merupakan dwimatra (2 dimensi), sedangkan objek yang dilihat umumnya berbentuk trimatra (3 dimensi).

f. Tekstur (*texture*)

Menurut Rinaldi Munir (2004 : 26) “Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan *pixel-pixel* yang bertetangga”. Jadi tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang

cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Tekstur adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital.

Menurut Sutoyo dkk (2009 : 24) Kegunaan analisis tekstur :

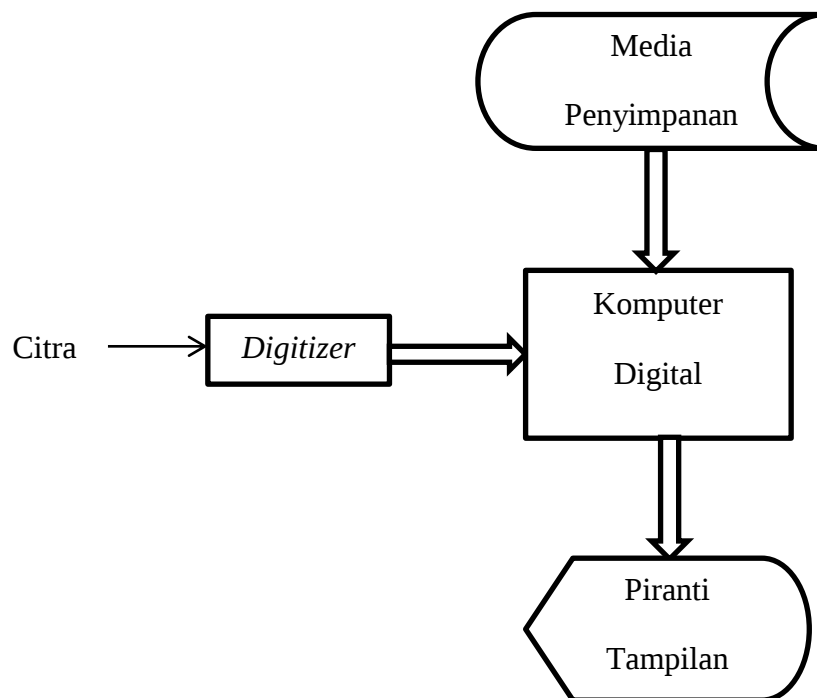
- 1) Tekstur memainkan peran penting dalam banyak tugas pada sistem visual, seperti pengelompokan objek pemandangan, penentuan bentuk objek dan lainnya.
- 2) Digunakan untuk segmentasi citra, mengidentifikasi pola-pola yang teratur dan berulang, pola-pola intensitas permukaan benda yang berhubungan dengan sifat kasar dan halus bahkan sampai sifat pada permukaan bumi atau planet lainnya.
- 3) Untuk tujuan pengolahan citra, analisis tekstur adalah menjadikan pola variasi lokal intensitas yang berulang sebagai pembeda, mana kala pola variasi tersebut terlalu kecil bila dibandingkan dengan objek yang diamati dalam resolusi yang dipakai.

Menurut Sutoyo dkk (2009 : 25) Syarat terbentuknya tekstur adalah :

- 1) Adanya pola primitif yang terdiri dari satu atau lebih piksel.
- 2) Pola-pola primitif muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik peluangnya.

5. Elemen Sistem Pemrosesan Citra Digital

Rinaldi Munir (2004 : 27) Elemen yang terlibat dalam pemrosesan citra digital dapat dibagi menjadi empat komponen yaitu :



Gambar 7. Elemen pemrosesan citra

- a. *Digitizer*, merupakan sistem penangkap citra digital yang melakukan penjelajahan citra dan mengkonversinya ke representasi numerik sebagai masukan bagi komputer digital.
- b. Komputer digital yang digunakan pada sistem pemrosesan citra dapat bervariasi dari komputer mikro ke komputer besar yang mampu melakukan berbagai macam fungsi pada citra digital resolusi tinggi.
- c. Piranti Tampilan, berfungsi mengkonversi matriks intensitas yang merepresentasikan citra ke tampilan yang dapat diinterpretasikan oleh mata manusia.

- d. Media penyimpanan, adalah piranti yang mempunyai kapasitas memori besar, sehingga gambar dapat disimpan secara permanen agar dapat diproses lagi pada waktu yang lain.

6. Jenis-Jenis Citra Digital

Sutoyo dkk (2009 : 21) Beberapa jenis citra digital yang sering digunakan adalah :

a. Citra Biner

Citra biner adalah citra yang hanya memiliki 2 nilai derajat keabuan yaitu hitam dan putih, dimana pixel-pixel objek bernilai 1 dan pixel-pixel latar belakang bernilai 0.

Keuntungan citra biner adalah :

- 1) kebutuhan memori kecil
- 2) waktu pemrosesan lebih cepat.

Citra biner didapatkan dengan cara meng-*threshold* kan citra *grayscale*. Operasi thresholding akan mengelompokkan nilai derajat keabuan setiap pixel ke dalam dua kelas, hitam dan putih.

b. Citra *Grayscale*

Citra skala keabuan mempunyai kemungkinan warna antara hitam pada bagian yang intensitas lemah (minimal) dan putih pada intensitas kuat (maksimal). Jumlah warna sesuai dengan bit penyimpanan yang digunakan.

c. Citra Warna

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar. Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak 2^8 untuk warna merah (R), 2^8 untuk warna Hijau (G) dan 2^8 untuk warna Biru (B), jadi $2^8 \cdot 2^8 \cdot 2^8 = 16$ juta warna. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

Penyimpanan citra *true color* di dalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* 256 gradasi warna diwakili 1 byte, sedangkan 1 piksel citra *true color* diwakili oleh 3 byte, dimana masing-masing byte mempresentasikan warna merah (*Red*), hijau (*Green*), dan biru (*Blue*).

Citra yang akan diolah adalah citra warna menjadi citra skala keabuan (*Grayscale*) dan citra biner. Citra yang dibaca dikenali sebagai citra warna (RGB), proses pengolahan citra menggunakan citra warna lebih sulit di lakukan, karena citra mengandung tiga komponen warna utama (merah, hijau, biru).

Pengolahan citra warna (RGB) membutuhkan pengolahan yang kompleks dan rumit, sehingga citra perlu di ubah menjadi citra beraras keabuan (*Grayscale*) dan citra biner untuk mempermudah pengolahan.

7. Format File Citra

Dua jenis format file citra yang sering digunakan dalam pengolahan citra, yaitu citra bitmap dan citra vektor.

a. Format File Citra Bitmap

Citra bitmap sering disebut juga dengan citra raster. Citra bitmap menyimpan data kode citra secara digital dan lengkap, dimana cara penyimpanannya adalah per pixel. Citra bitmap dipresentasikan dalam bentuk matriks atau dipetakan dengan menggunakan bilangan biner atau sistem bilangan lainnya. Citra Bitmap diperoleh dengan cara Scanner, Camera Digital, Video Capture, dan lain-lain. Sutoyo dkk (2009 : 26) Beberapa format yang umum digunakan dalam pemrograman pengolahan citra adalah :

Tabel 1. Format Citra File Bitmap

Nama Format	Ekstensi	Kegunaan
Microsoft Windows Bitmap Format	BMP	Format umum untuk menyimpan citra bitmap yang dikembangkan microsoft
Compuserve Graphics Interchange Format	GIF	Format umum citra yang dirancang untuk keperluan transmisi melalui modem
Aldus Tagged Image File Format	TIFF	Format kompleks dan multi guna yang dikembangkan oleh Aldus bersama Microsoft
WordPerfect Graphics Format	WPG	Format vektor yang juga mendukung citra bitmap

GEM Image Format	IMG	Format bitmap yang dikembangkan untuk riset digital dilingkungan GEM
Zsoft Pengolahan Citra Paintbrush Format	PCX	Dirancang untuk menyimpan citra layar dan merupakan format bitmap yang didukung luas
Microsoft Paint Bitmap Format	MSP	Secara fungsional mirip dengan IMG dan PCX, tapi kurang populer
AT & T Targa Format	TGA	Format untuk 16-bit dan 24-bit citra warna penuh diciptakan untuk sistem Truevision
Apple Macpaint Format	PNTG	Format asli dari Macintosh Macpaint program
Sun Microsystem rastes Format	RAS	Format bitmap asli yang digunakan pada Sun SPARCS
X Windows X-11 Bitmap Format	XBM	Format umum untuk menyimpan citra bitmap yang dikembangkan untuk X Windows.

b. Format File Citra Vektor

Citra vektor dihasilkan dari perhitungan matematis dan tidak berdasarkan piksel, yaitu data tersimpan bentuk vektor posisi, dimana yang tersimpan hanya informasi vektor posisi dengan bentuk sebuah fungsi.

Pada citra vektor, mengubah warna lebih sulit, tetapi membentuk objek dengan cara mengubah nilai lebih mudah, oleh karena itu jika citra diperbesar atau dikecilkan, kualitas citra tetap baik dan tidak berubah.

Beberapa format file citra vektor adalah AutoCAD drawing Format (DAG), AutoCAD drawing Exchange Format (DXF),

Microstation Drawing Format (DGN), dan scalable Vector Graphics (SVG).

c. Format File yang Digunakan

Format file citra yang digunakan dalam tugas akhir adalah format file citra Bitmap dengan berkas penyimpanan berekstensi ***. bmp** yang merupakan format citra yang baku di lingkungan sistem operasi Microsoft Windows. Format BMP memiliki ukuran berkas yang besar tetapi mempunyai kelebihan dari segi kualitas citra gambar.

Menurut Rinaldi Munir (2004 : 38) “Citra dalam format BMP lebih bagus dari pada citra dalam format yang lain, karena citra dalam format BMP umumnya tidak dimampatkan sehingga tidak ada informasi yang hilang.”

Citra dalam format BMP ada tiga macam, yaitu citra biner, citra berwarna, dan citra keabuan (*grayscale*). Pada citra yang mempunyai 256 warna, setiap piksel panjangnya 8-bit, tetapi komponen RGB-nya disimpan di dalam tabel RGB yang disebut palet. Setiap komponen panjangnya 8-bit, jadi ada 256 nilai untuk warna merah, 256 nilai untuk warna hijau, 256 nilai untuk warna biru. Pada citra keabuan, nilai $R = G = B$ untuk menyatakan bahwa citra keabuan hanya mempunyai satu kanal warna.

8. Peningkatan Kualitas Citra

a. Pengertian Peningkatan Kualitas Citra

Menurut Sutoyo dkk (2009 : 29) “Peningkatan kualitas citra adalah suatu proses untuk mengubah suatu citra menjadi citra baru sesuai dengan kebutuhan”. Bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra, memperoleh keindahan citra, kepentingan analisis citra, serta mengoreksi citra dari segala gangguan yang terjadi pada waktu perekaman data.

Menurut Sutoyo dkk (2009 : 29) Peningkatan kualitas citra dibagi dalam dua kategori, yaitu:

- 1) Metode domain spasial berdasarkan manipulasi langsung dari piksel didalam citra.
- 2) Metode domain frekuensi berdasarkan perubahan transformasi fourier pada citra.

b. Histogram

Menurut Sutoyo dkk (2009 : 29) “Histogram adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai gradasi warna”.

Menurut Sutoyo dkk (2009 : 31) Manfaat histogram adalah sebagai berikut :

- 1) Sebagai indikasi visual untuk menentukan skala keabuan yang tepat sehingga diperoleh kualitas citra yang diinginkan.

2) Untuk pemilihan batas ambang.

c. Transformasi Intensitas Citra

Peningkatan kualitas citra dapat dilakukan melalui transformasi citra, yaitu besar intensitas setiap piksel pada citra diubah, tetapi posisi piksel tetap. Transformasi ini dilakukan melalui sebuah fungsi yang disebut fungsi transformasi atau skala keabuan atau *Grayscale*.

Transformasi citra warna menjadi citra *grayscale*, citra warna diubah menjadi citra *grayscale* dengan cara menghitung rata-rata elemen warna *Red*, *Green*, dan *Blue*. Secara sistematis perhitungannya sebagai berikut :

$$f_o(x,y) = \frac{f_i^R(x,y) + f_i^G(x,y) + f_i^B(x,y)}{3} \quad (1)$$

Dimana :

$f_o(x,y)$ = fungsi output citra

$f_i^R(x,y)$ = fungsi input citra untuk *Red*

$f_i^G(x,y)$ = fungsi input citra untuk *Green*

$f_i^B(x,y)$ = fungsi input citra untuk *Blue*.

C. Metode Template Matching

1. Pengertian Metode Template Matching

Menurut Ion Marques (2010 :40) “*Template matching* adalah proses mencari suatu objek (template) pada keseluruhan objek yang berada dalam suatu citra”. *Template* dibandingkan dengan keseluruhan

objek tersebut dan bila *template* cocok (cukup dekat) dengan suatu objek yang belum diketahui pada citra tersebut maka objek tersebut ditandai sebagai *template*. Perbandingan antara *template* dengan keseluruhan objek pada citra dapat dilakukan dengan menghitung selisih jaraknya sebagai berikut :

$$D(m,n) = \sum_j \sum_k [f(j,k) - T(j-m, k-n)]^2 \quad (2)$$

Dimana $f(j,k)$ menyatakan citra tempat objek yang akan dibandingkan dengan *template* $T(j,k)$, sedangkan $D(m,n)$ menyatakan jarak antara *template* dengan objek pada citra.

Ion Marques (2010: 42) menyatakan Template matching dapat dibagi menjadi dua pendekatan, yaitu :

a. Pendekatan Berbasis Fitur

Pendekatan ini tidak mempertimbangkan keseluruhan dari *template* gambar, komputasi dapat lebih efisien bekerja dengan sumber gambar beresolusi lebih besar.

b. Pendekatan Berbasis Template

Pendekatan berbasis *template*, memerlukan sampling dari sejumlah besar poin, untuk mengurangi jumlah sampling poin dengan mengurangi resolusi pencarian dan *template* gambar oleh faktor yang sama, menyediakan pencarian titik data dalam pencarian gambar sehingga *template* tidak harus mempunyai pencarian titik data.

Teori *Template Matching* memiliki keunggulan dan kekurangan, yaitu :

1) Keunggulan

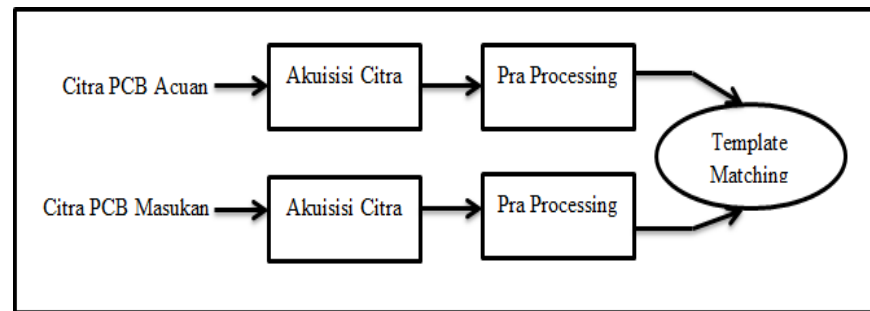
Dikutip dari Abikindo (2010), keunggulan teori *Template Matching* adalah :

- a) Jelas bahwa mengenal bentuk, huruf atau bentuk-bentuk visual lainnya diperlukan kontak dengan bentuk-bentuk internal.
- b) *Template matching* adalah prosedur pengenalan pola yang sederhana didasarkan pada ketetapan konfigurasi informasi penginderaan dengan konfigurasi otak.

2) Kelemahan

Dikutip dari Abikindo (2010) kelemahan dari teori *Template Matching* adalah “Jika perbandingan eksternal objek dengan internal objek 1:1, maka objek yang berbeda sedikit saja dengan template tidak akan dikenali”.

Ion Marques (2010 : 40) “*Template Matching* merepresentasikan model, pixel, sampel, kurva dan tekstur. Pengenalan ini berfungsi untuk sebuah korelasi atau pengukuran jarak.”



Gambar 8. Diagram Algoritma *Template Matching*

a) Akuisisi Citra

adalah tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra adalah menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap dimulai dari objek yang akan diambil gambarnya.

Beberapa peralatan yang digunakan untuk proses akuisisi citra adalah : kamera digital, video kamera, scanner.

b) Pra Processing

adalah tahapan-tahapan pengolahan citra digital, hal yang dilakukan pada tahapan ini adalah : Pengubahan citra warna ke aras keabuan, peningkatan mutu citra, pengambangan.

c) Template Matching

adalah metode untuk membandingkan antara citra PCB acuan dengan citra PCB masukan, untuk mengidentifikasi cacat pada keping PCB.

2. Metode Template Matching Untuk PCB

Menurut Ion Marques “Salah satu metode dalam pemisahan atau segmentasi citra adalah pencocokan (*matching*) dengan

menggunakan *template*”. Metode *Template Matching* digunakan untuk mengklasifikasikan objek, dengan cara membandingkan antara citra masukan PCB dengan citra PCB sumber acuan yang telah disimpan. Citra masukan dan citra sumber acuan didapatkan dari PCB yang dipindai.

Metode *Template Matching* sering digunakan untuk mengidentifikasi citra karakter huruf, angka, objek sederhana. Proses membandingkan antara citra masukan PCB berupa aras keabuan dengan citra PCB sumber acuan berupa aras keabuan, akan didapatkan nilai korelasi antara -1 dan +1.

Hal-hal yang mempengaruhi dalam pengidentifikasian cacat pada PCB adalah :

a. Posisi

Citra PCB bisa berbeda-beda tergantung pada keadaan waktu diambil oleh pemindai, bisa karena pergeseran, perpindahan, dan perputaran posisi. Oleh karena itu, posisi citra perlu diperbaiki apabila posisi citra tidak tepat, sebelum menuju ke proses *Template Matching*.

b. Kondisi Citra

Ketika sebuah citra dibentuk, faktor seperti pencahayaan dan karakteristik pemindai mempengaruhi citra PCB. Penerapan metode *Template Matching* pada identifikasi kecacatan PCB dapat dilakukan dengan utama sebagai berikut :

- 1) Pengepasan posisi : dilakukan dengan mencuplikasi 80 % area citra untuk mendapatkan posisi ideal.
- 2) Hitung nilai korelasi silang: Kris (2003 : 3) untuk mengklasifikasikan suatu citra PCB adalah baik tanpa cacat sedikit pun, maka nilai korelasi adalah 1 dan cacat total maka nilai korelasinya adalah -1 rumus yang digunakan adalah :

$$r = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1} (x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1} (y_i - \bar{y})^2}} \quad (3)$$

Dengan :

x = Template berupa citra keabuan
 $\bar{x}$ = Nilai rata-rata template berupa citra keabuan
 y = Citra sumber berupa citra keabuan
 $\bar{y}$ = Nilai rata-rata citra sumber, citra keabuan
 N = Jumlah pixel pada citra
 r = Nilai korelasi silang pada citra aras keabuan

- 3) Deteksi akhir : dari nilai korelasi yang didapat, nilai tersebut kemudian dikonversikan dalam rentang 0 sampai 255 pada *channel red* untuk digambarkan dalam bentuk segiempat pada titik koordinat citra PCB yang cacat.

Prinsip metode ini adalah membandingkan antara image objek yang akan dikenali dengan image template yang ada. Image objek yang akan dikenali mempunyai tingkat kemiripan sendiri terhadap masing-

masing image template. Pengenalan dilakukan dengan melihat nilai tingkat kemiripan berada dibawah nilai batas ambang maka image objek tersebut dikategorikan sebagai objek tidak dikenal.

Adhitya (2008 : 2) Metode *Template Matching* memiliki beberapa karakteristik seperti :

- a) Relatif mudah untuk diaplikasikan dalam teknik pengolahan citra digital.
- b) Memiliki hasil akurat karena mendeteksi kesalahan hingga ukuran piksel.
- c) Walaupun demikian, metode ini rentan terhadap perbedaan orientasi antara citra acuan (template) dengan citra yang diidentifikasi, yang meliputi: ukuran, posisi dan kualitas citra.
- d) Untuk mendapatkan hasil yang maksimal maka metode ini sangat tergantung pada teknik pengolahan citra digital yang lain seperti filtering, enhancement dan lain-lain.

Implementasi metode *Template Matching*, diperlukan operasi pengolahan citra digital, yaitu :

a. Penapisan citra (*Filter*)

Proses ini dilakukan bila citra memiliki derau atau noise sehingga perlu dihaluskan dengan tapis citra. Perancangan tapis dengan memanipulasi pixel-pixel tetangga membuat citra lebih halus, bentuk sudut, dan tepi citra tetap terjaga.

Pada proses perekaman citra digital terjadi gangguan yang bersifat frekuensi rendah, dimana terjadi proses pemerataan intensitas cahaya pada suatu titik sampel dengan titik-titik tetangganya. Gangguan lain yang sering terjadi pada proses perekaman citra digital adalah gangguan berbentuk garis-garis akibat

adanya kerusakan pada sebagian sektor detektor sensor serta juga dijumpai bercak hitam yang acak.

Proses penapisan dengan tapis median berfungsi untuk menghilangkan tampilan gambar yang mengandung bercak-bercak putih atau hitam. proses tapis medina di lakukan per kelompok piksel yang terdiri dari suatu piksel yang akan diproses dengan piksel-piksel tetangga.

Gambar 9 merupakan proses penapisan dengan menggunakan tapis median yang dilakukan pada matriks 5 x 5. Nilai intensitas piksel disusun dari nilai piksel terendah sampai piksel tertinggi, kemudian nilai intensitas yang baru dari piksel yang sedang diproses adalah sama dengan median, yaitu nilai piksel yang berada pada posisi tengah.

$$\begin{bmatrix} 2 & 5 & 5 & 5 & 15 \\ 2 & 5 & 5 & 12 & 4 \\ 5 & 4 & 32 & 10 & 4 \\ 4 & 4 & 15 & 15 & 3 \\ 16 & 4 & 5 & 5 & 4 \end{bmatrix}$$

Gambar 9. Penapisan dengan Tapis Median pada Matriks 5 x 5

Nilai intensitas piksel pada Gambar 9, dan delapan nilai piksel tetangga disusun dari nilai terendah sampai nilai tertinggi, yaitu :

4 4 5 5 10 12 15 15 32

nilai yang digaris bawah, merupakan nilai median.

b. Pengembangan (*Threshold*)

Proses ini digunakan untuk mengubah citra format keabuan yang mempunyai nilai lebih dari dua ke format citra biner yang hanya memiliki dua nilai (0 atau 1). Dalam hal ini titik dengan rentang nilai keabuan tertentu diubah menjadi warna hitam dan sisanya menjadi warna putih atau sebaliknya.

c. Perbaikan Kualitas Citra (*Enhancement*)

Perbaikan kualitas citra dilakukan dengan penapisan citra, yang hanya bersifat pilihan. Jika citra yang akan dianalisis masih mengandung derau maka dilakukan penapisan.

3. Algoritma Identifikasi Cacat dengan Metode *Template Matching*

Kris (2003 : 3) Algoritma identifikasi cacat pada PCB menggunakan metode *Template Matching* adalah sebagai berikut:

1. Membaca berkas citra berwarna dengan format .bmp. citra yang dibaca adalah citra acuan yang berupa citra berwarna, sedangkan citra yang diolah adalah citra digital yang diperoleh dari hasil pemindaian PCB.
2. Melakukan proses pengubahan citra acuan menjadi citra aras keabuan. Hal ini dilakukan karena citra yang dibaca dikenali sebagai citra warna (RGB). Proses pengolahan citra warna lebih sulit dilakukan karena citra warna mengandung tiga komponen warna utama yang membutuhkan pengolahan lebih kompleks, sehingga citra perlu diubah terlebih dahulu menjadi citra beraras keabuan untuk mempermudah pengolahan.
3. Melakukan proses penipisan citra, bertujuan untuk menghilangkan *noise* sehingga diperlukan penghalusan.
4. Menerapkan proses pengembangan terhadap citra aras keabuan. Wilayah tembaga mempunyai warna yang lebih terang dibanding bagian lain dari citra, maka wilayah citra tembaga dipisahkan dari wilayah lainnya melalui proses pengembangan.

5. Melakukan pencocokan model (*Template Matching*) antara PCB masukan dengan citra PCB acuan. Sebelum masuk ke proses *Template Matching*, dilakukan proses pengambilan citra masukan dan diikuti pengolahan citra dengan parameter seperti yang dilakukan oleh citra acuan.
6. Menggambarkan segi empat pada citra masukan apabila terjadi cacat pada keping PCB.
7. Deteksi akhir hasil pengolahan dideskripsikan dalam bentuk teks yang menyatakan PCB dalam keadaan baik apabila tidak terjadi cacat dan apabila terjadi cacat maka akan ditampilkan posisi cacat citra PCB yang mengalami cacat. Penandaan letak cacat dengan pemberian warna yang besarnya sesuai nilai blok korelasi yang diisikan dan intensitasnya berdasarkan nilai korelasi yang didapatkan.

D. Flowchart

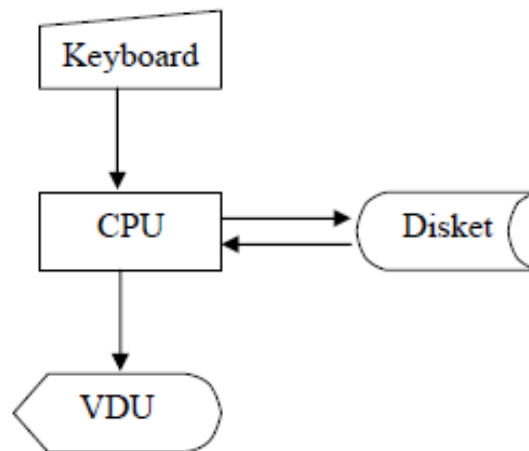
1. Pengertian Flowchart

Ramadona (2004:1) “Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah”. Flowchart merupakan cara penyajian dari suatu algoritma.

Tujuan membuat flowchart menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas serta menggunakan simbol-simbol standar. Ramadona (2004: 2) Dua model penulisan Flowchart yaitu :

a. Sistem Flowchart

adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file di dalam media tertentu.



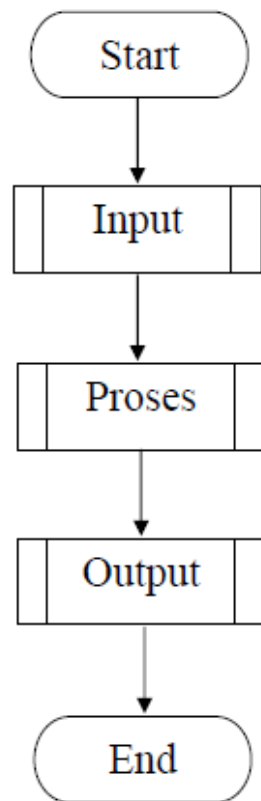
Gambar 10. Contoh Sistem Flowchart

b. Program Flowchart

adalah bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Dua jenis metode penggambaran program flowchart adalah :

1) *Conceptual flowchart*

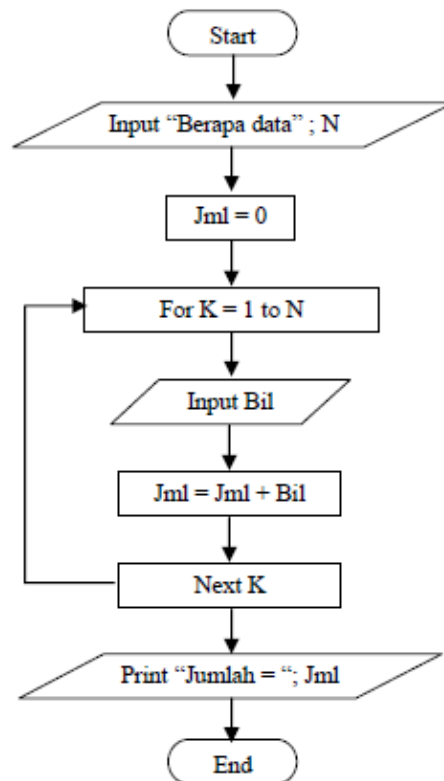
Menggambarkan alur pemecahan masalah secara global.



Gambar 11. Contoh Conceptual Flowchart

2) *Detail flowchart*

Berfungsi menggambarkan alur pemecahan masalah secara rinci.



Gambar 12. Contoh Detail Flowchart

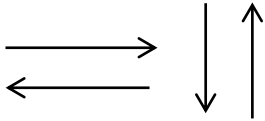
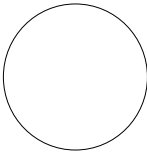
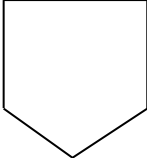
2. Simbol-Simbol Flowchart

Ramadona (2004:4) Simbol-simbol yang dipakai dalam flowchart dibagi menjadi tiga kelompok, yaitu :

a. Flow Direction Symbols

berfungsi untuk menghubungkan simbol yang satu dengan yang lain, disebut juga *connection line*.

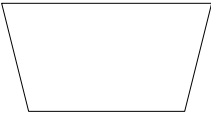
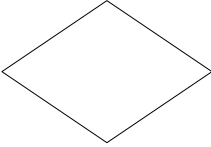
Tabel 2. Flow Direction Symbols

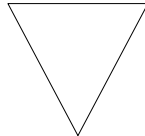
	Simbol arus <i>I flow</i> , yaitu menyatakan jalannya arus suatu proses.
	Simbol <i>communication link</i> , yaitu menyatakan transmisi data dari satu lokasi ke lokasi lain.
	Simbol <i>connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
	Simbol <i>offline connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman berbeda.

b. Processing Symbols

Berfungsi untuk menunjukkan jenis operasi pengolahan dalam suatu proses/ prosedur.

Tabel 3. Processing Symbols

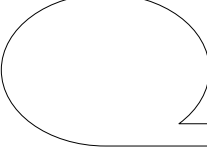
	Simbol <i>process</i> , yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
	Simbol <i>manual</i> , yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
	Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
	Simbol predefined process, yaitu menyatakan penyediaan tempat

	penyimpanan suatu pengolahan untuk memberi harga awal.
	Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program.
	Simbol <i>keying operation</i> , menunjukkan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.
	Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
	Simbol <i>manual input</i> , memasukkan data secara manual dengan menggunakan <i>online</i> keyboard.

c. Input / Output Symbols

Berfungsi untuk menunjukkan jenis peralatan yang digunakan sebagai media input atau output.

Tabel 4. Input / Output Symbols

	Simbol <i>input/output</i> , yaitu menyatakan proses input atau output tanpa tergantung jenis peralatan.
	Simbol <i>punched card</i> , yaitu menyatakan input berasal dari kartu atau output ditulis ke kartu.
	Simbol <i>magnetic tape</i> , yaitu menunjukkan input berasal dari pita magnetis atau output disimpan ke pita magnetis.

	Simbol <i>disk storage</i> , yaitu menyatakan input berasal dari disk atau output disimpan ke disk.
	Simbol <i>document</i> , yaitu mencetak keluaran dalam bentuk dokumen.
	Simbol <i>display</i> , mencetak keluaran dalam layar monitor.

E. Pemograman Delphi

1. Pengenalan Delphi 7.0

Dikutip dalam buku Panduan Praktis Pemograman Delphi 7.0 (2003:1) “Delphi berasal dari bahasa pemograman yang cukup terkenal, yaitu bahasa *pascal*”. Bahasa *pascal* diciptakan pada tahun 1971 oleh ilmuwan dari swiss, yaitu Niklaus Wirth. Nama *pascal* diambil dari ahli matematika dan filsafat Perancis, yaitu *Blasie Pascal* (1623-1622).

Pemograman *windows* dengan *turbo pascal* di anggap cukup sulit, maka pada tahun 1993 *Borland Internasional* mengembangkan bahasa *pascal* yang bersifat visual. Hasil pengembangan ini adalah dirilisnya Delphi 1 pada tahun 1995. Perkembangan Delphi tidak berhenti sampai disitu. Pada tahun 1996 *Borland Internasional* merilis Delphi 2 untuk *windows 95/NT*. Kemudian tahun-tahun berikutnya *Borland Internasional* merilis beberapa versi pengembangan Delphi yang memiliki tambahan fitur baru dibanding sebelumnya.

Lingkungan kerja yang terdapat pada Delphi 7 dalam buku Panduan Praktis Pemograman Delphi 7.0 adalah sebagai berikut :

a. Menu

Menu Delphi memiliki kegunaan seperti pada aplikasi windows lainnya. Dari menu ini, bisa digunakan untuk memanggil atau menyimpan program, menjalankan program dan lainnya. Sesuatu yang berhubungan dengan IDE delphi dapat dilakukan dari menu.

b. Spee dbar

Spee bar atau yang sering dikenal dengan *toolbar* berisi kumpulan tombol yang tidak lain adalah pengganti dari beberapa item menu yang sering digunakan. Dengan kata lain, setiap tombol pada toolbar menggantikan salah satu item menu

c. Component Palette

Component palette berisi kumpulan icon yang melambangkan komponen-komponen pada VCL (*Visual Component Library*) atau CLX (*Component Library for Cross Platform*). Pada komponen palette terdapat beberapa tab yaitu : standard, additional, data accsess, dan tab lainnya. Icon yang ditampilkan pada *Component Palette* tidak memiliki keterangan yang menyatakan nama komponen yang dapat dilihat pada Gambar 13.



Gambar 13. Component Palette Delphi

d. Form Designer

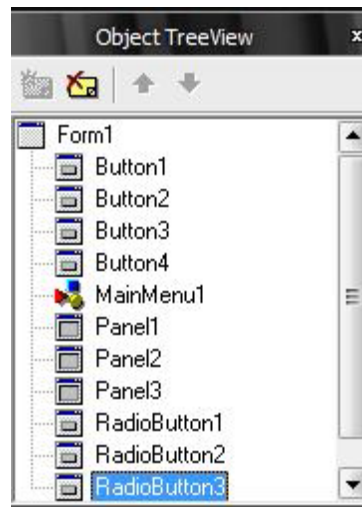
Sesuai namanya, form designer merupakan tempat untuk merancang jendela aplikasi. Perancangan form dilakukan dengan meletakkan komponen-komponen yang diambil dari *component palette*.

e. Code Explorer

Code explorer adalah tempat dimana akan menulis program. Disini diletakkan pernyataan-pernyataan dalam bahasa *pascal*. Yang perlu diperhatikan dalam code explorer adalah tidak perlu menuliskan semua kode sumber. IDE Delphi telah menuliskan semacam kerangka program.

f. Object Treeview

Object treeview berisi daftar komponen yang telah diletakkan pada form designer.



Gambar 14. Object Treeview Delphi

g. Object Inspector

Object inspector digunakan untuk mengubah karakteristik sebuah komponen, pada *object inspector* dapat dilihat terdapat dua tag yaitu properties dan events. Pada tags properties digunakan untuk mengubah properti dari komponen. Pada tags events digunakan untuk menyisipkan kode menangani kejadian tertentu.



Gambar 15. Object Inspector Delphi

h. Variabel

Dalam dunia pemograman, variabel digunakan untuk menyimpan data. Pada Delphi, pendeklarasian variabel mengikuti sintak dibawah ini:

Var nama_variabel1:tipe_variabel;

Variabel pada Delphi harus mengikuti beberapa aturan sebagai berikut :

- 1) Nama variabel maksimum terdiri 63 karakter.
- 2) Nama variabel hanya boleh mengandung huruf, angka garis bawah, dan tidak boleh diawali dengan angka.

2. Kebutuhan Sitem Minimum

Dikutip dalam buku Panduan Praktis Pemograman Delphi 7.0 (2003:8) Untuk mendapatkan unjuk kerja Delphi dengan baik, paling tidak sistem komputer harus memiliki spesifikasi sebagai berikut:

- a. Menggunakan prosessor Intel Pentium 166MHz atau yang lebih tinggi.
- b. Sistem operasi Microsoft Windows 98, 2000, atau XP.
- c. RAM 256 MB.
- d. Ruang hard disk kosong yang diperlukan untuk instalasi lengkap sekitar 475 Mb.
- e. Memiliki CD-ROM drive.

- f. Menggunakan monitor VGA atau monitor dengan resolusi yang lebih tinggi.
- g. Memiliki mouse atau alat pointing lainnya.

F. Penelitian Yang Relevan

1. Thiang dkk (2001) “ Penerapan metode *Template Matching* untuk pendeteksian kendaraan”. Hasil penelitian disimpulkan bahwa penggunaan metode template matching telah memberikan hasil yang memuaskan untuk mengenali jenis-jenis kendaraan.
2. Bowo Leksono (2011) “ Aplikasi Metode *Template Matching* untuk klasifikasi sidik jari”. Hasil penelitian ini disimpulkan metode template matching dapat mencocokkan sidik jari acuan dengan sidik jari hasil pengolahan.
3. Nur Wakhidah (2012) “ Deteksi plat nomor kendaraan bermotor berdasarkan area pada *Image Segmentation*”. Hasil penelitian disimpulkan bahwa pengolahan citra digital dengan mengenali tekstur pada plat kendaraan bermotor memberikan hasil yang memuaskan untuk mengenali plat kendaraan.
4. Kris Adhy Nugroho. (2003) “ Identifikasi Cacat Pada Keping PGB Menggunakan Pencocokan Model (*Template Matching*)”. Hasil penelitian disimpulkan bahawa pengolahan citra digital dengan pencocokkan model dapat menggambarkan ada tidaknya cacat pada keping PCB.

BAB V PENUTUP

A. Kesimpulan

Kesimpulan dari pembuatan program untuk mengidentifikasi cacat pada keping PCB adalah:

1. Tersedianya sebuah aplikasi untuk mengidentifikasi cacat jalur pada sebuah keping PCB.
2. Metode *Template Matching* dapat diterapkan dalam pengolahan citra digital, khususnya untuk mengidentifikasi dan menganalisis cacat pada keping PCB.

B. Saran

Saran-saran yang diberikan setelah membuat program untuk mengidentifikasi cacat pada keping PCB, adalah:

1. Diperlukan pengembangan penelitian untuk format berkas pada citra masukan yang berguna untuk mengoptimalkan dan menambah kemampuan sistem dalam mengolah berbagai berkas citra.
2. Diperlukan pengembangan penelitian untuk rotasi citra masukan guna mengoptimalkan kemampuan sistem dalam mendeteksi cacat pada keping PCB.
3. Diperlukan pengembangan penelitian lanjutan menggunakan bahasa pemrograman lainnya untuk diteliti keefisienan waktu dan keringkasan pengkodean program.

DAFTAR PUSTAKA

- Achmad Hidayatno dkk. (2005). "Penentuan Wilayah Wajah Manusia Pada Citra Berwarna Berdasarkan Kulit Wajah Dengan Metode *Template Matching*". Jurnal Teknologi Elektro.
- Adhita Wishnu Wardhana, (2008). "Penggunaan Metode Template Matching Untuk Identifikasi Kecacatan PCB". Jurnal Teknologi Informasi
- Bowo Leksono. (2011). " Aplikasi Metode Template Matching Untuk Klasifikasi Sidik Jari". Jurnal Teknik Elektro. Universitas Diponegoro
- Chogwang. 2014. " Mengenal Dasar Jenis dan Bagian PCB " www.Chogwang.com. Diakses 02 Februari 2015.
- Elean. 2010. "Template Matching". www.akibindo.blogspot. Diakses 24 April 2014.
- Hasmi. 2014. " Pengenalan tentang PCB " [http //hasmimunikom.blogspot.com](http://hasmimunikom.blogspot.com). Diakses 02 Februari 2015.
- Kris Adhy Nugroho. (2003). " Identifikasi Cacat Pada Keping PCB Menggunakan Pencocokan Model (*Template Matching*). " Jurnal Teknik Elektro. Universitas Diponegoro
- Marques Ion. (2010). *Face Regoction Algorithms* . Jurnal. Universidad del Pais Vasco.
- Mohammad Imran. (2013). *Pengolahan Citra* . Jurnal. Teknik Informatika
- Mulyadi,Thiang (2002). "Pengenalan Rambu Lalu Lintas Sederhana Dengan Menggunakan Metode *Template Matching*, Proceedings, Komputer dan Sistem Intelegen." *Skripsi*. Universitas Gunadarma
- Murni,Aniati. (1992). *Penghantar Pengolahan Citra*. Elex Media Komputindo.
- Nur Wakhidah. (2012). " Deteksi Plat Nomor Kendaraan Bermotor Berdasarkan Area Pada *Image Segmentation*. Jurnal Transformatika. Universitas Semarang
- Ramadona Nilawati.(2004). " Definisi dan Simbol Flowchart". Jurnal. Universitas Gunadarma.
- Rinaldi Munir. (2004). *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Bandung : Informatika.

T. Sutoyo. *et. al.* (2009). *Teori Pengolahan Citra Digital*. Yogyakarta : ANDI

Wanahana Komputer. (2003). "Panduan Praktis Pemograman Borland Delphi7.0.
Semarang : ANDI