

**PENGEMBANGAN SISTEM PENDETEKSI PLAT NOMOR DI
TEMPAT PARKIR MENGGUNAKAN WEBCAM**

SKRIPSI



**Disusun Oleh :
ARI YOSWARA
08.12.909**

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012**

THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
NATIONAL BUREAU OF STANDARDS

1975

1975
1975
1975

THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
NATIONAL BUREAU OF STANDARDS
100 BUREAU DRIVE
GAITHERSBURG, MARYLAND 20899
301-975-3000

LEMBAR PERSETUJUAN

PENGEMBANGAN SISTEM PENDETEKSI NOMOR PLAT MOBIL DI TEMPAT PARKIR MENGGUNAKAN WEBCAM

SKRIPSI

Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna memperoleh gelar Sarjana Teknik

Disusun Oleh :

ARI YOSWARA
NIM : 08.12.909

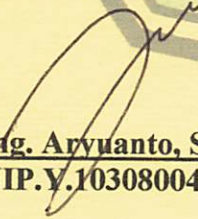
Mengetahui

Ketua Jurusan Teknik Elektro S-1


Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.101880089

Diperiksa dan Disetujui

Dosen Pembimbing I


(Dr. Eng. Arvuanto, ST, MT)
NIP.Y.1030800417

Dosen Pembimbing II


(Ahmad Faisal, ST)
NIP.Y.0721048502

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2012

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Ari Yoswara
NIM : 0812909
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

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, Desember 2012



Ari Yoswara
0812909

PENGEMBANGAN SISTEM PENDETEKSI NOMOR PLAT MOBIL MENGUNAKAN WEBCAM DI TEMPAT PARKIR

Ari Yoswara

**Jurusan Teknik Elektro S-1, Konsentrasi T.Komputer dan Informatika
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
arihurt@rocketmail.com**

Dosen Pembimbing : **1. Dr. Eng. Aryuanto Soetedjo, ST, MT**
 2. Ahmad Faisol, ST

Abstraksi

Sampai saat ini untuk mendapatkan data nomor plat mobil sebagian banyak masih menggunakan cara manual. Cara manual ini kurang efektif dan efisien. Apabila pengambilan data dilakukan dengan kamera (*webcam*), lalu diproses dengan PC (*Personal Computer*), maka jalannya proses akan jadi lebih efisien, karena tidak memerlukan banyak tenaga untuk pengambilan data. Skripsi ini dibuat sebuah program pendeteksi nomor plat mobil dengan pencitraan digital menggunakan *webcam*. *Image* dari *webcam* kemudian diubah ke gambar *greyscale* sehingga menjadi warna hitam putih dilanjutkan dengan pengolahan citra *invers biner* dimana untuk warna putih diindikasikan dengan biner 1 dan warna hitam diindikasikan dengan biner 0. Kemudian dilakukan proses *edge detection* (deteksi tepi) untuk diambil selisih piksel antara biner 1 dan biner 0 sehingga dapat diambil batas tepi plat nomor.

Kata kunci: *webcam, edge detection, invers biner, greyscale.*

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiratMu Ya Allah yang telah memberikan Rahmat dan HidayahNya, sehingga penulis dapat menyelesaikan skripsi dengan judul **“PENGEMBANGAN SISTEM PENDETEKSI NOMOR PLAT MOBIL MENGGUNAKAN WEBCAM DI TEMPAT PARKIR”** dengan lancar. Skripsi merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noetjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1,
4. Bapak Dr. Aryuanto Soetedjo, ST, MT selaku Dosen pembimbing I.
5. Bapak Ahmad Faisol, ST selaku Dosen Pembimbing II.
6. Ayah dan Ibu serta saudara-saudara yang selalu memberikan do'a, motivasi dan semangat.
7. Teman - teman Lab. PK&M yang selalu memberikan motivasi dan semangat.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Juli 2012

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	xi
BAB I	
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	2
1.5. Metode Penelitian	2
1.6. Sistematika Penulisan	3
BAB II	
LANDASAN TEORI	4
2.1. Citra Digital	4
2.2. Pixel dan Resolusi Citra	4
2.3. Edge Detection	5
2.4. Plat Nomor Mobil	6
2.5. Jenis-Jenis Plat Nomor Mobil	7
2.6. Jenis Citra	8

2.6.1. Citra Berwarna	8
2.6.2. Konversi Citra Berwarna	9
2.6.3. Koversi Citra <i>Greyscale</i> Menjadi Citra Biner	11
2.7. Segmentasi Citra	12
2.7.1. Pengambangan (<i>Thresholding</i>)	12
2.7.2. Metode Histogram	12
2.8. Color Filtering	14
2.9. Webcam	15
2.9.1. Cara Kerja Webcam	16
2.10. Delphi 7	17
2.11. Invert Biner	18
BAB III	
ANALISA DAN DESAIN SISTEM	19
3.1. Analisa sistem	19
3.2. Data Penelitian	19
3.3. Penentuan Spesifikasi Alat	20
3.4. Desain Sistem	21
3.5. Perancangan Dan Implementasi Sistem	21
3.6. Blok Diagram Jalanya Program	22
3.7. Aplikasi Pendeteksi Plat Mobil	24
3.7.1. Blok Diagram Jalanya Program	24
3.7.2. Perancangan Perangkat Lunak	25
3.7.3. Perancangan Inisialisasi Webcam	26
3.7.4. Perancangan Program	27
3.8. Implemtasi Interface <i>Open Local Capture Device</i>	32

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM	33
4.1. Implementasi Sistem	33
4.1.1. Implementasi Antarmuka untuk Menjalankan Aplikasi	34
4.2. Pengujian Sistem	34
4.2.1. Pengujian Pendeteksian Plat Nomor Menggunakan Webcam Terhadap Kondisi yang diamati	35
4.2.2. Pengujian Pendeteksian Plat Nomor Menggunakan Webcam Terhadap jarak	37
4.2.3. Pengujian pendeteksian plat nomor menggunakan webcam terhadap warna Background	40
4.2.6. Analisa faktor kegagalan	43
4.2.7. Faktor kegagalan pendeteksian plat nomor	43

BAB V

PENUTUP	46
5.1. Kesimpulan	46
5.2. Saran	46
DAFTAR PUSTAKA	48
LAMPIRAN	

DAFTAR GAMBAR

2.1.	Representasi citra digital.....	4
2.2.	<i>Image size and resolution</i>	5
2.3.	Proses Deteksi Tepi Citra.....	5
2.4.	Hasil beberapa deteksi tepi.....	6
2.5.	Plat nomor pribadi.....	7
2.6.	Plat nomor kendaraan umum.....	8
2.7.	Plat nomor kendaraan pemerintah.....	8
2.8.	Citra berwarna dengan kanal setiap warna dasar penyusunnya.....	9
2.9.	Pemisahan Citra berwarna pada tiap kanal RGB dan luminance.....	10
2.10.	Konversi citra berwarna menjadi citra <i>grayscale</i>	10
2.11.	Konversi citra <i>grayscale</i> menjadi citra biner.....	12
2.12.	Histogram.....	13
2.13.	histogram warna.....	14
2.14.	Ruang warna HSL.....	14
2.15.	Webcam tipe MD-907.....	16
2.16.	Tampilan umum program Delphi 7.....	17
2.17.	Invert biner.....	18
3.1.	Diagram program.....	19
3.2.	Flowchart diagram sistem.....	21
3.3.	Flowchart jalannya program.....	22
3.4.	Flowcart operasi penyeleksian pixel untuk mencari posisi plat nomor ..	23
3.5.	Perancangan Sistem Dari Depan.....	24

3.6	Perancangan Sistem Dari Samping	25
3.7	Detail Disain Aplikasi	26
3.8	Graft filter mengakses webcam	27
3.9	Tampilan Disain Program	31
3.10	Tampilan Form USB Video Divice	32
4.7.	Tampilan proram dijalankan.....	34
4.9.	Pengujian pada saat mendung.....	36
4.10.	Pengujian pada saat siang hari.....	36
4.11.	Pengujian pada saat sore hari.....	37
4.12.	Pengajuan program terhadap jarak 1 meter	38
4.13.	Pengujian program terhadap jarak 1,5 meter	38
4.14.	Pengujian program terhadap jarak 2 meter	39
4.15.	Kegagalan pendeteksian plat nomor pada jarak 4 meter	40
4.16.	Pengujian program dengan warna mobil silver	41
4.17.	Pengujian program dengan warna mobil hitam	41
4.18.	Pengujian program dengan warna mobil merah	42
4.19.	Pengujian program dengan warna mbil abu-abu	42
4.20.	Intensitas cahaya yang berlebihan yang tidak bisa terdeteksi oleh program Menggunakan webcam Okaya MD-907	43

DAFTAR TABEL

3.1. Daftar class interface yang ada di dalam Delphi 7	26
3.2. Form 1	28
3.3 Form ARI ITN	28
4.4. Spesifikasi Perlengkapan Implementasi.....	33
4.6. Pengujian Respon Program Terhadap kondisi yang diamati	35
4.7 Pengujian Respon Program Terhadap Jarak	37
4.15 Pengujian pendeteksian plat nomor mobil terhadap Background	40
4.16 Pengujian tabel keseluruhan	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di dalam era modern seperti saat ini penggunaan komputer merupakan hal yang umum dan bahkan dapat dikatakan menjadi kebutuhan vital bagi kalangan pelajar maupun orang dewasa yang menempuh jenjang karir. Hal tersebut wajar terjadi karena keunggulan komputer yang dapat sangat diandalkan untuk membantu memudahkan pekerjaan manusia yang bermacam-macam jenisnya.

Image Processing adalah suatu jenis teknologi yang digunakan untuk menyelesaikan masalah mengenai pemrosesan gambar. Pada *Image Processing* gambar yang kita peroleh akan kita proses sedemikian rupa sehingga gambar tersebut akan lebih mudah kita proses sesuai kemauan kita .

Kamera atau *webcam* merupakan salah satu perangkat komputer yang dewasa ini sering dipakai dalam *image processing*. *Webcam* ini digunakan untuk merekam atau mencuplik gambar sesuai apa yang kita inginkan, sehingga *wabcam* merupakan salah satu alat yang wajib untuk bisa menyelesaikan masalah pada *Image Processing*.

Banyak *webcam* pada gedung-gedung hanya digunakan untuk mengawasi suatu ruangan, namun *webcam* juga dapat dimanfaatkan untuk mendeteksi kebakaran. Sehingga dapat digunakan untuk menggantikan alat pengaman kebakaran yang banyak digunakan sekarang, yaitu dengan mendeteksi panas atau asap menggunakan suatu sensor yang membutuhkan alat yang rumit dan biaya yang mahal.

Dalam tugas akhir ini *webcam* dipakai sebagai sensor yang berguna untuk mendeteksi nomor plat mobil yang akan memasuki tempat parkir, skripsi ini merupakan pengembangan dari skripsi sebelumnya, dalam skripsi sebelumnya pendeteksi nomor plat mobil dimulai dengan pengambilan plat nomor secara manual yaitu pengambilan gambar menggunakan kamera digital dan difoto sendiri oleh pembuat program dan mengeditnya terlebih dahulu dengan software kemudian dicrop sampai tepat pada tepi plat nomor, sehingga tidak ada proses citra digital penyeleksi posisi plat nomor. Sedangkan pada skripsi ini dilakukan pemrosesan citra digital penyeleksi posisi plat nomor untuk mengetahui posisi plat nomor kemudian menampilkannya Untuk itu diperlukan teknologi *image processing*, sehingga *webcam* dapat berperan menjadi alat bantu mendeteksi nomor plat mobil. Hasil akhir dari tugas

akhir ini adalah membuat suatu aplikasi pengembangan sistem *pendeteksi plat nomor* mobil di tempat parkir menggunakan *webcam*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan secara jelas diatas, maka rumusan masalah ditekankan pada:

- Bagaimana cara mendeteksi nomor plat mobil dan menampilkannya.

1.3 Tujuan Penelitian

Tujuan penyusunan skripsi ini adalah merancang dan membangun suatu aplikasi yang dapat mendeteksi nomor plat mobil menggunakan webcam untuk dikembangkan lagi pada penelitian selanjutnya yaitu mengubah hasil scan nomor plat mobil ke abjad.

1.4 Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam pembuatan program ini antara lain:

1. Pengenalan ini hanya untuk plat nomor wilayah Indonesia saja. Karakter pada setiap plat nomor hanya sesuai dengan standart kepolisian indonesia.
2. Program ini digunakan hanya pada plat mobil yang warna plat dasar hitam dan warna karakter putih.
3. Posisi kamera tepat berada didepan mobil.
4. Program ini hanya sampai pada proses mendeteksi nomor plat mobil dari hasil segmentasi tidak sampai mengubah hasil segmentasi ke abjad.
5. Program ini tidak terhubung dengan *Database*.

1.5 Metode Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem di mana nantinya akan digunakan sebagai acuan perancangan sistem.

3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan kedalam system.

4. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6 Sistematika Penulisan

Sistematika penulisan laporan skripsi ini adalah sebagai berikut :

BAB I Pendahuluan

Menjelaskan latar belakang, rumusan masalah, ruang lingkup, tujuan dan sistematika penulisan skripsi

BAB II Landasan Teori

Menjelaskan kajian pustaka dan dasar teori yang digunakan

BAB III Analisa dan Disain Sistem

Menjelaskan metode yang digunakan dalam pengerjaan skripsi

BAB IV Implementasi dan Pengujian Sistem

Menjelaskan langkah langkah Perancangan Aplikasi Pendeteksi Plat Nomor Mobil beserta hasil implementasinya

BAB V Kesimpulan dan Saran

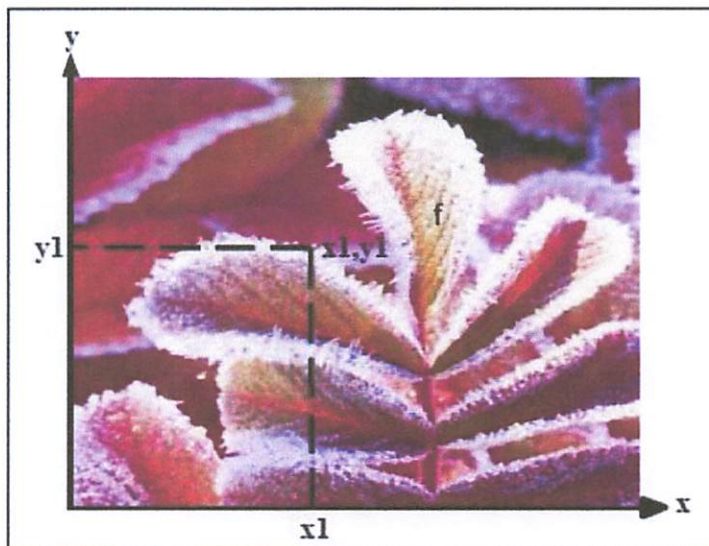
Menjelaskan langkah-langkah pengujian dari sistem yang telah dibuat dan membahas hasil pengujiannya.

BAB II

LANDASAN TEORI

2.1 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada gambar 2.1. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue* - RGB).

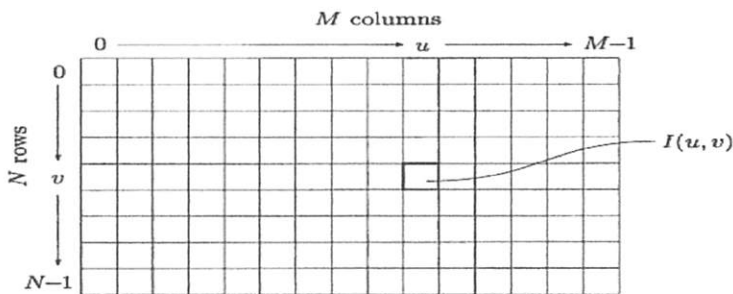


Gambar 2.1 Representasi citra digital [1]

2.2 Pixel Dan Resolusi Citra

Satuan atau bagian terkecil dari suatu citra disebut pixel (picture element) yang berarti elemen citra. Umumnya citra dibentuk dari kotak-kotak persegi empat yang teratur sehingga jarak horizontal dan vertikal antar pixel adalah sama pada seluruh bagian citra. Dalam komputer, setiap pixel diwakili oleh dua buah bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat (integer) untuk menunjukkan cahaya atau keadaan terang-gelap pixel tersebut. Untuk menunjukkan lokasi suatu pixel, koordinat $(0,0)$ digunakan untuk posisi kanan atas dalam bidang citra, dan koordinat $(m-1,n-1)$ digunakan untuk posisi kanan bawah dalam citra berukuran $m \times n$ pixel. Untuk menunjukkan tingkat pecahayaan suatu pixel, seringkali digunakan bilangan bulat

yang besarnya 8-bit, dengan lebar selang nilai 00255, dimana 0 untuk warna hitam, 255 untuk warna putih dan tingkat abu-abu berada di antara nilai 0 dan 255. seperti yang diilustrasikan pada gambar dibawah ini:



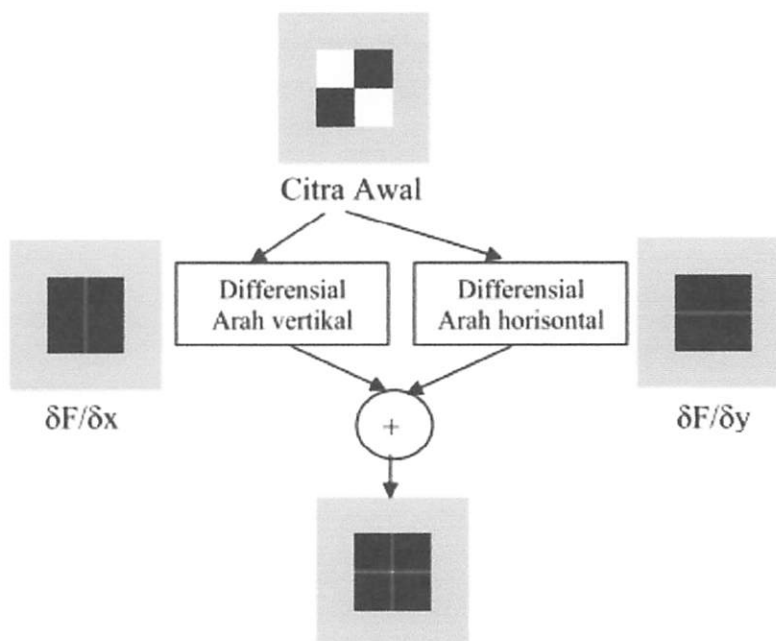
Gambar 2.2 Image size and resolution [2]

2.3 Edge Detection (Deteksi Tepi) [3]

Deteksi tepi (Edge Detection) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah :

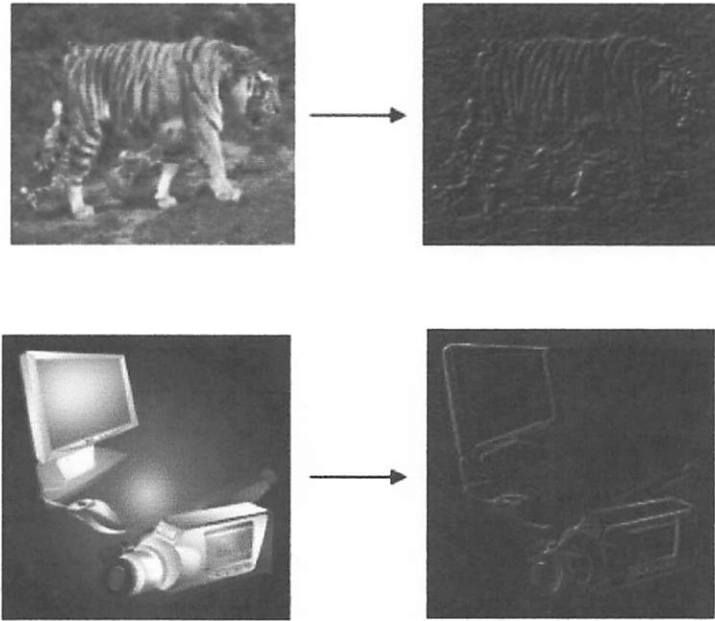
- Untuk menandai bagian yang menjadi detail citra
- Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra

Suatu titik (x,y) dikatakan sebagai tepi (edge) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Gambar 8.1 berikut ini menggambarkan bagaimana tepi suatu gambar diperoleh.



Gambar 2.3. Proses Deteksi Tepi Citra [3]

Perhatikan hasil deteksi dari beberapa citra menggunakan model differensial di atas:



Gambar 2.4. Hasil beberapa deteksi tepi [3]

Pada gambar diatas terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi

2.4 Plat Nomor Mobil

Tanda Nomor Kendaraan Bermotor (TNKB), atau sering kali disebut plat nomor atau nomor polisi (nopol), adalah plat aluminium tandakendaraan bermotor di Indonesia yang telah didaftarkan pada Kantor Bersama Samsat. Penggunaan tanda nomor kendaraan bermotor di Indonesia, terutama di Jawa, merupakan warisan sejak zaman Hindia Belanda, yang menggunakan kode wilayah berdasarkan pembagian wilayah karesidenan.

Tanda Nomor Kendaraan Bermotor berbentuk plat aluminium dengan cetakan tulisan dua baris.

1. Baris pertama menunjukkan: kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf).
2. Baris kedua menunjukkan bulan dan tahun masa berlaku.

Bahan baku TNKB adalah aluminium dengan ketebalan 1 mm. Ukuran TNKB untuk kendaraan bermotor roda 2 dan roda 3 adalah 250x105 mm, sedangkan untuk kendaraan bermotor roda 4 atau lebih adalah 395x135 mm. Terdapat cetakan garis lurus pembatas lebar 5 mm di antara ruang nomor polisi dengan ruang angka masa berlaku. Pada sudut

kanan atas dan sudut kiri bawah terdapat tanda khusus (*security mark*) cetakan lambang Polisi Lalu Lintas; sedangkan pada sisi sebelah kanan dan sisi sebelah kiri ada tanda khusus cetakan "DITLANTAS POLRI" (Direktorat Lalu Lintas Kepolisian RI) yang merupakan hak paten pembuatan TNKB oleh Polri dan TNI.

Korps Lantas Mabes Polri terhitung mulai April 2011 ini mengganti desain pelat nomor kendaraan. Ukurannya lebih panjang 5 centimeter daripada pelat nomor sebelumnya. Perubahan ukuran pelat dilakukan karena ada penambahan menjadi tiga huruf di belakang nomor, sementara sebelumnya hanya dua huruf. Perubahan ini membuat angka dan huruf pada pelat nomor berdesakan, sehingga sulit dibaca. Dengan diperpanjangnya pelat tersebut, jarak antara nomor dan huruf pada pelat lebih luas sehingga mudah terbaca.

Selain itu, perbedaan lainnya terdapat pada tampilan. Pelat TNKB baru memiliki lis putih di sekeliling pelat. Antara nomor TNKB dengan masa berlaku TNKB, tidak diberi pembatas lis putih. Namun seperti pelat nomor lama, di pelat ada 2 baris yakni baris pertama yang menunjukkan kode wilayah kendaraan, nomor polisi dan kode seri akhir wilayah. Baris kedua menunjukkan masa berlaku pelat nomor. Ukuran TNKB untuk kendaraan roda 2 dan 3 sekarang menjadi 275 mm dengan lebar 110 mm, sedangkan untuk kendaraan roda 4 atau lebih adalah panjang 430 mm dengan lebar 135 mm. Sementara ini, pelat resmi yang lama masih berlaku. [3]

2.5 Jenis-Jenis Plat Nomor Mobil

1. Plat Nomor warna dasar hitam dengan tulisan berwarna putih



Gambar 2.5. Plat nomor pribadi [4]

Plat nomor kendaraan bermotor bukan umum dan kendaraan bermotor sewa atau mobil pribadi.

2. Plat Nomor warna dasar kuning dengan tulisan berwarna hitam



Gambar 2.6. Plat nomor kendaraan umum. [4]

3. Plat Nomor warna dasar merah dengan tulisan berwarna putih



Gambar 2.7. Plat nomor kendaraan pemerintah. [4]

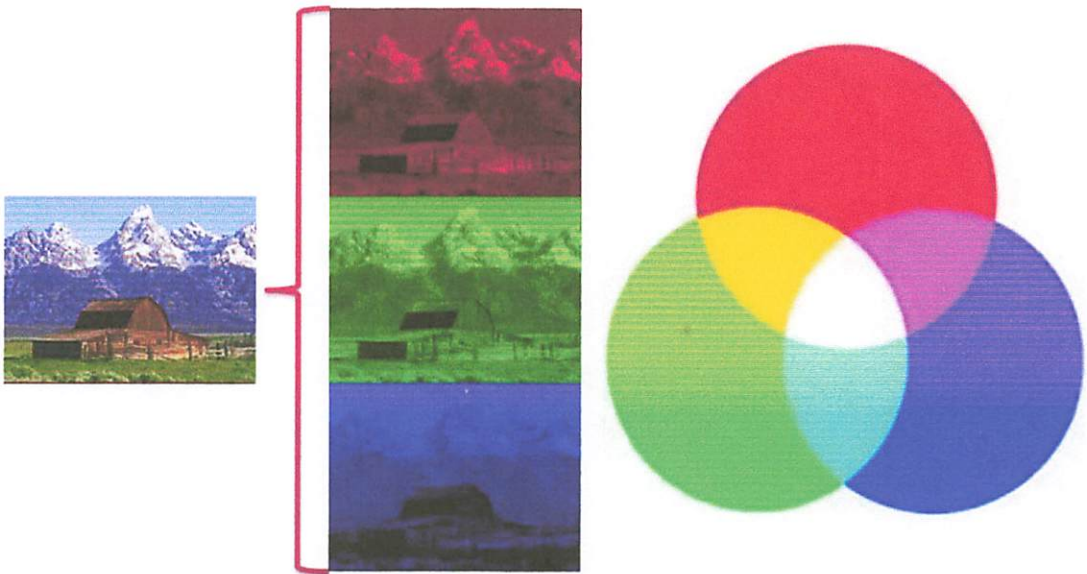
2.6 Jenis Citra

Informasi penting dari suatu citra adalah informasi mengenai intensitas cahaya yang tertangkap, ini erat hubungannya dengan warna. Suatu citra digital dapat dikelompokkan berdasarkan format intensitas penyusunnya yaitu citra biner (*monokrom*), citra skala keabuan (*grayscale*), dan citra berwarna.

2.6.1 Citra Berwarna

Citra berwarna merupakan gabungan dari beberapa lapis citra kanal warna. Teknologi dasar untuk menciptakan warna pada citra digital merupakan kombinasi dari kanal tiga warna dasar yaitu merah, hijau, dan biru (*RGB- Red, Green, Blue*) ditunjukkan pada Gambar 2.5. Kombinasi kanal warna ini didapat dari penangkapan intensitas cahaya pada frekuensi yang berbeda, yaitu frekuensi 560 nm (merah), 530 nm

(hijau) dan 430 nm (biru). Masing masing kanal mempunyai nilai yang menggambarkan intensitas setiap warna dasarnya. Prinsip dasar pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra warna direpresentasikan dengan nilai hexadecimal dari 0x00000000 sampai 0x00ffffff. Definisi nilai warna tersebut berisikan variable 0x00 yang menyatakan angka dibelakangnya adalah hexadecimal. Terlihat bahwa setiap warna mempunyai range nilai 00 (angka decimal bernilai 0) sampai ff (angka desimal bernilai 255) atau mempunyai derajat keabu-abuan $256 = 2^8$, dengan demikian range warna RGB adalah $(2^8)(2^8)(2^8) = 2^{24}$ atau dikenal dalam windows dengan istilah *true colour*. Nilai warna yang digunakan merupakan gabungan warna cahaya merah, hijau, dan biru. Jadi, untuk menyajikan warna warna tertentu dapat dengan mudah dilakukan dengan cara mencampur ketiga warna dasar RGB. Contoh format lain adalah CMY (*cyan, magenta, dan yellow*), format CMY biasa digunakan pada industri percetakan sedangkan RGB format umum yang digunakan kamera, monitor. [3]



Gambar 2.8 Citra berwarna dengan kanal setiap warna dasar penyusunnya [3]

2.6.2 Konversi Citra Berwarna (RGB) ke Citra *Grayscale*

Citra warna terdiri dari beberapa lapisan kanal warna RGB dan setiap kanal warna mewakili suatu intensitas suatu warna cahaya. Pada setiap kanal RGB dapat diubah formatnya secara ekuivalen kedalam citra skala keabuan, ditunjukkan pada Gambar 2.6. Konversi citra kanal warna dengan skala keabuan dilakukan dengan

mencocokkan *luminance* (pancaran) dari citra berwarna dengan warna abu-abu. Pada citra RGB 8 bit, masing masing kanal RGB mempunyai nilai intensitas 0 sapaai 255. Setiap intensitas mempunyai ekuivalen dengan warna abu-abu pada citra skala keabuan.

$$Y = \text{luminance (RGB)} = \text{grayscale} = (w_r \times \text{red}) + (w_g \times \text{green}) + (w_b \times \text{blue})$$

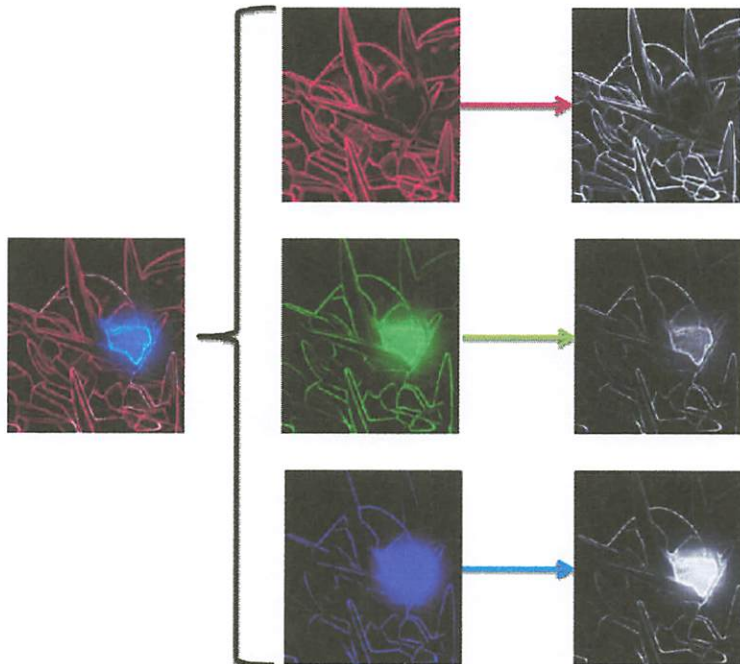
Untuk mendapatkan citra *grayscale* dari citra warna dengan bobot

$$w_r = 0,299$$

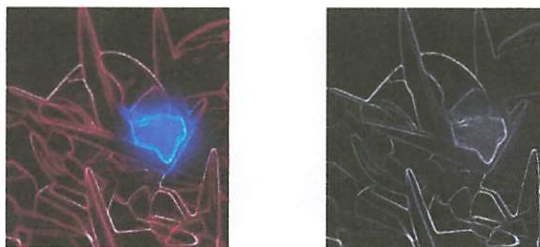
$$w_b = 0,587$$

$$w_g = 0,114$$

Setelah ketiga kanal warna mempunyai nilai skala keabuan dari hitungan matematis Persamaan diatas, hasil konversi citra berwarna ke citra *grayscale* seperti yang ditunjukkan pada gambar 2.7.



Gambar 2.9 Pemisahan Citra berwarna pada tiap kanal RGB dan luminance.



Gambar 2.10 Konversi citra berwarna menjadi citra *grayscale*. [3]

2.6.3 Konversi Citra *Grayscale* menjadi Citra Biner

Konversi citra *grayscale* menjadi citra biner dikenal dengan istilah *thresholding*, yaitu mengubah suatu citra menjadi citra hitam-putih atau biner. Proses pengubahannya dengan cara operasi nilai piksel yang memenuhi syarat ambang batas dan nilai yang dikehendaki (*threshold*). *Thresholding* merupakan proses pengelompokan warna dengan menentukan nilai ambang atau nilai batas (a_{th}) yang diterapkan untuk seluruh warna pada suatu daerah citra. Persamaan fungsi *threshold* merubah semua piksel menjadi dua keadaan dengan nilai a_0 atau a_1 seperti persamaan dibawah ini:

$$f_{threshold}(a) = \begin{cases} a_0 & \text{for } a < a_{th} \\ a_1 & \text{for } a \geq a_{th} \end{cases} \quad (2 - 2)$$

dengan $0 < a_{th} \leq a_{max}$, citra biner yang hanya memiliki dua nilai yaitu 1 atau 0 yang jika di implementasikan kedalam 8-bit nilai yang digunakan adalah 0 dan 255.

Algoritma ini umum digunakan untuk mempertegas objek citra maupun membuang noise dan gangguan yang tidak perlu. Secara umum algoritma *threshold* sederhana dengan membandingkan nilai setiap piksel pada citra dengan nilai *threshold* yang telah ditentukan. Setelah dibandingkan ditetapkan syarat berikut:

1. Jika nilai piksel yang dibandingkan dengan nilai *threshold* lebih besar, maka ganti nilai piksel dengan nilai 255 (putih).
2. Jika nilai piksel yang dibandingkan dengan nilai *threshold* lebih kecil, maka ganti nilai piksel dengan nilai 0 (hitam).

Sehingga didapat suatu citra baru yang hanya berwarna hitam atau putih saja, seperti yang diilustrasikan pada Gambar 2.8. Pemilihan nilai *threshold* juga disesuaikan dengan kebutuhan, biasanya pemilihan didasarkan pada sebaran terang-gelap pada citra originalnya. [3]



Gambar 2.11 Konversi citra *grayscale* menjadi citra biner [3]

2.7 Segmentasi Citra

Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (region) dimana setiap daerah memiliki kemiripan atribut. Salah satu teknik yang sering digunakan untuk segmentasi adalah teknik pengambangan (Thresholding).

2.7.1 Pengambangan (Thresholding)

Thresholding merupakan proses pengelompokan warna dengan menentukan nilai ambang atau nilai batas (T) yang diterapkan untuk seluruh warna pada suatu daerah citra. Persamaan fungsi threshold merubah semua pixel menjadi dua keadaan seperti persamaan dibawah ini:

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

Dengan $g(x,y)$ adalah citra biner dari citra grayscale $f(x,y)$, dan T menyatakan nilai ambang. Nilai T memegang peranan penting dalam proses pengambangan. Kualitas hasil citra biner sangat bergantung pada nilai T yang digunakan.

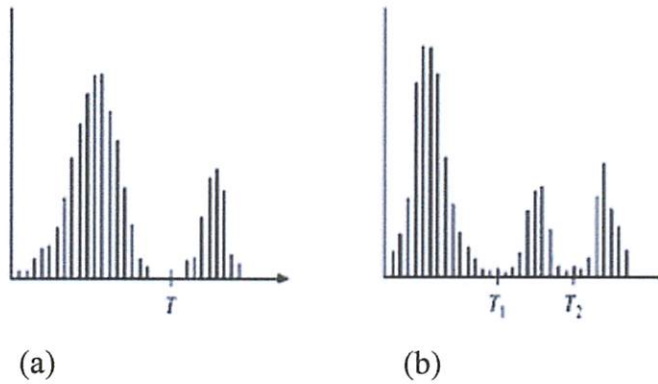
Terdapat dua jenis pengambangan, yaitu pengambangan global (global thresholding) dan pengambangan secara local adaptif (locally adaptive thresholding). Pada pengambangan global, seluruh pixel pada citra dikonversikan menjadi hitam atau putih dengan satu nilai ambang T . Kemungkinan besar pada pengambangan global akan banyak informasi yang hilang karena hanya menggunakan satu nilai T untuk keseluruhan pixel. Untuk mengatasi masalah ini dapat digunakan pengambangan secara local adaptif. Pada pengambangan local, suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan pengambangan local pada setiap blok dengan nilai T yang berbeda.

Berikut adalah salah satu contoh metode yang digunakan untuk menentukan nilai ambang (T). [3]

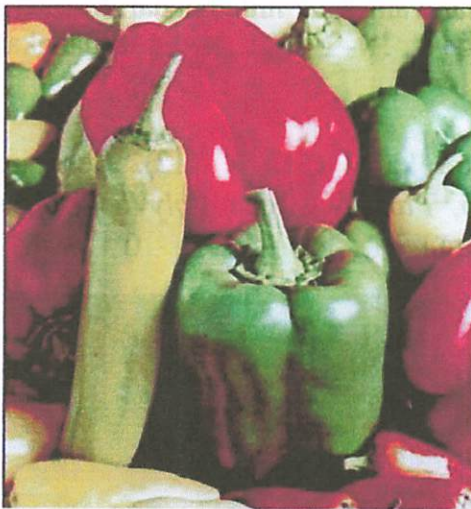
2.7.2 Metode Histogram

Metode histogram adalah metode yang paling sering digunakan. Nilai T ditentukan berdasarkan histogram dari citra yang akan diambangkan. Suatu citra yang memiliki objek tunggal dengan latar belakang homogen biasanya memiliki histogram yang bimodal (memiliki dua maksimum local atau dua puncak), seperti ditunjukkan

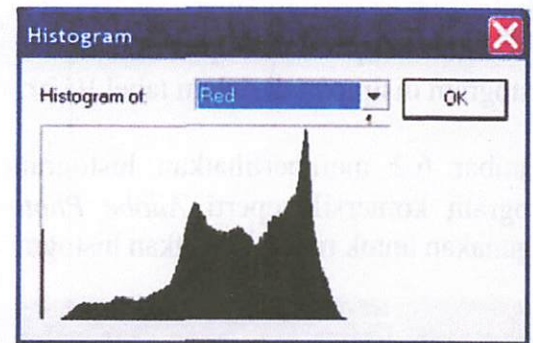
pada gambar 2.4. Citra yang memiliki histogram seperti pada gambar 2.4(a) dapat diambangkan dengan nilai ambang tunggal. Sedangkan gambar 2.4(b) menunjukkan histogram dengan 3 puncak sehingga pengambangan harus dilakukan dengan dua nilai T .



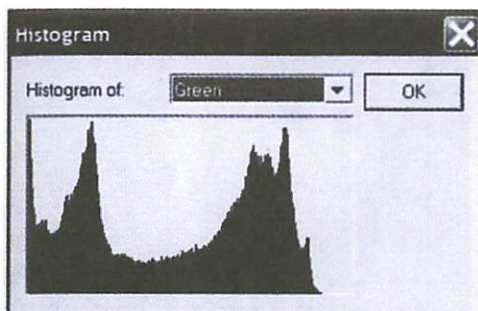
Gambar 2.12 histogram



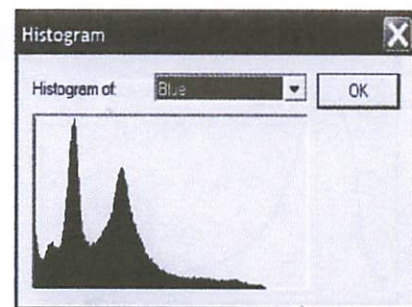
(a) *pepper (color)*, 512x512, 24-bit



(b) Histogram untuk kanal merah



(c) Histogram untuk kanal hijau



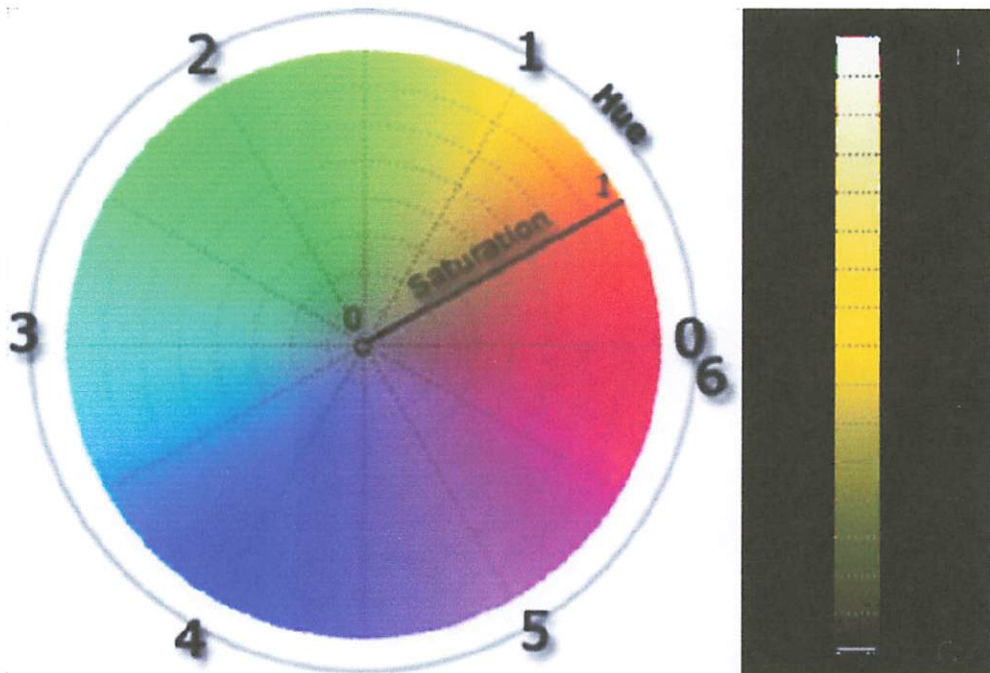
(d) Histogram untuk kanal biru

Gambar 2.13 histogram warna [5]

2.8 Color Filtering

Color Filtering adalah suatu teknik pengolahan citra yang yang dipakai untuk memanipulasi suatu citra berdasarkan warna spesifik [11]. Algoritma ini sudah banyak diimplementasikan oleh banyak orang. Salah satunya pada AForge.NET Framework [12] – [17]. Cara kerjanya adalah dengan membandingkan komponen warna setiap piksel citra dengan warna spesifik. Apabila warnanya sesuai dengan warna spesifik komponen warna piksel tersebut dibiarkan saja. Namun, bila warnanya tidak sesuai dengan warna spesifik maka komponen warna piksel tersebut diubah menjadi warna *background*, biasanya menjadi warna hitam.

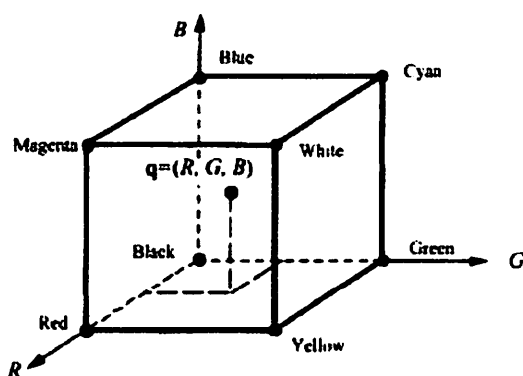
Warna yang digunakan dalam *Color Filtering* dapat direpresentasikan dalam berbagai ruang warna. Ada beberapa ruang warna yang dikenal, antara lain RGB (*Red, Green, Blue*), HSL (*Hue, Saturation, Luminance*), YCbCr, dsb. Dalam penelitian ini, penulis menggunakan ruang warna HSL (Gambar 1) [18]. Hal ini disebabkan HSL sangat cocok untuk mengidentifikasi warna-warna dasar, dimana warna dasar ini digunakan dalam penelitian sebagai warna identifikasi robot. Selain itu, HSL menoleransi terhadap perubahan intensitas cahaya. Inilah yang menjadi keunggulan HSL dibandingkan dengan ruang warna lainnya. Representasi suatu warna dalam ruang warna HSL juga tergolong mudah.



Gambar 2.14 Ruang warna HSL [3]

Tujuan utama dari model warna RGB untuk representasi, merasakan, dan menampilkan gambar dalam sistem elektronik, seperti televisi dan komputer, meskipun juga telah digunakan dalam fotografi konvensional. Sebelum masa elektronik, model warna RGB sudah mempunyai teori yang solid di balik itu semua yang berbasis pada persepsi manusia terhadap warna.

RGB adalah model warna yang tergantung pada perangkat yang berbeda atau memperbanyak nilai RGB yang diberikan, karena elemen warna (seperti fosfor atau pewarna) dan tanggapannya ke tingkat R, G, dan B individu bervariasi dari produsen ke produsen, atau bahkan pada perangkat yang sama dari waktu ke waktu. Jadi nilai RGB tidak mendefinisikan warna yang sama di seluruh perangkat tanpa semacam manajemen warna. Ketiga warna tersebut membentuk sistem koordinat 3 dimensi seperti pada gambar :



2.9 Webcam

Program pendeteksi nomor plat mobil ini menggunakan webcam dengan tipe OKAYA MD-907 sebagai mengambil gambar yang akan diproses. Kamera web atau kamera ramatraya (bahasa Inggris: *webcam*, singkatan dari *web* dan *camera*) adalah sebutan bagi kamera waktu-nyata (bermakna keadaan pada saat ini juga) yang gambarnya bisa dilihat melalui Waring Wera Wanua, program pengolahpesan cepat, atau aplikasi pemanggilan video. Istilah kamera ramatraya merujuk pada teknologi secara umumnya, sehingga kata ramatraya kadang-kadang diganti dengan kata lain yang memerikan pemandangan yang ditampilkan di kamera, misalnya StreetCam yang memperlihatkan pemandangan jalan. Ada juga Metrocam yang memperlihatkan pemandangan panorama kota dan perdesaan, TraffiCam yang digunakan untuk memantau keadaan jalan raya, cuaca dengan Weather Cam, bahkan keadaan gunung berapi dengan VolcanoCam. Kamera ramatraya adalah sebuah kamera video bergana

(*digital*) kecil yang dihubungkan ke komputer melalui (biasanya) colokan USB atau pun colokan COM. [7]



Gambar 2.15 Webcam tipe MD-907

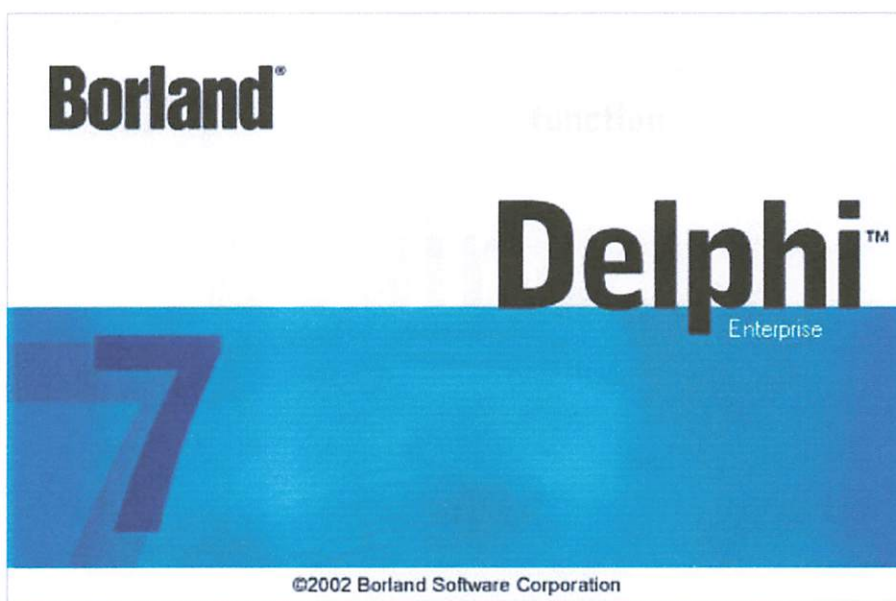
2.9.1 Cara Kerja Webcam

Sebuah web camera yang sederhana terdiri dari sebuah lensa standar, dipasang di sebuah papan sirkuit untuk menangkap sinyal gambar; casing (cover), termasuk casing depan dan casing samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di casing depan yang berguna untuk memasukkan gambar; kabel *support*, yang dibuat dari bahan yang fleksibel, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki connector, kabel ini dikontrol untuk menyesuaikan ketinggian, arah dan sudut pandang web camera. Sebuah web camera biasanya dilengkapi dengan software, software ini mengambil gambar-gambar dari kamera digital secara terus menerus ataupun dalam interval waktu tertentu dan menyiarkannya melalui koneksi internet. Ada beberapa metode penyiaran, metode yang paling umum adalah software mengubah gambar ke dalam bentuk file JPEG dan menguploadnya ke web server menggunakan *File Transfer Protocol* (FTP).

Frame rate mengindikasikan jumlah gambar sebuah software dapat ambil dan transfer dalam satu detik. Untuk streaming video, dibutuhkan minimal 15 frame per second (fps) atau idealnya 30 fps. Untuk mendapatkan frame rate yang tinggi, dibutuhkan koneksi internet yang tinggi kecepatannya. Sebuah web camera tidak harus selalu terhubung dengan komputer, ada web camera yang memiliki software webcam dan web *server built-in*, sehingga yang diperlukan hanyalah koneksi internet. Web camera seperti ini dinamakan “network camera”. Kita juga bisa menghindari penggunaan kabel dengan menggunakan hubungan radio, koneksi Ethernet ataupun WiFi. [3]

2.10 Delphi 7

Borland Delphi 7 merupakan bahasa pemrograman berbasis Windows. Delphi 7 dapat membantu untuk membuat berbagai macam aplikasi yang berjalan di sistem operasi Windows, mulai dari sebuah program sederhana sampai dengan program yang berbasiskan client/server atau jaringan. Delphi, termasuk aplikasi yang dapat digunakan untuk mengolah teks, grafik, angka, database dan aplikasi web. Tampilan umum delphi 7 seperti gambar dibawah ini



Gambar 2.16 Tampilan umum program Delphi 7

Berikut ini sebagian kecil dari banyak kelebihan Borland Delphi 7 : Berbasis Object Oriented Programming (OOP). Setiap bagian yang ada pada program dipandang sebagai suatu object yang mempunyai sifat-sifat yang dapat diubah dan diatur. Satu file .exe. Setelah program dirancang dalam IDE (Intergrated Development

Environment) Delphi, Delphi akan mengkompilasi menjadi sebuah file executable tunggal. Program yang dibuat dapat langsung didistribusikan dan dijalankan pada komputer lain tanpa perlu menyertakan file DLL dari luar. Ini merupakan sebuah kelebihan yang sangat berarti. Borland Delphi 7 hadir bersama Borland Kylix 3 yang berbasis Linux, sehingga memungkinkan programmer untuk membuat aplikasi multi-platform. [6]

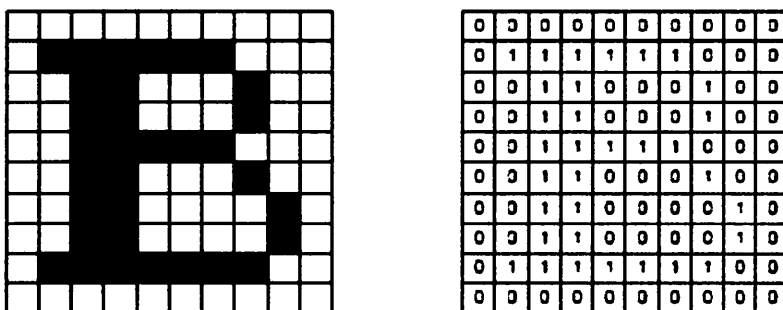
2.11 Invert Biner

Invert Image adalah suatu gambar yang terjadi adanya proses invert pada gambar asli sehingga tampak seperti gambar negative pada suatu susunan warna RGB. Citra biner, yaitu citra yang hanya terdiri atas dua warna, yaitu hitam dan putih. Oleh karena itu, setiap pixel pada citra biner cukup direpresentasikan dengan 1 bit.

Nilai pixel 0 > 1

Nilai pixel 1 > 0

Contoh citra biner adalah pada gambar 2.17 berikut.



Gambar 2.17 Invert biner [3]

BAB III

ANALISA DAN DISAIN SISTEM

3.1 Analisa Sistem

Pada bab ini dijelaskan mengenai analisa dan desain sistem aplikasi. Analisis ditujukan untuk memberikan gambaran secara umum terhadap aplikasi. Hal ini berguna untuk menunjang desain sistem aplikasi yang akan dikerjakan sehingga kebutuhan akan aplikasi tersebut dapat diketahui sebelumnya. Kemudian hasil analisa akan menjadi dasar untuk melakukan desain sistem atau perancangan aplikasi sesuai kebutuhan sistem. sistem deteksi dan pengenalan plat nomor kendaraan bermotor secara otomatis telah menjadi suatu aplikasi yang sangat penting dalam bidang *computer image processing*. Sistem ini hanya sampai pada menampilkan plat nomor mobil dan tidak mengubah karakter plat nomor kedalam bentuk abjad. Berikut diagram umum jalannya program.



Gambar 3.1 Diagram program

3.2 Data penelitian

Pada pembuatan aplikasi sistem pendeteksi nomor plat mobil yang akan di bangun, harus mempunyai data-data yang akan diolah dan digunakan pada sistem yang nantinya untuk identifikasi. Citra yang digunakan dalam sistem ini didapatkan dengan cara mengambil gambar dari hasil capture dari *webcam* dengan spesifikasi VGA 600 × 800. Ekstensi file yang dihasilkan adalah file dengan ekstensi *.jpg. dari citra tersebut dilakukan konversi kedalam format *Bitmap* (*.bmp). Dipilih format *.bmp karena format file citra tersebut tidak mengalami proses kompresi atau pemampatan, sehingga dapat langsung diolah kedalam aplikasi dan data pixel yang disimpan apa adanya.

Namun citra tersebut masih bertipe RGB (*Red Green Blue*). Citra tersebut perlu diubah bentuk tipenya menjadi citra *grayscale* hal ini dilakukan agar memudahkan dalam melakukan proses pengolahan citra pada tahap pra-proses selanjutnya. Lalu citra *grayscale* dipertajam (*Sharpening*) agar kualitas citra lebih baik. Citra plat yang telah dipertajam kemudian dicari ciri-ciri dari citra tersebut menggunakan metode deteksi tepian yaitu deteksi tepi dengan operator *Canny*. Karena metode ini memerlukan nilai variansi sebagai parameter

noise yang berfungsi untuk mereduksi noise pada citra dengan baik. Setelah pendeteksian tepi kemudian hasilnya diproses untuk mengenali pola plat nomor.

1. Pengumpulan Data Citra

Pengumpulan data citra sistem ini didapatkan dengan cara mengambil citra plat nomor dari capture *webcam* resolusi 600 ×800. Beberapa citra diambil dengan kualitas cahaya dan posisi plat nomor yang berbeda-beda karena posisi plat nomor setiap mobil tinggi rendahnya tidak sama. Contoh setiap plat nomor yang diambil gambarnya memiliki perbedaan, ada yang terlihat sudah kusam (pudar) dan ada pula angka atau huruf yang tidak sejajar (miring).

2. Proses pengolahan citra

Proses disini dilakukan dengan tujuan untuk mengolah citra yang diambil agar *noise* yang ada bisa dihilangkan semaksimal mungkin.

Tahapan-tahapan pra-proses yang akan dilakukan sebagai berikut :

- a. Pengambilan gambar diambil menggunakan webcam. Awalnya citra yang dihasilkan dari webcam adalah citra tipe RGB dan disimpang dengan format BMP.
- b. Citra tersebut perlu diubah tipenya menjadi citra greyscale untuk mempermudah proses pengolahan citra pada tahap selanjutnya.
- c. Ekstrasi ciri
 Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keberadaan tepi (*edge*) dari objek di dalam citra. Dalam tahapan proses ini, menggunakan deteksi tepi *Sobel*.

3.3 Penentuan Spesifikasi Alat

Untuk menentukan perangkat yang digunakan untuk menunjang pembuatan aplikasi pendeteksian kebakaran, antara lain:

a) Perangkat Keras:

1. Laptop Compaq Presario B1200, Intel Core 2 duo, DDR II 2048MB, Hardisk 120GB.
2. Satu buah webcam
 - a. Okaya MD-907

b) Perangkat Lunak:

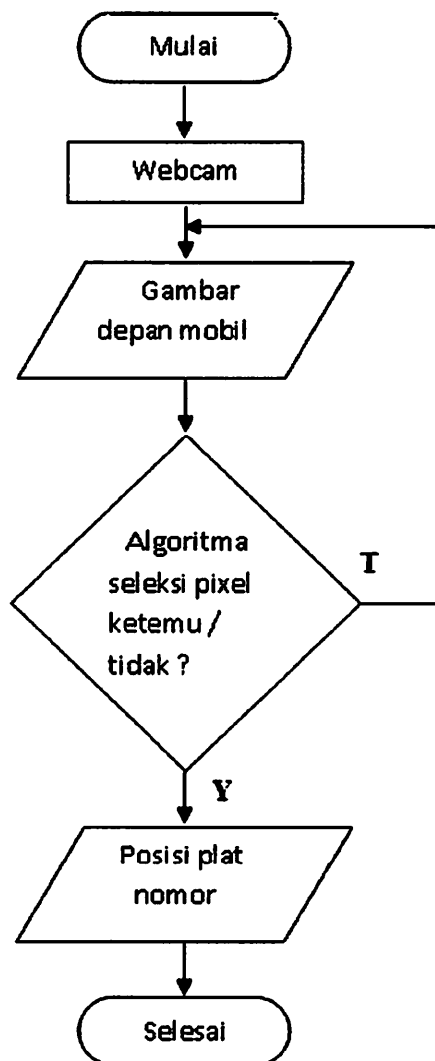
1. OS Windows XP
2. Delphi 7

3.4 Desain Sistem

Perancangan aplikasi global merupakan tahap awal sebagai acuan dalam perancangan aplikasi yang akan dibuat. Perancangan ini didahului dengan pendefinisian kegiatan pelaku atau *user* dalam menggunakan program deteksi plat nomor mobil melalui karakter huruf dan angka dengan pengolahan citra serta perangkat yang digunakan meliputi blok sistem diagram dan cara kerja aplikasi.

3.5 Perancangan dan Implementasi Sistem

Sistem pendeteksian nomor plat mobil ini terdiri dari beberapa komponen yang dapat digambarkan dengan model seperti pada gambar 3.1.

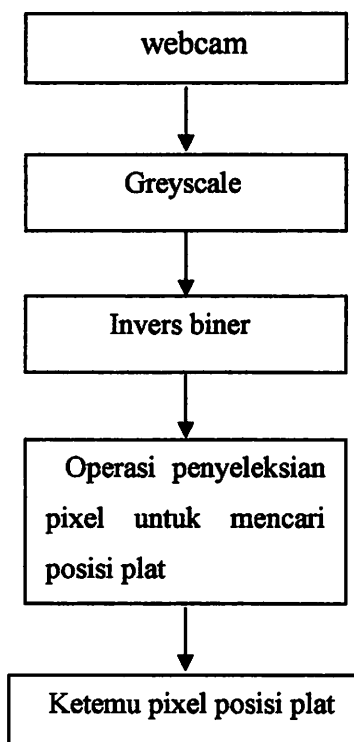


Gambar 3.2 Flowchart sistem

Flowchart sistem diatas menjelaskan dari gambar yang di capture oleh webcam yaitu gambar depan mobil untuk diseleksi pixel mencari posisi plat, jika ditemukan maka langsung berlanjut ke proses berikutnya, jika tidak maka langsung kembali ke proses capture.

3.6 Blok Diagram Jalannya Program

Jalannya program secara garis besar yang didesain untuk aplikasi deteksi nomor plat mobil melalui pengenalan karakter huruf dan angka terdiri dari beberapa langkah. Berikut ini blok diagram pendeteksi nomor plat mobil melalui pengenalan karakter huruf dan angka pada plat mobil, ditunjukkan dalam Gambar 3.3 dibawah ini :



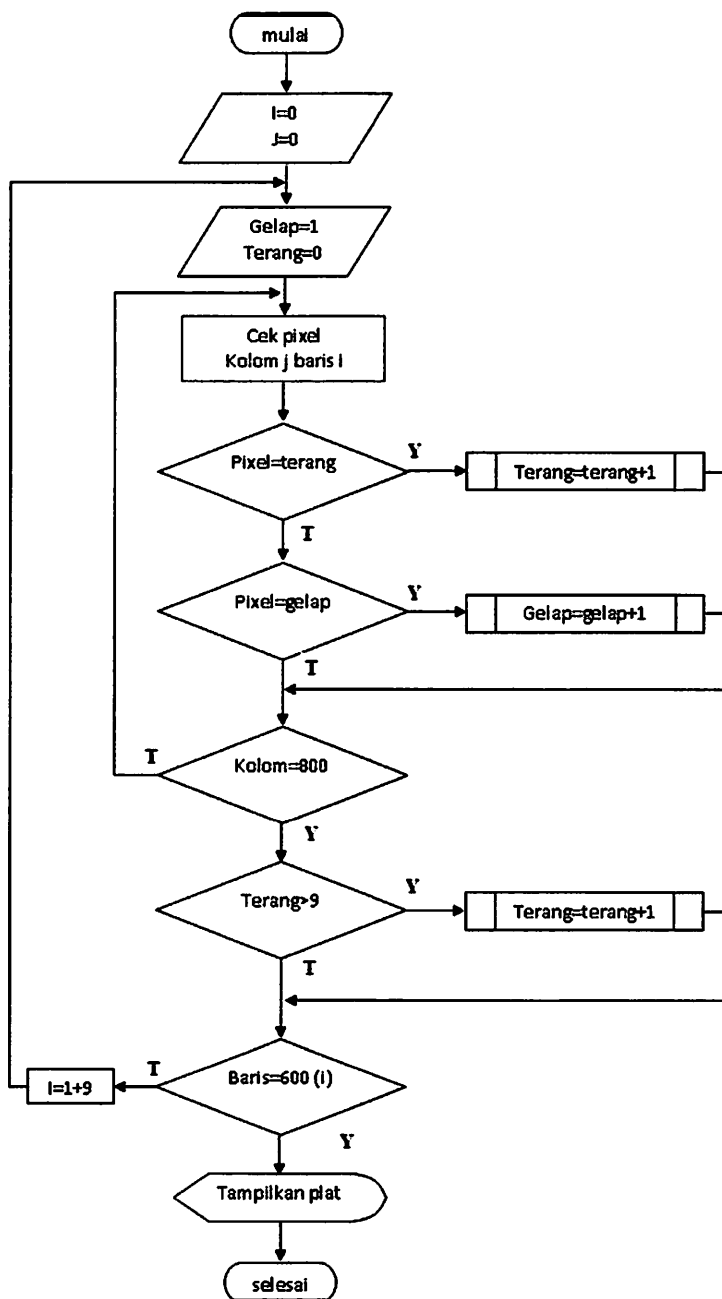
Gambar 3.3 Diagram jalannya program

Fungsi masing-masing bagian dalam diagram blok ini adalah sebagai berikut:

1. Dalam sistem ini piranti masukan yang dipakai adalah *webcam* yang digunakan untuk meng-*capture* citra.
2. *Greyscale* yaitu pengolahan citra digital dengan cara merubah warna RGB pada hasil capture webcam di rubah ke dalam bentuk skala keabuan.
3. *Invers biner* adalah proses pengolahan citra dari dari greyscale menjadi citra hitam putih dimana putih dalam biner disini 1 dan warna hitam dalam biner disini adalah 0.
4. Operasi penyeleksian pixel disini mencari posisi plat nomor dari hasil citra yang di

- 0.
4. Operasi penyeleksian pixel disini mencari posisi plat nomor dari hasil citra yang di tangkap dengan cara menscane baris-baris dari pixel citra sampai ketemu yang menjadi ciri bnetuk plat nomor.
5. Posisi pixel plat nomor ditemukan.

Berikut ini *flowcart* operasi penyeleasi pixel yang terdapat dalam diaram blok jalannya program diatas.



Gambar 3.4 Flowcart seleksi pixel

Flowchart ini menggambarkan bagaimana proses seleksi pixel untuk mencari posisi plat nomor dalam citra.

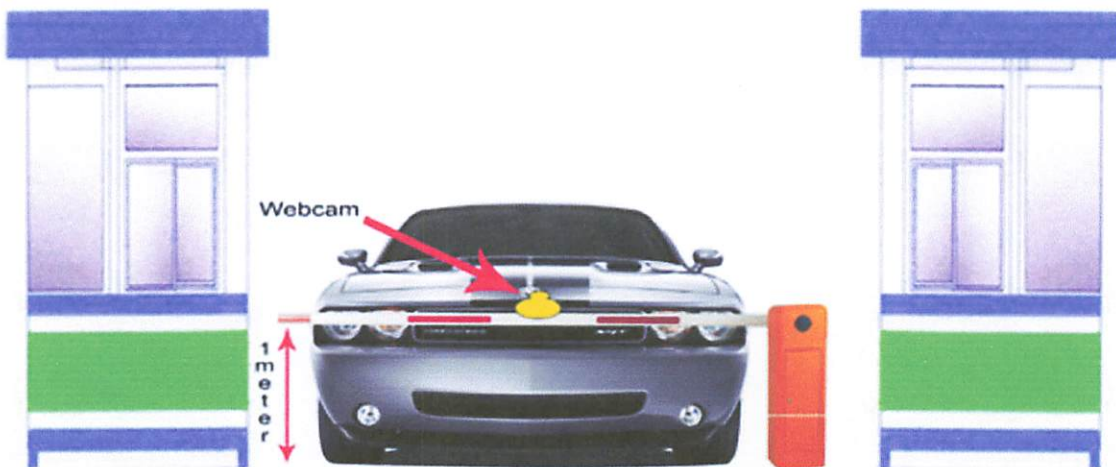
3.7 Aplikasi Pendeteksi Plat Nomor Mobil

3.7.1 Konfigurasi Perangkat Keras

Pengolahan citra ini dapat bekerja dengan baik bila citra memiliki karakteristik tertentu. Citra yang diolah merupakan citra plat nomor yang memenuhi parameter dari program, sehingga dapat terdeteksi. Untuk mendapatkan citra yang sesuai dengan keinginan, diperlukan konfigurasi *webcam* yang tepat pada lapangan tempat sistem diimplementasikan.

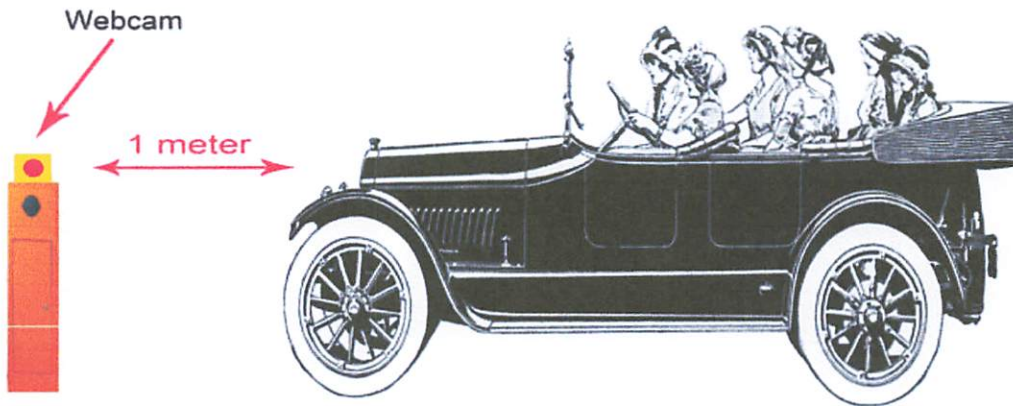
Gambaran umum konfigurasi perancangan sistem aplikasi berikut in

Depan



Gambar 3.5 Perancangan Sistem Dari Depan

Samping



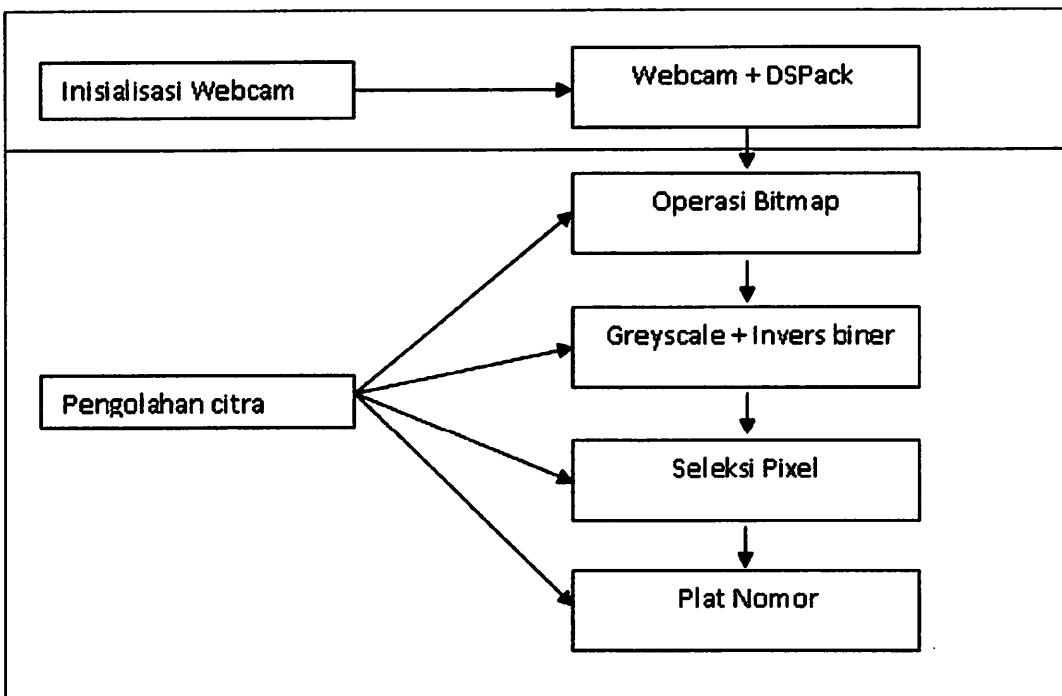
Gambar 3.6 Perancangan Sistem Dari Samping

3.7.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dibangun dengan menggunakan bahasa pemrograman Borland Delphi 7. Sistem ini dirancang dengan spesifikasi sebagai berikut:

1. Mengakses perangkat keras *webcam*.
2. Mencari posisi plat nomor dengan deteksi tepi.
3. Melakukan proses invers biner .
4. Melakukan deteksi penyeleksian pixel-pixel plat nomor.
5. Menampilkan hasil pengenalan karakter dikomputer.

Proses pembuatan program dibagi menjadi 2 bagian yaitu inialisasi *webcam*, dan pengolahan citra. Detail desain aplikasi secara umum ditunjukkan pada Gambar 3.



Gambar 3.7 Detail desain aplikasi

3.7.3 Perancangan Inisialisasi Webcam

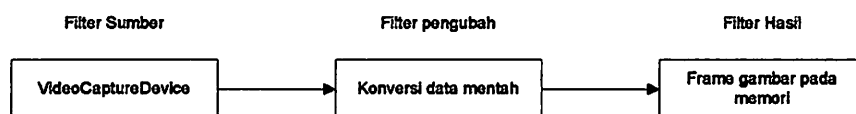
Perancangan inisialisasi *webcam* dilakukan memanfaatkan beberapa fitur komponen Delphi 7 *DSPack* sebagai antar muka perangkat keras. *DSPack* adalah komponen untuk dapat mengakses perangkat keras berbasis *Windows* seperti *webcam*. Berikut ini adalah pengelompokan class yang ada di dalam *DSPack* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar class interface yang ada di dalam Delphi 7

No	Class	Keterangan
1	FilterGraph	Sumber video untuk file video
2	VideoWindow	Jendela video
3	SampleGrabber	Informasi Filter
4	Filter	Koleksi objek informasi filter
	ASFWriter	Kemampuan perangkat video seperti ukuran

5	ASFWriter	Kemampuan perangkat video seperti ukuran frame dan frame rate
6	DSTackBar	Sumber video untuk pengambilan gambar video pada perangkat local (<i>webcam</i>)
7	DSVideoWindowEx2	Pengaturan video

Pada perancangan inialisasi *webcam* ini, DSPack mengerjakan tugas dengan cara menghubungkan serangkaian filter bersama-sama, sehingga keluaran dari satu filter menjadi masukan untuk filter lainnya. Kumpulan filter-filter yang terhubung membentuk graf disebut graf filter, seperti ditunjukkan pada Gambar 4.4 .



Gambar 3.8 Graf filter mengakses *webcam* [5]

3.7.4 Perancangan Program

Pada perancangan pengolahan citra terdiri atas beberapa bagian yaitu operasi *bitmap*, filter warna (seleksi piksel pada tiap kanal RGB) ,deteksi gerak,deteksi tepi dan pengenalan karakter huruf dan angka . Operasi *bitmap* digunakan untuk mengakses secara langsung piksel beserta data informasi yang terletak di memori. Kemudian proses filter warna yaitu proses pencarian piksel pada setiap kanal RGB, dimana kita harus mengetahui range dari plat nomor untuk kita masukkan pada program, sehingga apabila terdapat objek yang ciri-ciri pixelnya tidak seperti plat nomor maka akan diabaikan. Program ini terdiri dari dua form pada Delphi 7, yaitu Form 1 dan Form ARI ITN. Form 1 sebagai lembarkerja tampilan awal dan formARI ITN adalah link dari Form 1 yang berisi tombol Bitbtn1, Bitbtn2, Bitbtn3, Bitbtn4 yang merupakan button proses jalannya program.

Berikut ini komponen-komponen yang di gunakan dalam aplikasi pada Tabel 4.3 berikut ini :

1. Form1

No	Komponen	Keterangan	Properties	Events
1	Button (BitBtn 1)	Sumber video untuk file video	<ul style="list-style-type: none"> • Caption = BitBtn1 • Left = 800 	<ul style="list-style-type: none"> • OnClick = BitBtn1Click
2	Button (BitBtn 2)	Kategori <i>Direct Show</i> filter	<ul style="list-style-type: none"> • Caption = BitBtn2 • Left = 800 	<ul style="list-style-type: none"> • OnClick = BitBtn2Click
3	Button (BitBtn 3)	Informasi Filter <i>Direct Show</i>	<ul style="list-style-type: none"> • Caption = BitBtn 3 • Left = 800 	<ul style="list-style-type: none"> • OnClick = BitBtn3Click
4	Button (BitBtn 4)	Koleksi objek informasi filter	<ul style="list-style-type: none"> • Caption = BitBtn4 • Left = 800 	<ul style="list-style-type: none"> • OnClick = BitBtn4Click

2. Form ARI ITN pada Tabel 4.4 berikut ini :

No	Komponen	Keterangan	Properties	Events
1	Filter (DSPack)	Sumber video untuk file video	Name = Filter	FilterGraph = FilterGraph
2	FilterGraph (DSPack)	Kategori <i>Direct Show</i> filter	GraphEdit = True LinearVolume = True Mode = Capture	

3	SampleGrabber (DSPack)	Informasi Filter <i>Direct Show</i>	FilterGraph = FilterGraph Name = SampleGrabber	FilterGraph = FilterGraph OnBuffer = SampleGrabber Buffer
4	MainMenu	Koleksi objek informasi filter	Name = MainMenu1	
5	OpenDialog (Dialogs)	Kemampuan perangkat video seperti ukuran frame dan frame rate	Name = Opend	
6	VideoWindow (DSPack)	Sumber foto untuk pengambilan gambar pada perangkat local (<i>webcam</i>)	Align = alClient Color = clBlack FilterGraph= FilterGraph Height = 258 Width = 258	
7	Image (Additional)	Untuk menampilkan hasil dari <i>capture</i> masih dalam bentuk gambar asli.	Align = alClient Left = 4 Top = 4 Width = 258	
8	Image (Additional)	Untuk menmpilkan hasil pengolahan citra digital dari gambar asli menjadi gambar	Align = alClient Name = img1 Center = True Height = 258 Left = 4 Top = 4	

		<i>invers biner</i>	Width = 301	
9	ComboBox (StdCtrls)	Mengkoneksikan <i>webcam</i>	Name= ComboBox1 Height = 21 ItemHeight = 13 Left = 8 Text=Pilih Camera Top = 344 Width = 145	OnChange= ComboBox1Change
10	Image9 (Additional)	Menampilakn hasil plat nomor yang sudah terdeteksi	Name = Image9 Align = alClient Height = 117 Left = 2 Strech = True Top = 2 Width = 717	
11	Button4 (Std)	Tombol untuk menjalankan proses keseluruhan mulai dari capture plat sampai menampilkan hasilnya	Name = Button1 Caption= Capture + Scan Plat Height = 25 Left = 160 TabOrder = 6 Top = 342 Width = 113	OnClick= Button4Click

12	Button3 (Std)	Tombol keluar dari aplikasi	Name = Button3 Caption = EXIT Height = 49 Left = 752 TabOrder = 8 Top = 448 Width = 121	OnClick= Button3Click

3. Disain aplikasi berikut ini :

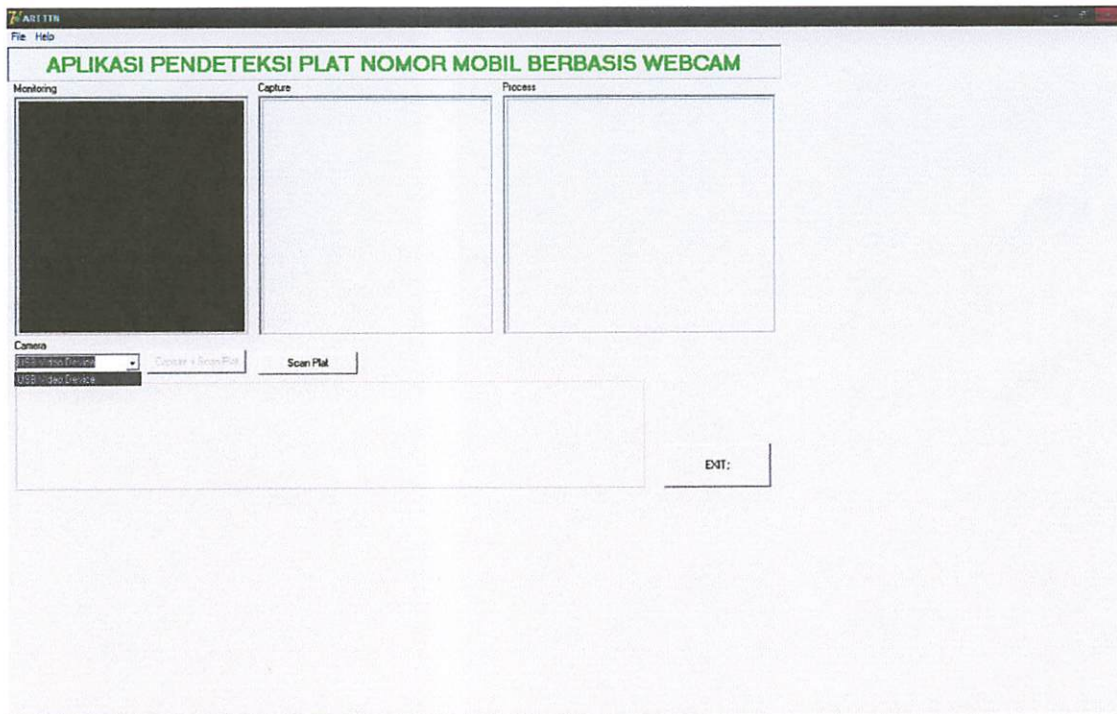
The design shows a window with three main panels at the top: "Monitoring dari WEBCAM", "Hasil Capture WEBCAM", and "Hasil Invers Biner WEBCAM". Below these panels are three buttons: "Combo box", "Tombol Capture +", and "Tombol Scan plat". At the bottom left is a large rectangular area labeled "Tampilan Plat Nomor". At the bottom right is a button labeled "Tombol Exit".

Gambar 3.9 Tampilan disain program

3.8 Implementasi Interface *Open Local Capture Device*

Program deteksi plat nomor menggunakan webcam mempunyai tampilan seperti pada Gambar. bentuk interface atau antarmuka didesain sesederhana mungkin. Ini bertujuan agar pemakai tidak bingung dan cepat terbiasa menggunakan program ini. Untuk mengakses webcam yang terkoneksi dengan komputer dengan cara memilih menu pada ComboBox yaitu *USB Video Device*.

Langkah Pertama yang dilakukan sebelum melakukan proses pendeteksian adalah mengaktifkan *webcam* pada program ini. Proses pengaktifan dengan cara mengklik pada panel menu item yaitu *file* → *Open Local Device* yang akan memunculkan dialog Combobox. Pada dialog Combobox digunakan untuk memilih webcam, seperti pada gambar 4.14.



Gambar 3.10 Tampilan Form USB Video Device

BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi sistem

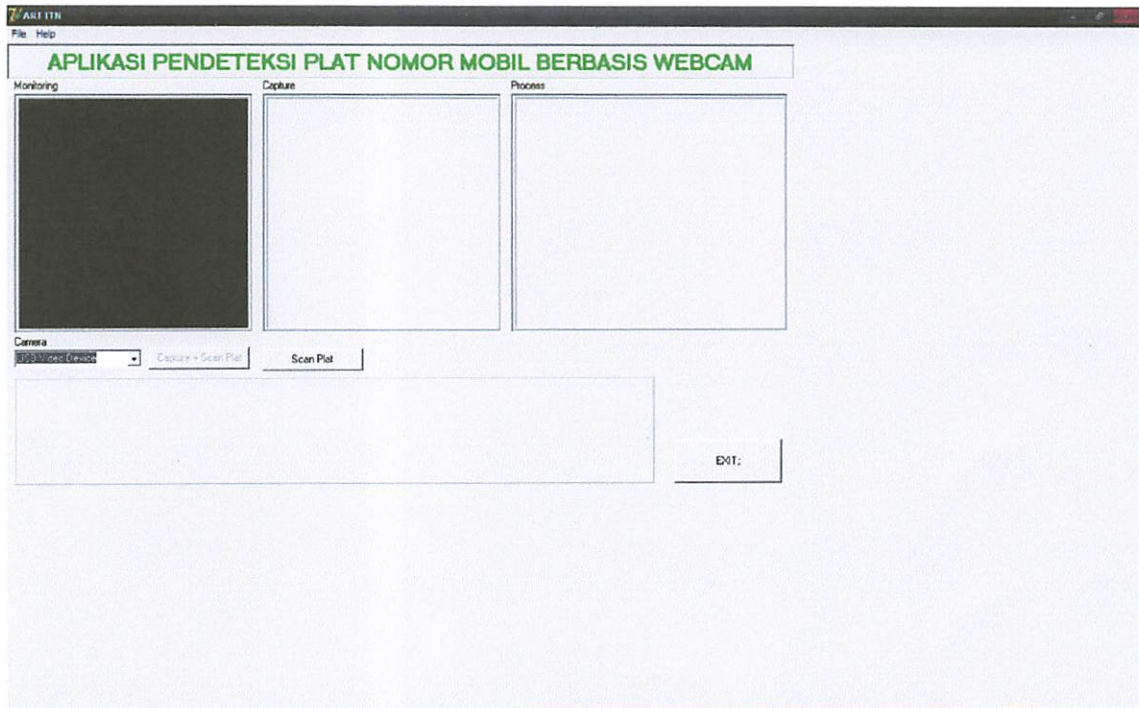
Implementasi dilakukan dengan menerapkan hasil desain yang telah dibuat kedalam Delphi 7, sehingga prosedur-prosedur yang telah dibuat dapat dimengerti oleh mesin dan menghasilkan keluaran seperti apa yang diharapkan. Berikut ini adalah perlengkapan dan kebutuhan yang digunakan dalam implementasi sistem :

Tabel 4.1 Spesifikasi Perlengkapan Implementasi

No	Perlengkapan	Spesifikasi	Keterangan
1	Software PC	Sistem Operasi	Windows XP Service Pack 2
		Bahasa Pemrograman	Borland Delphi 7
2	Personal Komputer	Processor	Intel Core 2 Duo 1,83 Ghz
		Memory	2 Gb DDR2
		Harddisk	150 GB
3	Webcam	Okaya	MD-907
		Jack Plug USB	

4.2 Implementasi Antarmuka untuk Menjalankan Aplikasi

Setelah mengaktifkan *webcam*, maka pada akan muncul satu buah camera window dan dua image window. Camera window yang sebelah kiri untuk menampilkan tampilan yang difoto oleh *webcam*. Sedangkan dua image window yang sebelah kiri untuk menampilkan hasil foto *webcam* dan *image window* sebelah kanan untuk menampilkan hasil pengalahn citra *invers biner*. Seperti ditunjukkan pada Gambar 4.7.



Gambar 4.1 Tampilan Form program dijalankan

4.3 Pengujian sistem

Untuk mengetahui apakah sistem bekerja dengan baik dan sesuai dengan perancangan, maka diperlukan serangkain pengujian. Pengujian yang dilakukan dalam bab ini adalah sebagai berikut:

1. Pengujian *webcam*.
2. Pengujian program pendeteksi nomor plat mobil.
3. Analisis faktor kegagalan.

4.4 Pengujian Pendeteksian Plat Nomor Menggunakan Webcam Terhadap Kondisi Keadaan yang diamati

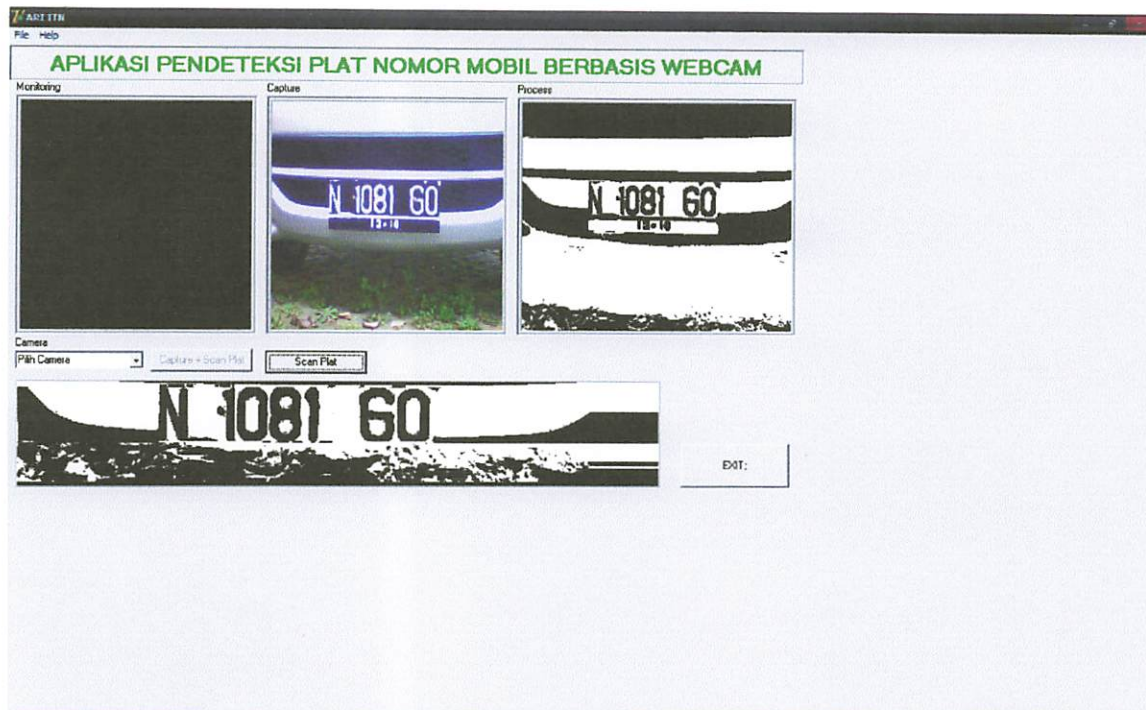
Pada pengujian ini daerah di kondisikan pada waktu yang berbeda-beda. Waktunya dibedakan mejadi 3 kondisi, tanpa penerangan kurang dimana saat cuaca mendung, pada malam hari dimana kondisinya terang (terdapat lampu), pada siang hari dan pada saat sore hari. Sedangkan jarak *webcam* terhadap objek yang diamati tetap dan objek yang diamati sama yaitu mobil ditempat parkir. Seperti pada Tabel berikut ini :

Keadaan	Respon Program dengan Webcam Okaya
Cuaca saat mendung tanpa penerangan	Terdeteksi
Siang hari tanpa penerangan	Terdeteksi
Sore hari tanpa penerangan	Terdeteksi

Tabel 4.2 Pengujian Respon Program terhadap kondisi keadaan yang diamati menggunakan webcam Okaya MD-907

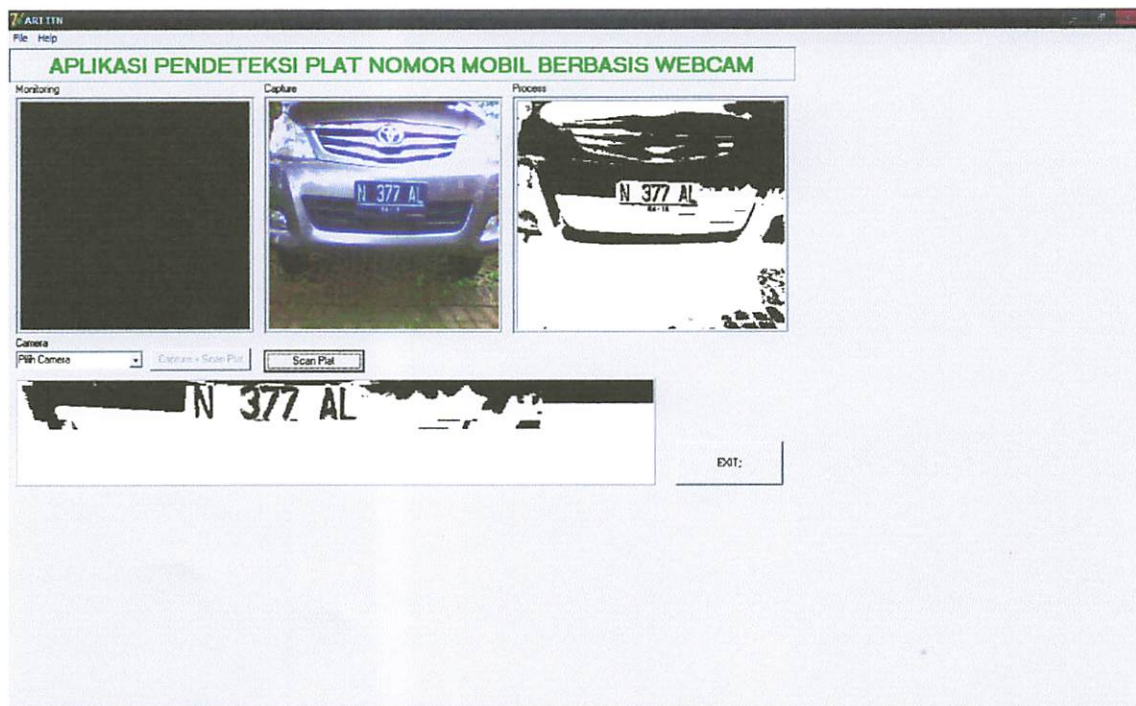
Pada tabel pengujian di atas hasil dari pengujian kondisi keadaan yang diamati, program dapat mendeteksi plat nomor pada semua intensitas penerangan dikarenakan sat ini cuaca tidak menentu mengingat faktor-faktor non teknis juga berpengaruh terhadap kinerja program, untuk mengimbangi kondisi non teknis diperlukan penyesuaian dan perubahan hardware yang diperlukan misalnya penataan tempat webcam, penataan sumber cahaya, dan penataan porta pada pintu masuk parkir. Berikut ini program diuji berdasarkan kondisi cuaca. Seperti yang diujikan seperti berikut ini :

a. Cuaca saat mendung dengan penerangan :



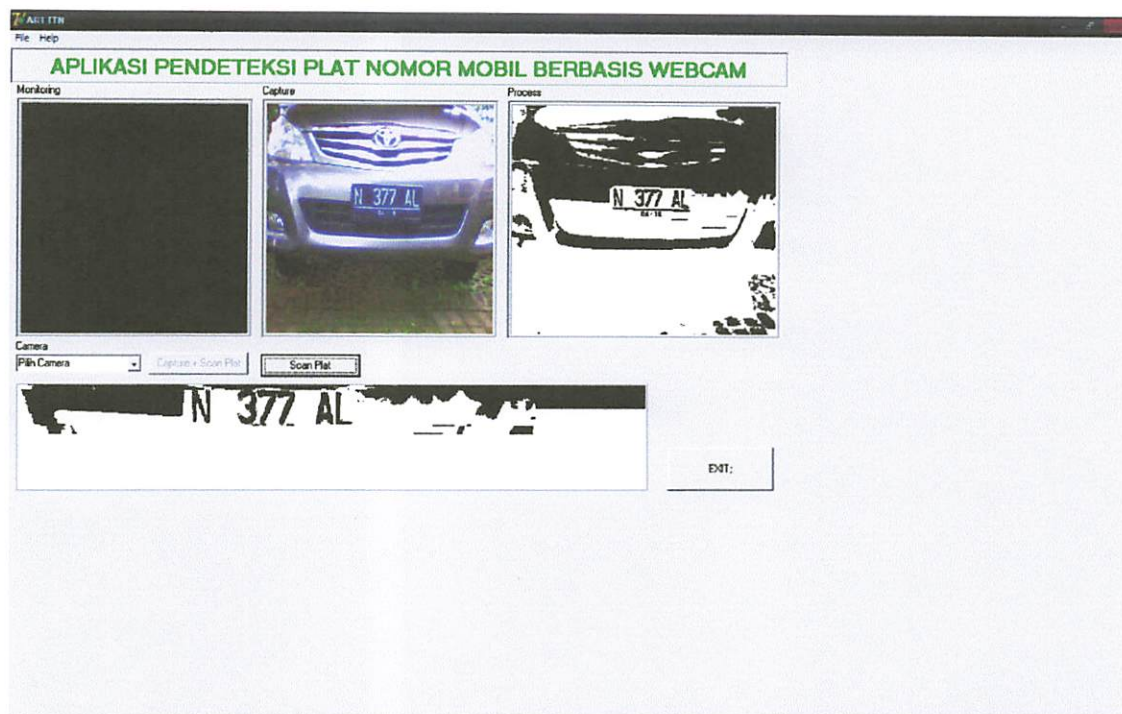
Gambar 4.2 Pengujian pada saat cuaca mendung menggunakan webcam Okaya MD-907

b. Siang hari tanpa penerangan



Gambar 4.3 Pengujian saat siang hari menggunakan webcam Okaya MD-907

c. Pengujian pada Sore hari tanpa penerangan



Gambar 4.4 Pengujian saat sore hari menggunakan webcam Okaya MD-907

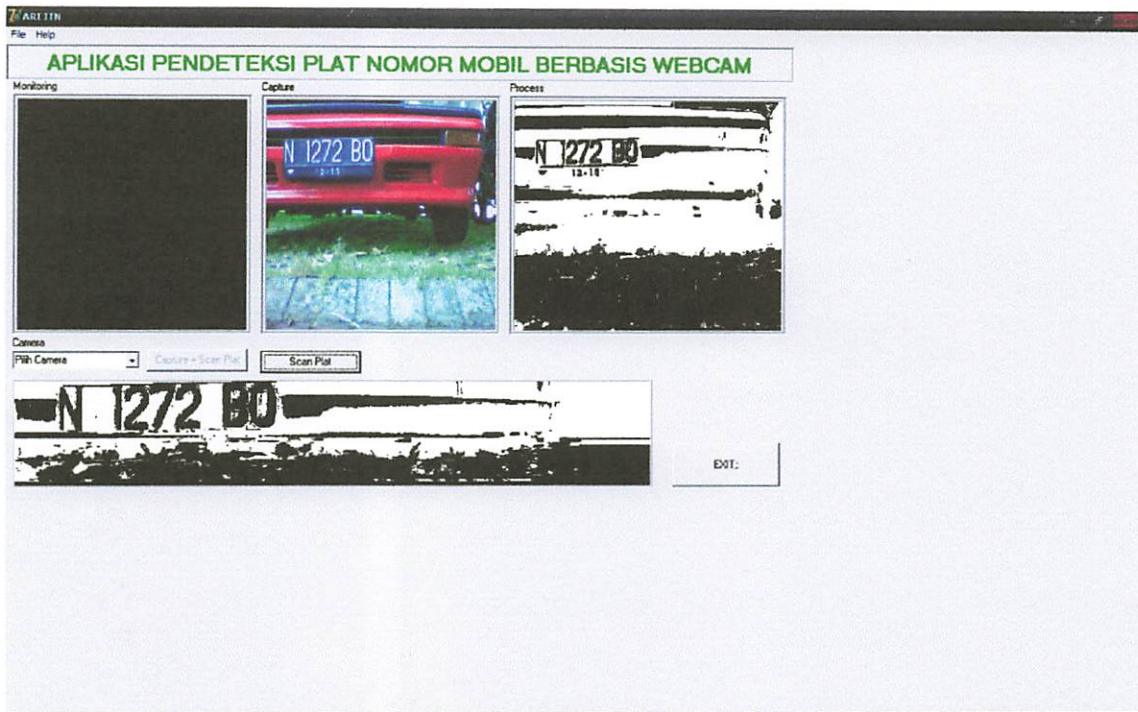
4.5 Pengujian Pendeteksian Plat Nomor Mobil Menggunakan Webcam Terhadap Jarak

Pada pengujian ini posisi *webcam* terhadap mobil di uji dengan jarak berbeda, intensitas penerangan dibuat konstan. Jarak yang digunakan mulai dari 1 meter, kemudian tiap kali pengujian ditambah setengah meter hingga 2 meter.

Tabel 4.3 Pengujian Respon Program Terhadap Jarak

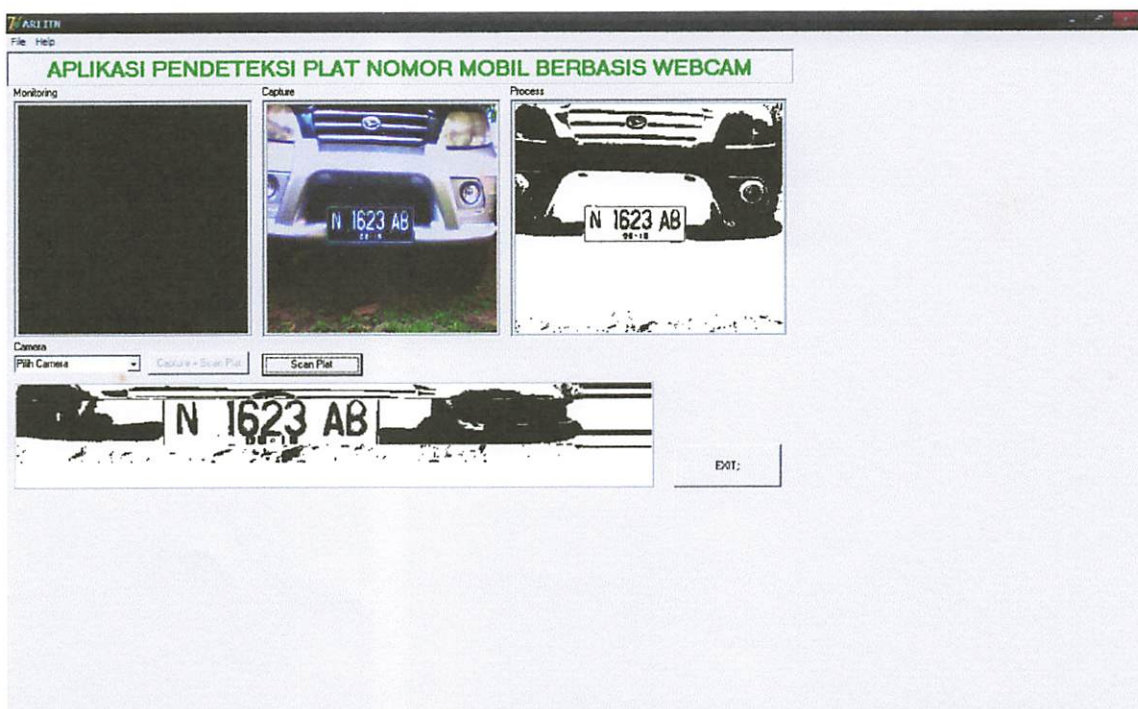
Intensitas Penerangan	Jarak dalam meter	Respon program dengan webcam
Siang hari	1	Terdeteksi
Siang hari	1,5	Terdeteksi
Siang hari	2	Terdeteksi

a. Pengujian program terhadap jarak 1 meter



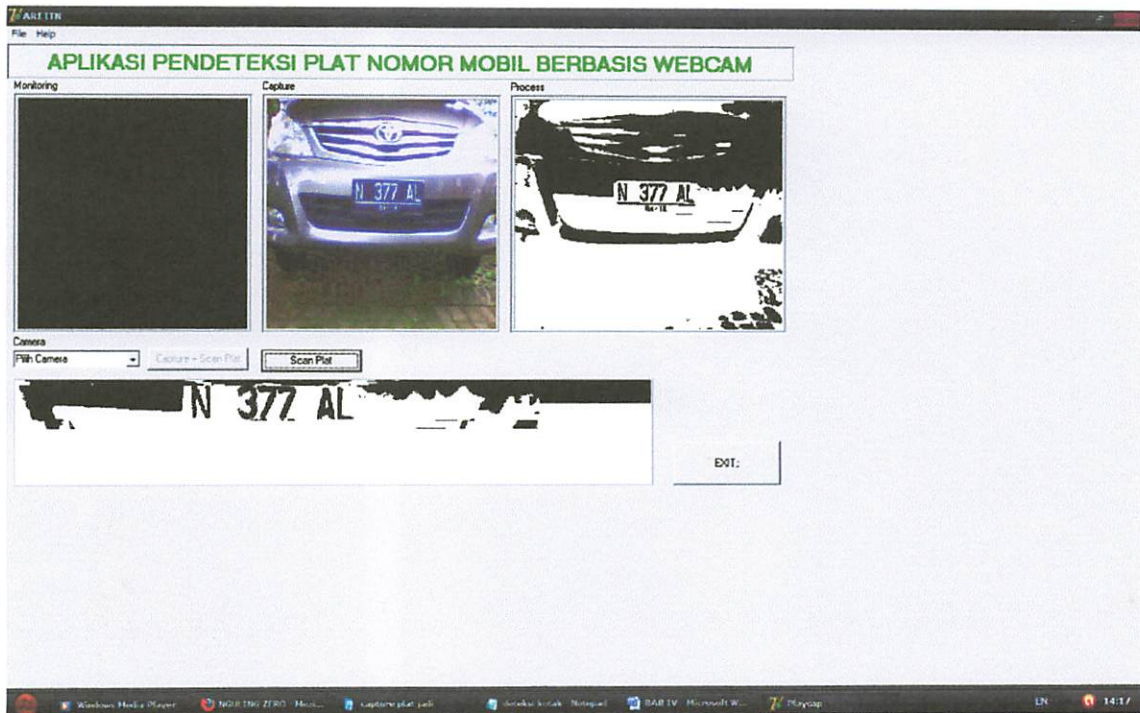
Gambar 4.5 Pengujian program terhadap jarak 1 meter menggunakan webcam
Okaya MD-907

b. Pengujian program terhadap jarak 1,5 meter



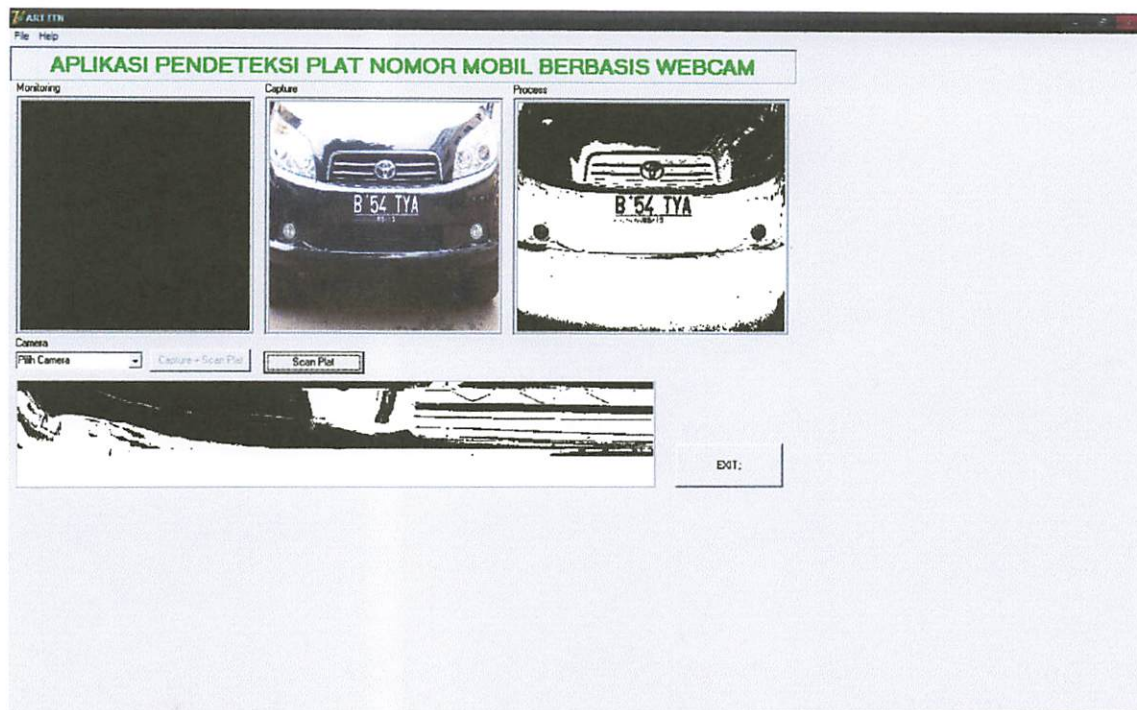
Gambar 4.6 Pengujian program terhadap jarak 1,5 meter menggunakan webcam
Okaya MD-907

c. Pengujian program terhadap jarak 2 meter



Gambar 4.7 Pengujian program terhadap jarak 2 meter menggunakan webcam
Okaya MD-907

Dari hasil pengujian diatas, program dapat mendeteksi pada jarak 1 sampai 2 meter untuk webcam jenis Okaya MD-907 denan jarak 1 sampai 2 meter dapat mendeteksi plat nomor mobil dengan webcam webcam jenis Okaya MD-907. Selain itu semakin jauh jarak dari webcam maka program tidak bisa mendeteksi plat nomor, program hanya bisa mendeteksi bentuknya, tetapi karakter plat nomornya tidak bisa terdeteksi karena semakin jauh mobil dari webcam, maka bentuk plat nomor hasil dari foto webcam terlihat kecil tidak dapat terbaca program dan program menganggap objek yang diamati tidak ada. Seperti dibawah ini :



Gambar 4.8 Kegagalan Pendeteksian plat nomor pada Jarak 4 Meter

4.6 Pengujian Pendeteksian Plat Nomor Mobil Terhadap Warna Background

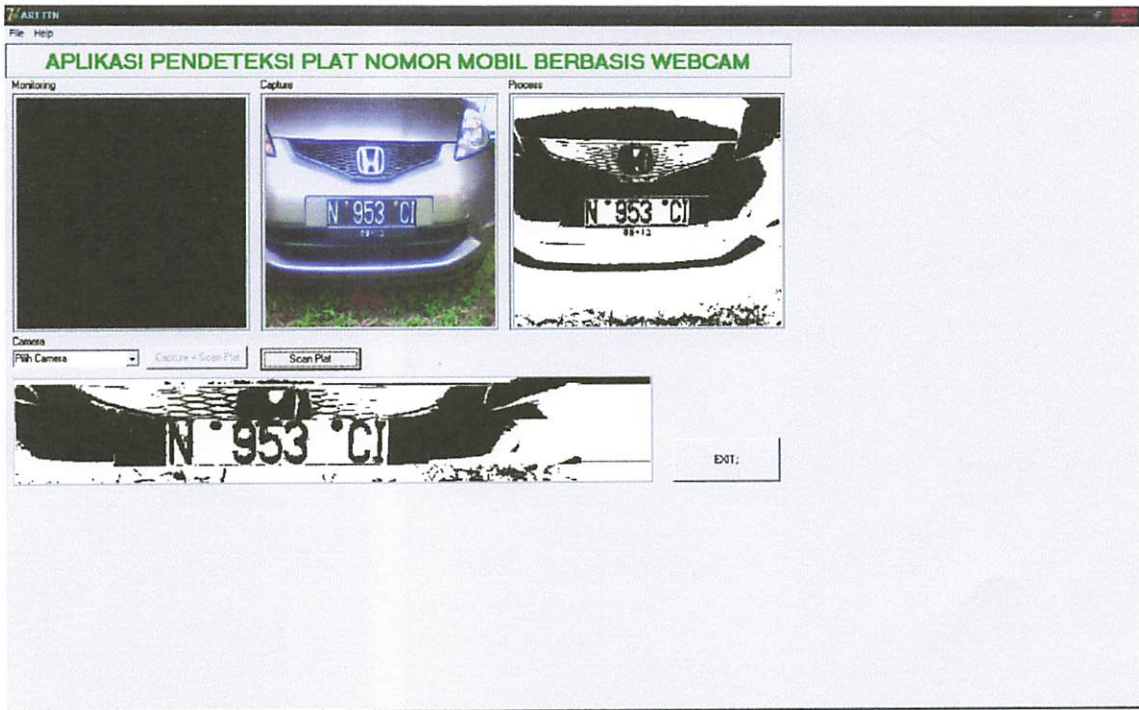
Pada pengujian ini dilakukan untuk mengetahui pengaruh background yang berbeda dimana warna body mobil terhadap proses pendeteksian pada aplikasi. Dalam pengujian ini perubahan warna *background* secara bergantian, jarak dibuat tetap dan intensitas penerangan dibuat tetap. Seperti tabel pengujian berikut ini :

Tabel 4.4 Pengujian pendeteksian plat nomor mobil terhadap Background

Warna Background	Respon Program dengan Webcam Okaya
Silver	Terdeteksi
Hitam	Terdeteksi
Merah	Terdeteksi
Abu-abu	Terdeteksi

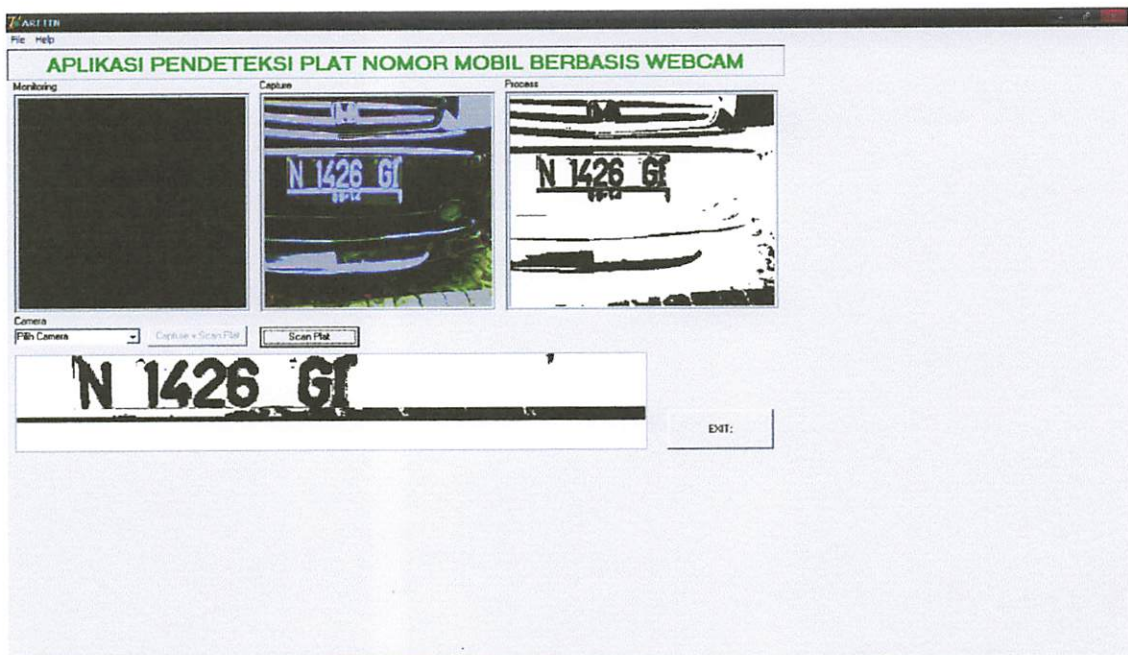
Pada pengujian ini, maka dari semua body mobil yang dicoba dengan jarak 1 meter mendapatkan hasil bahwa plat nomor terdeteksi oleh program, sehingga program dapat dikatakan berhasil berjalan sebagaimana mestinya sesuai body mobil yang telah ditentukan. seperti percobaan berikut ini :

a. Pengujian program dimana body mobil berwarna silver :



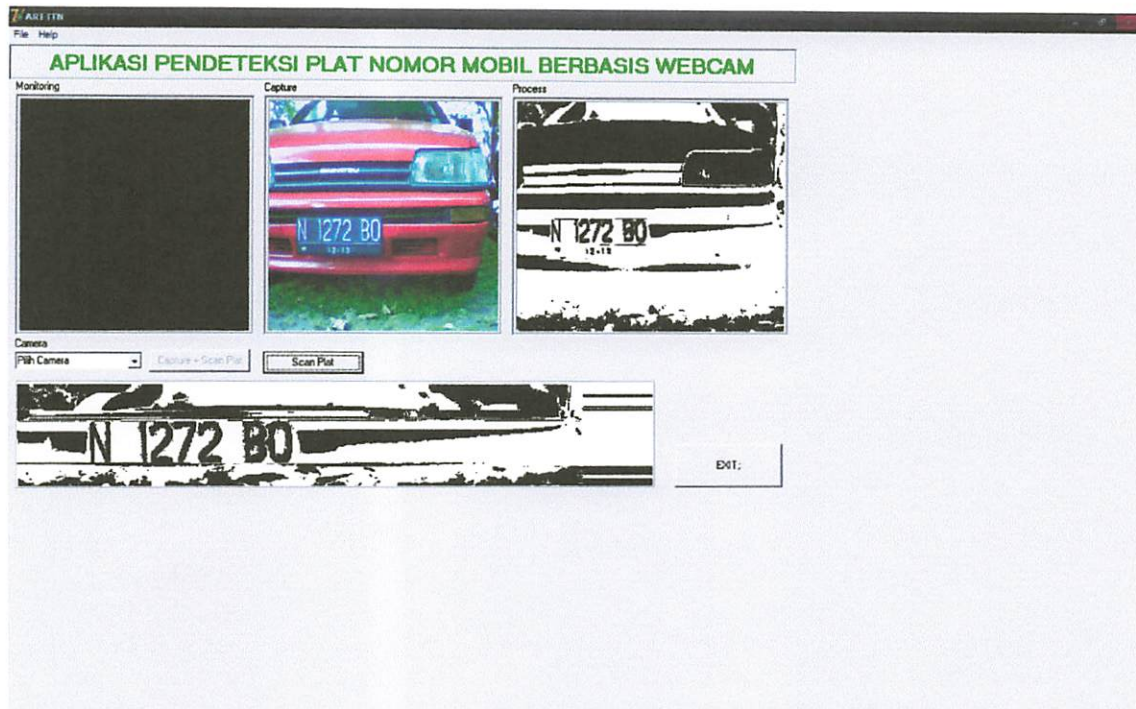
Gambar 4.9 Pengujian program dimana body mobil berwarna silver

b. Pengujian program dimana body mobil berwarna Hitam :



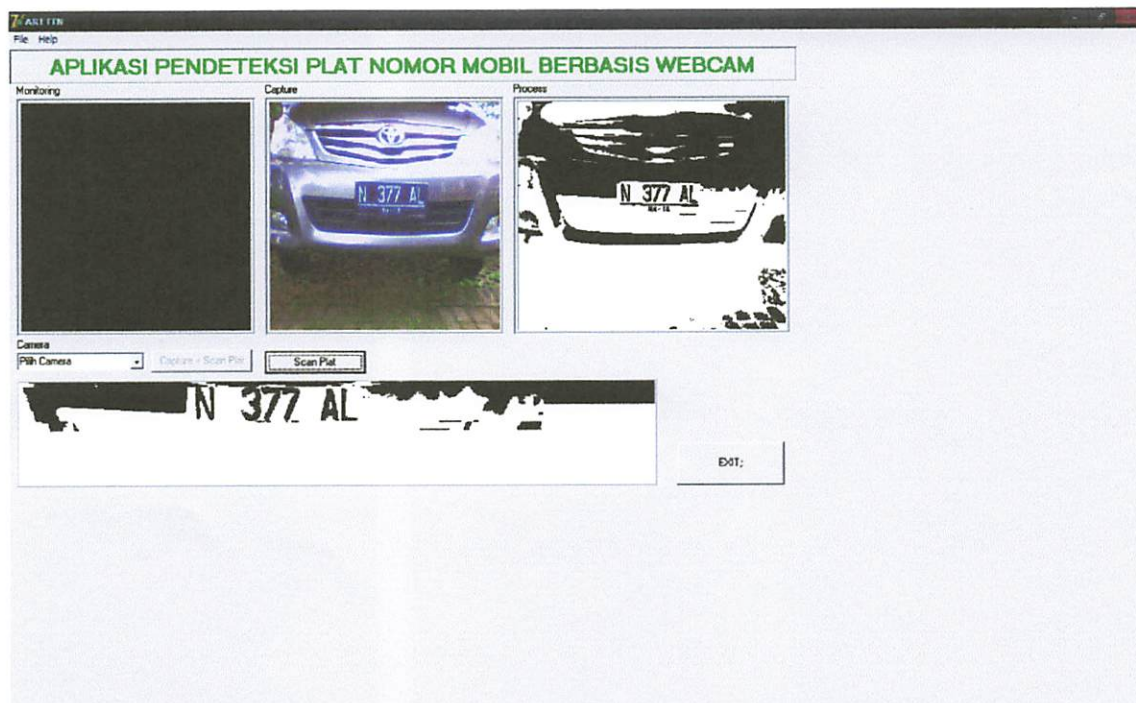
Gambar 4.10 Pengujian program dimana body mobil berwarna hitam

c. Pengujian program dimana body mobil berwarna merah :



Gambar 4.11 Pengujian program dimana body mobil berwarna merah

d. Pengujian program dimana body mobil berwarna abu-abu :



Gambar 4.12 Pengujian program dimana body mobil berwarna abu-abu

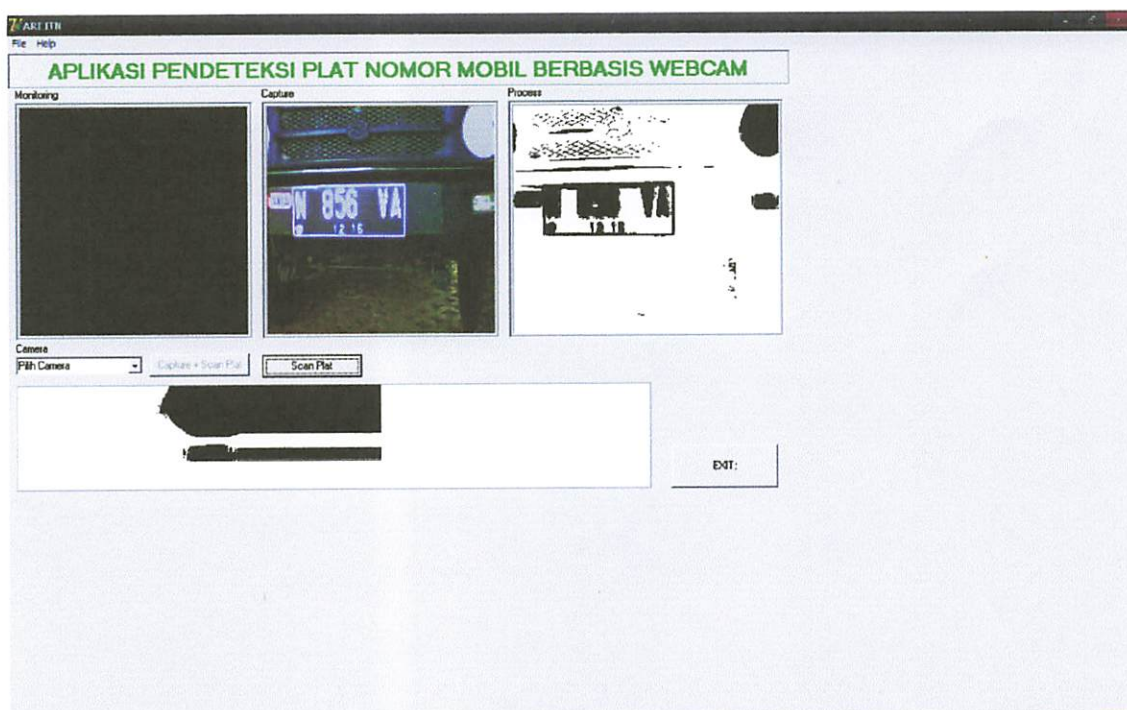
4.7 Analisis Faktor Kegagalan

Pada kenyataannya, aplikasi pendeteksian ini tidak selalu berjalan optimal. Hal ini dipengaruhi beberapa faktor baik faktor dari dalam (perangkat lunak dan perangkat keras) maupun faktor dari luar (kondisi lingkungan sekitar).

4.8 Faktor Kegagalan Pendeteksian Plat Nomor

Pada pengeksekusian program deteksi plat nomor tidak selalu didapat hasil yang optimal. Hal ini disebabkan beberapa faktor:

1. Kondisi penerangan ruangan yang kurang baik seperti intensitas cahaya terlalu rendah atau terlalu tinggi dan cahaya yang tidak konstan.
2. Pengaruh dari benda lain, bila ada cahaya kuat baik dari lampu penerangan membuat benda yang ditangkap oleh webcam menjadi terkena pantulan cahaya sehingga pendeteksian plat nomor terjadi kegagalan karena sensor *webcam* melihatnya seperti tidak ada benda yang terdeteksi. Seperti pengujian dibawah ini:



Gambar 4.13 Intensitas cahaya yang berlebihan yang tidak bisa terdeteksi menggunakan webcam Okaya MD-907

3. Kualitas sensor *webcam* yang kurang bagus yang mengakibatkan kualitas gambar hasil capture kurang bagus atau buram, sehingga program tidak bisa mengolah citra.

Dari percobaan-percobaan di atas dapat di ambil kesimpulan dalam tabel keseluruhan pengujian berikut ini :

No	Keterangan	Berhasil	Tidak berhasil
1	Cuaca saat mendung tanpa penerangan	√	
2	Cuaca saat siang hari tanpa penerangan	√	
3	Cuaca saat sore hari tanpa penerangan	√	
4	Siang hari dengan jarak 1 meter	√	
5	Siang hari dengan jarak 1,5 meter	√	
6	Siang hari dengan jarak 2 meter	√	
7	Siang hari dengan jarak 4 meter		√
8	Pengujian dengan body mobil warna silver	√	
9	Pengujian dengan body mobil warna silver	√	
10	Pengujian dengan body mobil warna silver	√	
11	Pengujian dengan body mobil warna silver	√	
12	Pengujian dengan body mobil warna silver	√	
13	Pengujian dengan internsitas cahaya		√

	berlebihan		
--	------------	--	--

Tabel 4.5 Pengujian tabel keseluruhan

Berdasarkan hasil pengujian yang telah dilakukan, memberikan kesimpulan bahwa secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan. Dari analisa tabel di diatas dapat disimpulkan bahwa tingkat keberhasilan program sistem pendeteksi plat nomor mobil menggunakan webcam dari 13 percobaan yang gagal 2 jadi $11 \times 7,69 = 84,59 \%$.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis sistem maka dapat diambil kesimpulan sebagai berikut:

1. Proses pendeteksian plat nomor mobil memakai teknik seleksi piksel warna dengan memfilter nilai pada masing masing kanal RGB. Dalam inisialisasi webcam menggunakan komponen DSPack pada Delphi 7
2. Untuk mengetahui objek yang ditangkap webcam merupakan plat nomor dengan cara mengubah gambar asli dirubah ke greyscale kemudian dilakukan pengolahan citra invers biner. Apabila tidak ditemukan titik yang memenuhi syarat ciri dari plat nomor maka program tidak mendeteksi.
3. Keberhasilan pendeteksian plat nomor menggunakan webcam menggunakan software Delphi 7 dipengaruhi oleh kondisi penerangan ruangan yang dipantau dan kualitas sensor *webcam*.
4. Aplikasi ini dapat mendeteksi nomor plat mobil ketika :
 - a. Saat waktu siang, sore, dan malam dengan penerangan yang cukup.
 - b. Jarak mobil berada tidak sampai melebihi 2 meter dari webcam.
 - c. Warna body mobil berbeda digunakan sebagai background oleh program.
 - d. Plat nomor merupakan standart kepolisian indonesia atau belum dimodifikasi.
 - e. Untuk body mobil yang berwarna putih hasil pendeteksian agak rancu, dikarenakan program kesulitan mendeteksi antara plat dan body mobil.
5. Berdasarkan tabel keseluruhan pengujian bisa dikatakan Aplikasi ini tingkat keberhasilan 90%

5.2 Saran

Dalam perancangan dan pembuatan aplikasi pendeteksian plat nomor mobil ini masih terdapat kekurangan dan kelemahan, oleh karena itu masih diperlukan adanya penyempurnaan dalam rangka pengembangan ke depan. Adapun hal yang dapat disempurnakan antara lain:

1. Mengatur intensitas penerangan ruangan sebaik mungkin dimana sistem bisa berjalan optimal.

2. Menggunakan *webcam* dengan kualitas sensor yang bagus.
3. Untuk penelitian selanjutnya agar aplikasi dilanjutkan pada tahap mengubah hasil cropping plat kedalam bentuk Abjad.

DAFTAR PUSTAKA

- [1] Pardosi,Mico,2005,"Belajar Delphi 7",Selaras,Surabaya.
- [2] Sigit, Riyanto., Basuki, Achmad., Ramadijanti, Nana., Pramadihanto, Dadet., 2005. Step by Step Pengolahan Citra Digital. Yogyakarta: Andi
- [3] Kuncara,Chandra,2004,"Belajar Pengolahan Citra Digital Tingkat Dasar.",URL: http://rois.its.ac.id/jiunkpe_dg_4414.html (diakses 14 Desember 2011)
- [4] Kuncara,Chandra,2004,"jenis-jenis plat kendaraan di indonesia.",URL: http://wijaya.wordpress_dg_4414.html (diakses 14 Desember 2011)
- [5] Kurnia,Dede,2011," Pengukuran Kepadatan Arus Lalu Lintas Menggunakan Sensor Kamera ", URL: <http://dhetiastapurwatna.wordpress.com/2009/05/19/sistem-lalu-lintas/>. (diakses tanggal 16 Desember 2011)
- [6] Suntoyo,Andi,2011," Pengambilan Gambar dari *Web Camera* untuk Membangun Sistem Informasi",URL : <http://www.andisun.com/jurnal/pengambilan-gambar-dari-web-camera-untuk-membangun-sistem-informasi-2>. (diakses tanggal 18 Desember 2011)
- [7] Anonim . Avicap32.dll . <http://www.bigresource.com/Delphi-dll--g1FiUXUdEX.html> (diakses tanggal 18 Desember 2011)
- [8] Kurniawan,Andi,2011," Pengambilan Gambar dari *Web Camera* untuk Membangun Sistem Informasi",URL : <http://www.andisun.com/jurnal/computer-vision-pengambilan-gambar-dari-web-camera-untuk-membangun-sistem-informasi-3>. (diakses tanggal 18 Desember 2011)
- [9] Suntoyo,Andi,2011," Pengambilan Gambar dari *Web Camera* untuk Membangun SistemInformasi",URL:http://www.gunadarma.ac.id/library/articles/.../Artikel_50405669.pdf
f
(diakses tanggal 18 Desember 2011)

LAMPIRAN



INSTITUT TEKNOLOGI NASIONAL

Jl. Raya Karanglo, KM 2

MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Ari Yoswara
Nim : 08.12. 909
Masa Bimbingan : 30 November 2011 s/d 30 Mei 2012
Judul Skripsi : PENGEMBANGAN SISTEM PENDETEKSI PLAT NOMOR DI TEMPAT PARKIR MENGGUNAKAN WEBCAM

No	Tanggal	Uraian	Paraf Pembimbing
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing I

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



INSTITUT TEKNOLOGI NASIONAL

Jl. Raya Karanglo, KM 2

MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Muhammad Arif
Nim : 07.12.639
Masa Bimbingan : 30 November 2011 s/d 30 Mei 2012
Judul Skripsi : PENGEMBANGAN SISTEM PENDETEKSI PLAT NOMOR DI
TEMPAT PARKIR MENGGUNAKAN WEBCAM

No	Tanggal	Uraian	Paraf Pembimbing
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing II

Ahmad Faisol, ST
NIP. Y. 0721048502



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**

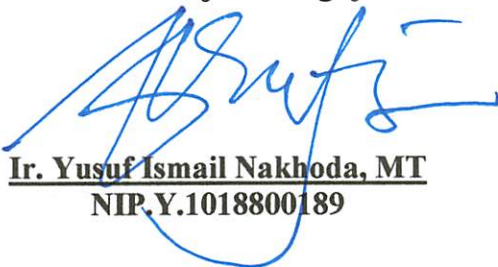
NAMA : ARI YOSWARA
NIM : 08.12.909
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN: Semester Genap Tahun Akademik 2011-2012
JUDUL : **PENGEMBANGAN SISTEM PENDETEKSI PLAT NOMOR DI
TEMPAT PARKIR MENGGUNAKAN WEBCAM**

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

Hari : Selasa
Tanggal : 31 Juli 2012
Dengan Nilai : B+ (72) *✓*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

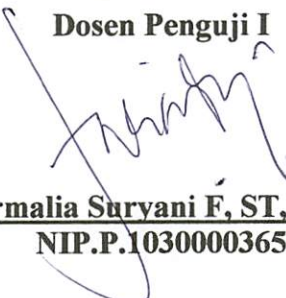

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Sekretaris Majelis Penguji



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

ANGGOTA PENGUJI

Dosen Penguji I


Irmalia Suryani F, ST, MT
NIP.P.1030000365

Dosen Penguji II


Bima Aulia Firmandani, ST
1121



FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : ARI YOSWARA
NIM : 08.12.909
PROGRAM STUDI : Teknik Komputer dan Informatika S-1
JUDUL : **PENGEMBANGAN SISTEM PENDETEKSI PLAT
NOMOR DI TEMPAT PARKIR MENGGUNAKAN
WEBCAM**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	31 Juli 2012	1. Gambar 3.2 Flowchart 2. BAB IV adalah pengujian	
2.	Penguji II	31 Juli 2012	1. Penomoran halaman 2. Format penulisan	

Disetujui :

Dosen Penguji I

Irmalia Suryani F, ST, MT
NIP.P. 1030000365

Dosen Penguji II

Bima Aulia Firmandani, ST
NIP.P. 1121

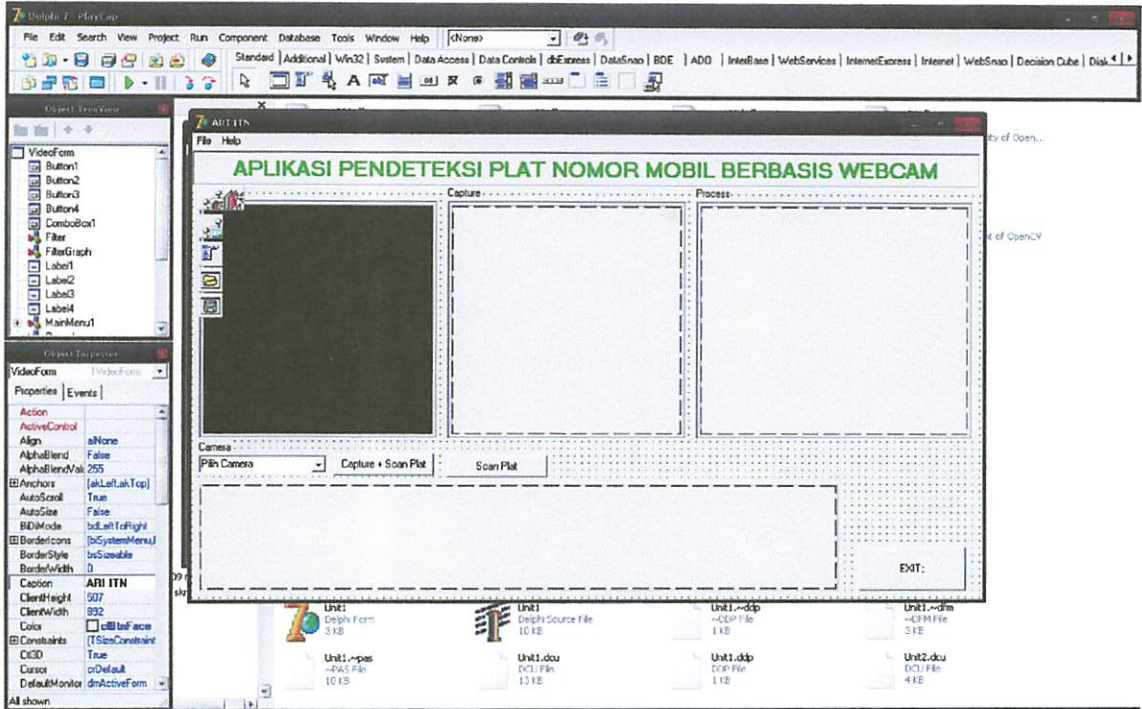
Mengetahui :

Dosen Pembimbing I

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

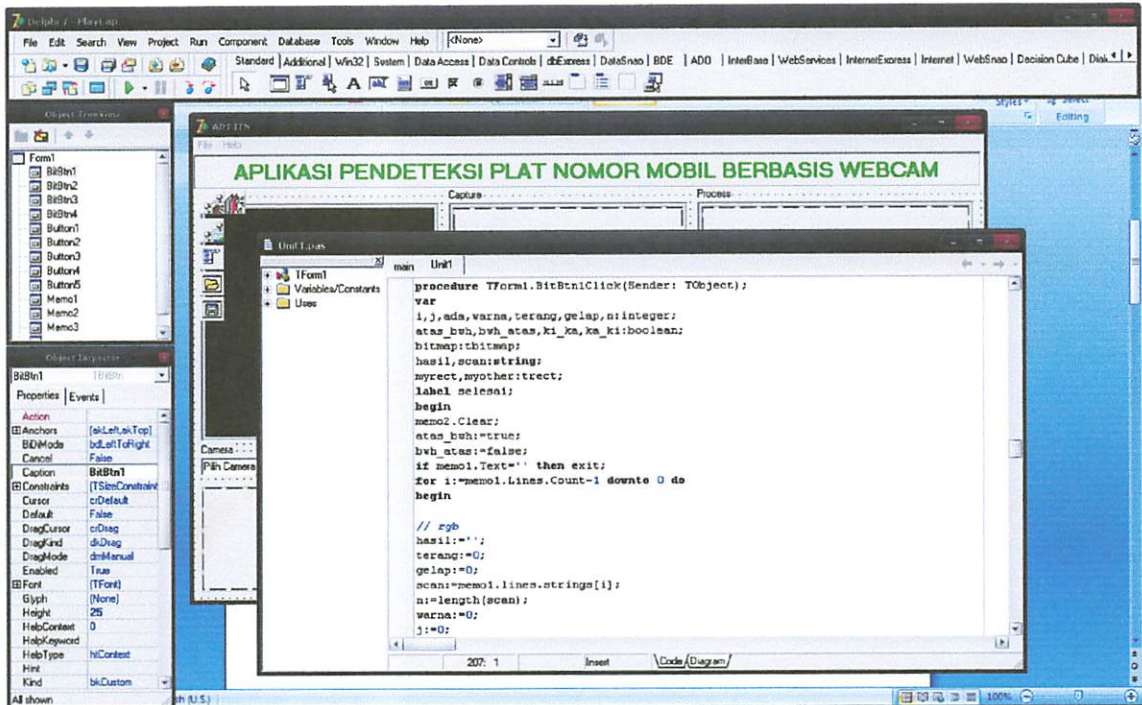
Dosen Pembimbing II

Ahmad Faisol, ST
NIP.P. 1031000431



Gambar B Tampilan program

Tampilan program aplikasi pendeteksi plat nomor berbasis webcam yang menggunakan komponen DSPack dalam interface webcam dan delphi 7.



Gambar B Contoh script

Script pada setiap tombol seperti Gambar B, yaitu mencari posisi plat nomor dengan menggunakan *script* . Dicari jika noise lebih dari 400 pixel maka dipastikan itu bukan plat nomor dan akan dihapus. Contoh seperti dibawah ini :

```
begin
memo2.Clear;
atas_bwh:=true;
bwh_atas:=false;
if memo1.Text="" then exit;
for i:=memo1.Lines.Count-1 downto 0 do
end
```

Pada *script* diatas gambar discan dari bawah ke atas sampai batas yang ditentukan pada memo1.

Contoh *script* dibawah ini untuk mengulang proses pixels terang dan gelap :

```
repeat
if scan[j]='0' then // terang
begin
terang:=terang+1;
repeat
j:=j+1;
if j=n then goto selesai;
until scan[j]='1';
end;
if scan[j]='1' then // gelap
begin
gelap:=gelap+1;
repeat
j:=j+1;
if j=n then goto selesai;
until scan [j]='1';
end;
```

Pada *script* diatas button dengan caption bitbtn1 memproses antara gelap dan terang dengan gelap dideklarasikan '1' dan terang '0'.sampai perulangan yng sudah ditentukan.

- **Script Program**

unit main;

interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DSUtil, StdCtrls, DSPack, DirectShow9, Menus, ExtCtrls, Buttons,
TeEngine, Series, TeeProcs, Chart, OpenCV;**

type

TVideoForm = class(TForm)

FilterGraph: TFilterGraph;

MainMenu1: TMainMenu;

Filter: TFilter;

SampleGrabber: TSampleGrabber;

Panel1: TPanel;

VideoWindow: TVideoWindow;

Panel2: TPanel;

Image: TImage;

mn_file: TMenuItem;

Devices: TMenuItem;

mn_Open: TMenuItem;

mn_Save: TMenuItem;

mn_Exit: TMenuItem;

mn_Edit: TMenuItem;

mn_Invers: TMenuItem;

mn_GrayScale: TMenuItem;

mn_Help: TMenuItem;

Opend: TOpenDialog;

Saved: TSaveDialog;

Panel3: TPanel;

img1: TImage;

mn_EdgeDetection: TMenuItem;

mn_Sobel: TMenuItem;

btn: TSpeedButton;

```

mn_Histogram: TMenuItem;
mn_Crop: TMenuItem;
mn_Capture: TMenuItem;
mn_Prewit: TMenuItem;
mn_Laplacian: TMenuItem;
FilterLPF1: TMenuItem;
Filterhpf1: TMenuItem;
Button1: TButton;
Button2: TButton;
Image9: TImage;
Button3: TButton;
// VideoWindow1: TVideoWindow;
procedure FormCreate(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure SampleGrabberBuffer(sender: TObject; SampleTime: Double;
  pBuffer: Pointer; BufferLen: Integer);
procedure mn_ExitClick(Sender: TObject);
procedure mn_OpenClick(Sender: TObject);
procedure mn_SaveClick(Sender: TObject);
procedure mn_InversClick(Sender: TObject);
function ConvertBitmapToInversBiner(const Bitmap: TBitmap): TBitmap;
procedure mn_GrayScaleClick(Sender: TObject);
function ConvertBitmapToGrayscale(const Bitmap: TBitmap): TBitmap;
procedure Histogram(const Bitmap: TBitmap);
procedure mn_CropClick(Sender: TObject);
function sobel(const Bitmap: TBitmap): TBitmap;
procedure mn_SobelClick(Sender: TObject);
procedure mn_CaptureClick(Sender: TObject);
function prewit(const Bitmap: TBitmap): TBitmap;
procedure mn_PrewitClick(Sender: TObject);
function Laplacian(const Bitmap: TBitmap): TBitmap;
procedure mn_LaplacianClick(Sender: TObject);
function Lpf(const Bitmap: TBitmap): TBitmap;
procedure FilterLPF1Click(Sender: TObject);

```

```

function hpff(const Bitmap: TBitmap): TBitmap;
procedure Filterhpff1Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure btnClick(Sender: TObject);
private
  { Déclarations privées }
public
  { Déclarations publiques }
  procedure OnSelectDevice(sender: TObject);
end;

var
  VideoForm: TVideoForm;
  SysDev: TSysDevEnum;
  atas,bawah:Integer;
  kanan,kiri:array[1..10] of Integer;
  datamod:array [1..255] of Real;
  h:array[0..255]of Integer;
implementation

uses Unit1;

{$R *.dfm}

procedure TVideoForm.FormCreate(Sender: TObject);
var
  i: integer;
  Device: TMenuItem;
begin
  WindowState:=wsMaximized;
  Panel1.Left:=0;

```

```

Panel1.Width:=Round(Screen.Width/3)-2;
Panel1.Height:=Round(Screen.Height/2);
Panel2.Left:=Round(Screen.Width/3);
Panel2.Width:=Round(Screen.Width/3)-2;
Panel2.Height:=Round(Screen.Height/2);
Panel3.Left:=2*Round(Screen.Width/3);
Panel3.Width:=Round(Screen.Width/3)-2;
Panel3.Height:=Round(Screen.Height/2);
SysDev:= TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
if SysDev.CountFilters > 0 then
  for i := 0 to SysDev.CountFilters - 1 do
    begin
      Device := TMenuItem.Create(Devices);
      Device.Caption := SysDev.Filters[i].FriendlyName;
      Device.Tag := i;
      Device.OnClick := OnSelectDevice;
      Devices.Add(Device);
    end;
  end;

procedure TVideoForm.OnSelectDevice(sender: TObject);
begin
  FilterGraph.ClearGraph;
  FilterGraph.Active := false;
  Filter.BaseFilter.Moniker := SysDev.GetMoniker(TMenuItem(Sender).tag);
  FilterGraph.Active := true;
  with FilterGraph as ICaptureGraphBuilder2 do
    RenderStream(@PIN_CATEGORY_PREVIEW, nil, Filter as IBaseFilter,
SampleGrabber as IBaseFilter, VideoWindow as IBaseFilter);
  FilterGraph.Play;
end;

procedure TVideoForm.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);

```

```
begin
// CallBack.Checked := False;
  SysDev.Free;
  FilterGraph.ClearGraph;
  FilterGraph.Active := false;
end;
```

```
procedure TVideoForm.SampleGrabberBuffer(sender: TObject;
  SampleTime: Double; pBuffer: Pointer; BufferLen: Integer);
begin
{ if CallBack.Checked then
  begin
    Image.Canvas.Lock;
    try
      SampleGrabber.GetBitmap(Image.Picture.Bitmap, pBuffer, BufferLen);
    finally
      Image.Canvas.Unlock;
    end;
  end; }
end;
```

```
procedure TVideoForm.mn_ExitClick(Sender: TObject);
begin
Application.Terminate;
end;
```

```
procedure TVideoForm.mn_OpenClick(Sender: TObject);
var
  gambar:TBitmap;
begin
if not Opend.Execute then Exit else
begin
  gambar := TBitmap.Create;
```

```

    gambar.LoadFromFile(Opend.FileName);
end;
if gambar.PixelFormat <> pf24bit then
gambar.PixelFormat:= pf24bit;
Image.Picture.Bitmap:=gambar;
gambar.Destroy;
end;

procedure TVideoForm.mn_SaveClick(Sender: TObject);
var
    gambar:TBitmap;
begin
if not Saved.Execute then Exit else
begin
gambar := Image.Picture.Bitmap;
gambar.SaveToFile(Saved.FileName);
end;
end;

function TVideoForm.ConvertBitmapToInversBiner(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 and RGB(255,0,0);
Green := (PixelColor and RGB(0,255,0))shr 8;
Blue := (PixelColor and RGB(0,0,255)) shr 16;

```

```

    Grayshade := Round(( Red + Green + Blue)/3);
    if Grayshade > 128 then Grayshade:=0 else Grayshade:=255;
    Canvas.Pixels[i, j] := RGB(Grayshade, Grayshade, Grayshade);
end;
Result := Bitmap;
end;

```

```

procedure TVideoForm.mn_InversClick(Sender: TObject);
var
    gambar:TBitmap;
begin
    gambar:=TBitmap.Create;
    gambar.Assign(image.Picture.Bitmap);
    img1.Picture.Bitmap:=ConvertBitmapToInversBiner(gambar);
    gambar.Destroy;
end;

```

```

function TVideoForm.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 and RGB(255,0,0);
                    Green := (PixelColor and RGB(0,255,0))shr 8;
                    Blue := (PixelColor and RGB(0,0,255)) shr 16;
                    Grayshade := Round(( Red + Green + Blue)/3);
                    Canvas.Pixels[i, j] := RGB(Grayshade, Grayshade, Grayshade);
                end;
            end;
        end;
    end;
end;

```

```

    end;
    Result := Bitmap;
end;

procedure TVideoForm.mn_GrayScaleClick(Sender: TObject);
var
    gambar:TBitmap;
begin
    gambar:=TBitmap.Create;
    gambar.Assign(image.Picture.Bitmap);
    img1.Picture.Bitmap:=ConvertBitmapToGrayscale(gambar);
    gambar.Destroy;
end;

procedure TVideoForm.mn_CaptureClick(Sender: TObject);
begin
    SampleGrabber.GetBitmap(Image.Picture.Bitmap);
end;

procedure TVideoForm.Histogram(const Bitmap: TBitmap);
var
    i, j: Integer;
    Grayshade, Red, Green, Blue: Byte;
    PixelColor: Longint;
begin
    for i:=1 to 256 do
    h[i]:=0;
    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 and RGB(255,0,0);
                    Green := (PixelColor and RGB(0,255,0))shr 8;

```

Blue := (PixelColor and RGB(0,0,255)) shr 16;

Grayshade := Round((Red + Green + Blue)/3);

h[Grayshade+1]:=h[Grayshade+1]+1;

```
end;  
    Result := Bitmap;  
end;
```

```
procedure TVideoForm.mn_GrayScaleClick(Sender: TObject);  
var  
    gambar:TBitmap;  
begin  
gambar:=TBitmap.Create;  
gambar.Assign(image.Picture.Bitmap);  
img1.Picture.Bitmap:=ConvertBitmapToGrayscale(gambar);  
gambar.Destroy;  
end;
```

```
procedure TVideoForm.mn_CaptureClick(Sender: TObject);  
begin  
SampleGrabber.GetBitmap(Image.Picture.Bitmap);  
end;
```

```
procedure TVideoForm.Histogram(const Bitmap: TBitmap);  
var  
    i, j: Integer;  
    Grayshade, Red, Green, Blue: Byte;  
    PixelColor: Longint;  
begin  
for i:=1 to 256 do  
h[i]:=0;  
    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 and RGB(255,0,0);  
                    Green := (PixelColor and RGB(0,255,0))shr 8;
```

```
end;
Result := Bitmap;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var
  Bitmap: TBitmap;
begin
  Bitmap := TBitmap.Create;
  Bitmap.Assign(Picture.Bitmap);
  Image1.Bitmap := Bitmap;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Bitmap := TBitmap.Create;
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
var
  i: Integer;
  Red, Green, Blue: Byte;
  PixelColor: Integer;
begin
  for i := 0 to 255 do
    with Bitmap do
      for i := 0 to Width - 1 do
        for j := 0 to Height - 1 do
          PixelColor := ColorOfPixel(i, j);
          Red := PixelColor and $FF;
          Green := (PixelColor and $FF0) shr 8;
```

```

    Blue := (PixelColor and RGB(0,0,255)) shr 16;
    Grayshade := Round(( Red + Green + Blue)/3);
    h[Grayshade+1]:=h[Grayshade+1]+1;
end;
end;

procedure TVideoForm.mn_CropClick(Sender: TObject);
var
    Bitmap: TBitmap;
    MyRect, MyOther: TRect;
begin
    MyRect := Rect(kiri[1],atas,kanan[5],bawah);//posisi crop(x1,y1,x2,y2)
    MyOther := Rect(0,0,60,80);//skala gambar wis fix
    Bitmap := TBitmap.Create;
    Bitmap.Assign(img1.Picture.Bitmap);
// Image1.Canvas.CopyRect(MyOther,Bitmap.Canvas,MyRect);
    Bitmap.Destroy;
end;
//function TVideoForm.cropgambar(var dest: TImage; a,b,c,d:Integer)
function Tvideoform.sobel(const Bitmap: TBitmap): TBitmap;
const
    con_sobel : array [1..3,1..3] of Integer = ((-1,0,1),(-2,0,2),(-1,0,1));
    con_invsobel: array [1..3,1..3] of Integer = ((-1,-2,-1),(0,0,0),(1,2,1));
var
    i,j,k,l:Integer;
    PixelColor,z1,z2,z: Longint;
begin
    with bitmap do
    for i:=1 to Width do
    for j:=1 to Height do
        begin
            z1:=0;
            z2:=0;
            for k:=-1 to 1 do

```

```

for l:=-1 to 1 do
  begin
    PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+l])and RGB(255,0,0);
    z1:= z1 + con_sobel[k+2,l+2] * PixelColor;
    z2:= z2 + con_invsobel[k+2,l+2] * PixelColor;
  end;
z:=Abs(z1+z2);
if z > 255 then z:=255;
Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
end;
Result := Bitmap;
end;
procedure TVideoForm.mn_SobelClick(Sender: TObject);
var
  gambar,gambar1,gambar2:TBitmap;
begin
gambar:=TBitmap.Create;
gambar.Assign(image.Picture.Bitmap);
gambar1:=ConvertBitmapToGrayscale(gambar);
gambar2:=sobel(gambar1);
img1.Picture.Bitmap:=gambar2;
gambar.Destroy;
end;

function Tvideoform.prewit(const Bitmap: TBitmap): TBitmap;
const
  con_sobel : array [1..3,1..3] of Integer = ((-1,0,1),(-1,0,1),(-1,0,1));
  con_invsobel: array [1..3,1..3] of Integer = ((-1,-1,-1),(0,0,0),(1,1,1));
var
  i,j,k,l:Integer;
  PixelColor,z1,z2,z: Longint;
begin
with bitmap do
for i:=1 to Width do

```

```

for j:=1 to Height do
  begin
    z1:=0;
    z2:=0;
    for k:=-1 to 1 do
      for l:=-1 to 1 do
        begin
          PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+l])and RGB(255,0,0);
          z1:= z1 + con_sobel[k+2,l+2] * PixelColor;
          z2:= z2 + con_invsobel[k+2,l+2] * PixelColor;
        end;
        z:=Abs(z1+z2);
        if z > 255 then z:=255;
        Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
      end;
    Result := Bitmap;
  end;
procedure TVideoForm.mn_PrewitClick(Sender: TObject);
var
  gambar,gambar1,gambar2:TBitmap;
begin

  gambar:=TBitmap.Create;
  gambar.Assign(image.Picture.Bitmap);
  gambar1:=ConvertBitmapToGrayscale(gambar);
  gambar2:=prewit(gambar1);
  img1.Picture.Bitmap:=gambar2;
  gambar.Destroy;
end;
function Tvideoform.Laplacian(const Bitmap: TBitmap): TBitmap;
const
  con_sobel : array [1..3,1..3] of Integer = ((0,1,0),(1,-4,1),(0,1,0));
var
  i,j,k,l:Integer;

```

```

    PixelColor,z1,z2,z: Longint;
begin
with bitmap do
for i:=1 to Width do
for j:=1 to Height do
    begin
        z1:=0;
        for k:=-1 to 1 do
        for l:=-1 to 1 do
            begin
                PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+l])and RGB(255,0,0);
                z1:= z1 + con_sobel[k+2,l+2] * PixelColor;
            end;
            z:=Abs(z1);
            if z > 255 then z:=255;
            Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
        end;
        Result := Bitmap;
    end;
procedure TVideoForm.mn_LaplacianClick(Sender: TObject);
var
    gambar,gambar1,gambar2:TBitmap;
begin
gambar:=TBitmap.Create;
gambar.Assign(image.Picture.Bitmap);
gambar1:=ConvertBitmapToGrayscale(gambar);
gambar2:=laplacian(gambar1);
img1.Picture.Bitmap:=gambar2;
gambar.Destroy;
end;
function Tvideoform.Lpf(const Bitmap: TBitmap): TBitmap;
const
    con_sobel : array [1..3,1..3] of Integer = ((1,1,1),(1,1,1),(1,1,1));
var

```

```

i,j,k,l:Integer;z1:real;
PixelColor,z2,z: Longint;
begin
with bitmap do
for i:=1 to Width do
for j:=1 to Height do
begin
z1:=0;
for k:=-1 to 1 do
for l:=-1 to 1 do
begin
PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+l])and RGB(255,0,0);
z1:= z1 + con_sobel[k+2,l+2] * PixelColor/9;
end;
z:=round(Abs(z1));
if z > 255 then z:=255;
Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
end;
Result := Bitmap;
end;
procedure TVideoForm.FilterLPF1Click(Sender: TObject);
var
gambar,gambar1,gambar2:TBitmap;
begin
gambar:=TBitmap.Create;
gambar.Assign(img1.Picture.Bitmap);
gambar1:=ConvertBitmapToGrayscale(gambar);
gambar2:=lpf(gambar1);
img1.Picture.Bitmap:=gambar2;
gambar.Destroy;
end;
function Tvideoform.hpf(const Bitmap: TBitmap): TBitmap;
const
con_sobel : array [1..3,1..3] of Integer = ((0,-1,0),(-1,4,-1),(0,-1,0));

```

```

var
  i,j,k,l:Integer;z1:real;
  PixelColor,z2,z: Longint;
begin
with bitmap do
for i:=1 to Width do
for j:=1 to Height do
  begin
    z1:=0;
    for k:=-1 to 1 do
    for l:=-1 to 1 do
      begin
        PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+l])and RGB(255,0,0);
        z1:= z1 + con_sobel[k+2,l+2] * PixelColor;
      end;
    z:=round(Abs(z1));
    if z > 255 then z:=255;
    Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
  end;
  Result := Bitmap;
end;
procedure TVideoForm.Filterhp1Click(Sender: TObject);
var
  gambar,gambar1,gambar2:TBitmap;
begin
gambar:=TBitmap.Create;
gambar.Assign(img1.Picture.Bitmap);
gambar1:=ConvertBitmapToGrayscale(gambar);
gambar2:=hpf(gambar1);
img1.Picture.Bitmap:=gambar2;
gambar.Destroy;
end;

procedure TVideoForm.Button1Click(Sender: TObject);

```

```

var
  i, j, ada, warna, n: Integer;
  pix: char;
  datapix: string;
  atas_bwh, bwh_atas, ki_ka, ka_ki: Boolean;
  Bitmap: TBitmap;
  MyRect, MyOther: TRect;
begin
  form1.Memo1.Clear;
  form1.Memonomor.Clear;

  with img1 do
    begin
      // ListBox1.Items.Add(IntToStr(img1.Width)+' x '+inttostr(img1.Height));
      //atas_bwh:=True;bwh_atas:=False;
      for i := 0 to Height - 1 do
        begin
          // rgb(
            warna:=0;
            datapix:="";
            for j := 0 to Width - 1 do
              begin
                if Canvas.Pixels[j,i] = RGB(255,255,255) then
                  begin
                    pix:='0';
                  end
                else if Canvas.Pixels[j,i] = RGB(0,0,0) then
                  begin
                    pix:='1';
                  end;
                datapix:=datapix+pix;
              end;
            datapix:=datapix+'|';
          form1.memo1.Lines.Add(datapix);
        end;
      end;
    end;
  end;

```



```

Bitmap.Assign(image9.Picture.Bitmap);
Bitmap.Destroy;

end;

procedure TVideoForm.Button3Click(Sender: TObject);
var
  i, j, ada, warna, n, crop_atas, crop_bawah, batas: Integer;
  pix: char;
  datapix, scan: string;
  atas_bwh, bwh_atas, ki_ka, ka_ki: Boolean;
  Bitmap: TBitmap;
  MyRect, MyOther: TRect;
begin
  with videoform.Image do
  begin
    scan:=form1.memonomor.Lines.Strings[0];
    batas:=strtoint(scan);
    crop_atas:=batas-4 ;//load 3 pixel sebelumnya
    i:=form1.memonomor.Lines.Count-1;
    scan:=form1.memonomor.Lines.Strings[i]; //batas bawah plat hasil crop
    batas:=strtoint(scan);
    crop_bawah:=batas+6 ;//load 6 pixel sesudahnya

    for i := 0 to Width - 1 do
      for j := crop_atas to crop_bawah do
        begin
          //PixelColor := ColorToRGB(Canvas.Pixels[i, j]);
          image9.Canvas.Pixels[i,j]:=Canvas.Pixels[i, j];
        end;
      end;
    end;
  end;
end;

```

```
    procedure TVideoForm.btnClick(Sender: TObject);  
    begin  
        end;  
    end.
```

```
unit main;
```

```
interface
```

```
uses
```

```
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
    Dialogs, DSUtil, StdCtrls, DSPack, DirectShow9, Menus, ExtCtrls, Buttons,  
    TeEngine, Series, TeeProcs, Chart,OpenCV;
```

```
type
```

```
TVideoForm = class(TForm)  
    FilterGraph: TFilterGraph;  
    MainMenu1: TMainMenu;  
    Filter: TFilter;  
    SampleGrabber: TSampleGrabber;  
    Panel1: TPanel;  
    VideoWindow: TVideoWindow;  
    Panel2: TPanel;  
    Image: TImage;  
    mn_file: TMenuItem;  
    Devices: TMenuItem;  
    mn_Open: TMenuItem;  
    mn_Save: TMenuItem;  
    mn_Exit: TMenuItem;  
    mn_Edit: TMenuItem;  
    mn_Invers: TMenuItem;  
    mn_GrayScale: TMenuItem;  
    mn_Help: TMenuItem;  
    Opend: TOpenDialog;  
    Saved: TSaveDialog;  
    Panel3: TPanel;  
    img1: TImage;
```

```
mn_EdgeDetection: TMenuItem;
mn_Sobel: TMenuItem;
btn: TSpeedButton;
mn_Histogram: TMenuItem;
mn_Crop: TMenuItem;
mn_Capture: TMenuItem;
mn_Prewit: TMenuItem;
mn_Laplacian: TMenuItem;
FilterLPF1: TMenuItem;
Filterhpf1: TMenuItem;
Button1: TButton;
Button2: TButton;
Image9: TImage;
Button3: TButton;
// VideoWindow1: TVideoWindow;
procedure FormCreate(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure SampleGrabberBuffer(sender: TObject; SampleTime: Double;
  pBuffer: Pointer; BufferLen: Integer);
procedure mn_ExitClick(Sender: TObject);
procedure mn_OpenClick(Sender: TObject);
procedure mn_SaveClick(Sender: TObject);
procedure mn_InversClick(Sender: TObject);
function ConvertBitmapToInversBiner(const Bitmap: TBitmap): TBitmap;
procedure mn_GrayScaleClick(Sender: TObject);
function ConvertBitmapToGrayscale(const Bitmap: TBitmap): TBitmap;
procedure Histogram(const Bitmap: TBitmap);
procedure mn_CropClick(Sender: TObject);
function sobel(const Bitmap: TBitmap): TBitmap;
procedure mn_SobelClick(Sender: TObject);
procedure mn_CaptureClick(Sender: TObject);
function prewit(const Bitmap: TBitmap): TBitmap;
procedure mn_PrewitClick(Sender: TObject);
function Laplacian(const Bitmap: TBitmap): TBitmap;
```

```

procedure mn_LaplacianClick(Sender: TObject);
function Lpf(const Bitmap: TBitmap): TBitmap;
procedure FilterLPF1Click(Sender: TObject);
function hpf(const Bitmap: TBitmap): TBitmap;
procedure Filterhpf1Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure btnClick(Sender: TObject);
private
  { Déclarations privées }
public
  { Déclarations publiques }
  procedure OnSelectDevice(sender: TObject);
end;

var
  VideoForm: TVideoForm;
  SysDev: TSysDevEnum;
  atas,bawah:Integer;
  kanan,kiri:array[1..10] of Integer;
  datamod:array [1..255] of Real;
  h:array[0..255]of Integer;
implementation

uses Unit1;

{$R *.dfm}

procedure TVideoForm.FormCreate(Sender: TObject);
var
  i: integer;
  Device: TMenuItem;

```

```

begin
WindowState:=wsMaximized;
Panel1.Left:=0;
Panel1.Width:=Round(Screen.Width/3)-2;
Panel1.Height:=Round(Screen.Height/2);
Panel2.Left:=Round(Screen.Width/3);
Panel2.Width:=Round(Screen.Width/3)-2;
Panel2.Height:=Round(Screen.Height/2);
Panel3.Left:=2*Round(Screen.Width/3);
Panel3.Width:=Round(Screen.Width/3)-2;
Panel3.Height:=Round(Screen.Height/2);
SysDev:= TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
if SysDev.CountFilters > 0 then
  for i := 0 to SysDev.CountFilters - 1 do
    begin
      Device := TMenuItem.Create(Devices);
      Device.Caption := SysDev.Filters[i].FriendlyName;
      Device.Tag := i;
      Device.OnClick := OnSelectDevice;
      Devices.Add(Device);
    end;
  end;
end;

procedure TVideoForm.OnSelectDevice(sender: TObject);
begin
  FilterGraph.ClearGraph;
  FilterGraph.Active := false;
  Filter.BaseFilter.Moniker := SysDev.GetMoniker(TMenuItem(Sender).tag);
  FilterGraph.Active := true;
  with FilterGraph as ICaptureGraphBuilder2 do
    RenderStream(@PIN_CATEGORY_PREVIEW, nil, Filter as IBaseFilter,
SampleGrabber as IBaseFilter, VideoWindow as IbaseFilter);
  FilterGraph.Play;
end;

```

```
procedure TVideoForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
// Callback.Checked := False;
  SysDev.Free;
  FilterGraph.ClearGraph;
  FilterGraph.Active := false;
end;
```

```
procedure TVideoForm.SampleGrabberBuffer(sender: TObject;
  SampleTime: Double; pBuffer: Pointer; BufferLen: Integer);
begin
{ if Callback.Checked then
  begin
    Image.Canvas.Lock;
    try
      SampleGrabber.GetBitmap(Image.Picture.Bitmap, pBuffer, BufferLen);
    finally
      Image.Canvas.Unlock;
    end;
  end; }
end;
```

```
procedure TVideoForm.mn_ExitClick(Sender: TObject);
begin
Application.Terminate;
end;
```

```
procedure TVideoForm.mn_OpenClick(Sender: TObject);
var
  gambar:TBitmap;
begin
if not Opend.Execute then Exit else
```

```

begin
  gambar := TBitmap.Create;
  gambar.LoadFromFile(Openend.FileName);
end;
if gambar.PixelFormat <> pf24bit then
gambar.PixelFormat:= pf24bit;
Image.Picture.Bitmap:=gambar;
gambar.Destroy;
end;

procedure TVideoForm.mn_SaveClick(Sender: TObject);
var
  gambar:TBitmap;
begin
if not Saved.Execute then Exit else
begin
gambar := Image.Picture.Bitmap;
gambar.SaveToFile(Saved.FileName);
end;
end;

function TVideoForm.ConvertBitmapToInversBiner(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 and RGB(255,0,0);

```

```

    Green := (PixelColor and RGB(0,255,0))shr 8;
    Blue := (PixelColor and RGB(0,0,255)) shr 16;
    Grayshade := Round(( Red + Green + Blue)/3);
    if Grayshade > 128 then Grayshade:=0 else Grayshade:=255;
    Canvas.Pixels[i, j] := RGB(Grayshade, Grayshade, Grayshade);
end;
Result := Bitmap;
end;

```

```

procedure TVideoForm.mn_InversClick(Sender: TObject);
var
    gambar:TBitmap;
begin
    gambar:=TBitmap.Create;
    gambar.Assign(image.Picture.Bitmap);
    img1.Picture.Bitmap:=ConvertBitmapToInversBiner(gambar);
    gambar.Destroy;
end;

```

```

function TVideoForm.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 and RGB(255,0,0);
                    Green := (PixelColor and RGB(0,255,0))shr 8;
                    Blue := (PixelColor and RGB(0,0,255)) shr 16;
                    Grayshade := Round(( Red + Green + Blue)/3);

```



```

    Green := (PixelColor and RGB(0,255,0))shr 8;
    Blue := (PixelColor and RGB(0,0,255)) shr 16;
    Grayshade := Round(( Red + Green + Blue)/3);
    h[Grayshade+1]:=h[Grayshade+1]+1;
end;
end;

procedure TVideoForm.mn_CropClick(Sender: TObject);
var
    Bitmap: TBitmap;
    MyRect, MyOther: TRect;
begin
    MyRect := Rect(kiri[1],atas,kanan[5],bawah);//posisi crop(x1,y1,x2,y2)
    MyOther := Rect(0,0,60,80);//skala gambar wis fix
    Bitmap := TBitmap.Create;
    Bitmap.Assign(img1.Picture.Bitmap);
// Image1.Canvas.CopyRect(MyOther,Bitmap.Canvas,MyRect);
    Bitmap.Destroy;
end;
//function TVideoForm.cropgambar(var dest: TImage; a,b,c,d:Integer)
function Tvideoform.sobel(const Bitmap: TBitmap): TBitmap;
const
    con_sobel : array [1..3,1..3] of Integer = ((-1,0,1),(-2,0,2),(-1,0,1));
    con_invsobel: array [1..3,1..3] of Integer = ((-1,-2,-1),(0,0,0),(1,2,1));
var
    i,j,k,l:Integer;
    PixelColor,z1,z2,z: Longint;
begin
    with bitmap do
    for i:=1 to Width do
    for j:=1 to Height do
        begin
            z1:=0;
            z2:=0;

```

```

for k:=-1 to 1 do
for l:=-1 to 1 do
begin
PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+l])and RGB(255,0,0);
z1:= z1 + con_sobel[k+2,l+2] * PixelColor;
z2:= z2 + con_invsobel[k+2,l+2] * PixelColor;
end;
z:=Abs(z1+z2);
if z > 255 then z:=255;
Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
end;
Result := Bitmap;
end;
procedure TVideoForm.mn_SobelClick(Sender: TObject);
var
gambar,gambar1,gambar2:TBitmap;
begin
gambar:=TBitmap.Create;
gambar.Assign(image.Picture.Bitmap);
gambar1:=ConvertBitmapToGrayscale(gambar);
gambar2:=sobel(gambar1);
img1.Picture.Bitmap:=gambar2;
gambar.Destroy;
end;

function Tvideoform.prewit(const Bitmap: TBitmap): TBitmap;
const
con_sobel : array [1..3,1..3] of Integer = ((-1,0,1),(-1,0,1),(-1,0,1));
con_invsobel: array [1..3,1..3] of Integer = ((-1,-1,-1),(0,0,0),(1,1,1));
var
i,j,k,l:Integer;
PixelColor,z1,z2,z: Longint;
begin
with bitmap do

```

```

for i:=1 to Width do
for j:=1 to Height do
  begin
    z1:=0;
    z2:=0;
    for k:=-1 to 1 do
    for l:=-1 to 1 do
      begin
        PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+l])and RGB(255,0,0);
        z1:= z1 + con_sobel[k+2,l+2] * PixelColor;
        z2:= z2 + con_invsobel[k+2,l+2] * PixelColor;
      end;
    z:=Abs(z1+z2);
    if z > 255 then z:=255;
    Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
  end;
  Result := Bitmap;
end;
procedure TVideoForm.mn_PrewitClick(Sender: TObject);
var
  gambar,gambar1,gambar2:TBitmap;
begin

gambar:=TBitmap.Create;
gambar.Assign(image.Picture.Bitmap);
gambar1:=ConvertBitmapToGrayscale(gambar);
gambar2:=prewit(gambar1);
img1.Picture.Bitmap:=gambar2;
gambar.Destroy;
end;
function Tvideoform.Laplacian(const Bitmap: TBitmap): TBitmap;
const
  con_sobel : array [1..3,1..3] of Integer = ((0,1,0),(1,-4,1),(0,1,0));
var

```

```

i,j,k,l:Integer;
PixelColor,z1,z2,z: Longint;
begin
with bitmap do
for i:=1 to Width do
for j:=1 to Height do
begin
z1:=0;
for k:=-1 to 1 do
for l:=-1 to 1 do
begin
PixelColor:=ColorToRGB(Canvas.Pixels[i+k,j+1])and RGB(255,0,0);
z1:= z1 + con_sobel[k+2,l+2] * PixelColor;
end;
z:=Abs(z1);
if z > 255 then z:=255;
Canvas.Pixels[i-1,j-1]:=RGB(z,z,z)
end;
Result := Bitmap;
end;
end;

```