

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS ♦

JUDUL: RECOGNITION OF EMERGENCY AND NON-EMERGENCY LIGHT USING MATROX AND VB6

SESI PENGAJIAN: 2006/2007

Saya MOHD NAZERI BIN MUHAMMAD (850926036017)
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

**0330 JALAN J4 FASA6 TAMAN
MELAWATI 53100 KUALA
LUMPUR**

MOHD RAZALI BIN DAUD
(Nama Penyelia)

Tarikh: **26 NOVEMBER 2007**

Tarikh: : **26 NOVEMBER 2007**

- CATATAN:
- * Potong yang tidak berkenaan.
 - ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
 - ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

RECOGNITION OF EMERGENCY AND NON-EMERGENCY LIGHT USING
MATROX AND VB6

MOHD NAZERI BIN MUHAMMAD

This thesis is submitted as partial fulfillment of the requirements for the award of the
Bachelor of Electrical Engineering (Power Systems)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2007

“I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Power Systems)”

Signature : _____

Name : MOHD RAZALI BIN DAUD

Date : 26 NOVEMBER 2007

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : _____

Author : MOHD NAZERI BIN MUHAMMAD

Date : 26 NOVEMBER 2007

ACKNOWLEDGEMENT

Alhamdulillah, thank to my Lord, Allah S.W.T for giving me strength and showing me the way to complete this project.

Special thank to Mr. Mohd Razali Bin Daud that has be patient and giving a lot support to finish this project. Which I really appreciate your understanding to my project. Thanks you also to my course member that had be given cooperation in developing this project.

I would like to thank my family that has given moral support and motivation to finish this project.

I would like to thank University Malaysia Pahang (UMP) for giving opportunities to learn something new. Lastly I want to wish special thank Balai Polis Kuantan that has helping to what I have achieve now.

ABSTRACT

In a daily life there is always a problem to an emergency vehicle such as ambulance, patrol car and fire-fighter to pass through traffic light because it was red and it's disturbing the drive. A camera system that consists of camera as sensor and some other parts that detect an emergency vehicle could solve the problem from blocked by another vehicle. When the camera that is placed at a certain distance from the traffic light detects an emergency vehicle it would sent a data and it will start the system, then it would activate the traffic light to change to green until the emergency vehicle pass through. This project would focus on developing the software for image processing of emergency light. The captured image will be analyzed using blob analysis to identify connection regions of pixel within the image. The result will be displayed on computer screen to indicate the processed image is emergency light or non-emergency light. This project would use a Visual Basic 6 and Matrox where the image will be process and analyze between emergency and non-emergency light.

ABSTRAK

Dalam kehidupan harian, lampu isyarat merah menjadi masalah kepada kenderaan kecemasan seperti ambulan, kereta peronda dan bomba kerana mengganggu perjalanan. Sebuah sistem kamera mengesan kenderaan kecemasan yang melibatkan kamera sebagai alat pengesanan dan beberapa bahagian lain bagi menyelesaikan masalah perjalanan dihalang oleh kenderaan lain. Apabila kamera yang diletak dalam jarak yang tertentu daripada lampu isyarat mengesan kenderaan kecemasan, maklumat akan dihantar dan menjana sistem lalu mengaktifkan lampu isyarat untuk bertukar kepada hijau sehingga kenderaan tersebut melepasinya. Projek ini akan mengfokuskan pada pembinaan perisian untuk pemprosesan gambar lampu kecemasan. Gambar yang diambil akan diproses menggunakan "Blob Analysis" untuk menentukan kawasan sambungan pixel didalam gambar. Keputusan akan dipaparkan pada skrin computer, samada gambar yang diproses lampu kecemasan atau sebaliknya. Projek ini akan menggunakan Visual Basic 6 dan Matrox dimana gambar akan diproses dan selidik antara lampu kecemasan atau bukan lampu kecemasan.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	BORANG PENGESAHAN STATUS TESIS	i
	SUPERVISOR ACKNOWLEDGEMENT	ii
	DECLARATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	LIST OF FIGURE	vii
	LIST OF TABLE	x
1	INTRODUCTION	
	1.1 Introduction	1
	1.2 Problem statement	2
	1.3 Research Objective	3
	1.4 Significance Of study (Main Objectives)	3
	1.5 Scope of work	4
2	LITERITURE REVIEW	
	2.1 Introduction	5
	2.2 Visual Basic	5
	2.3 Matrox imaging library (MIL)	6
	2.4 Image Processing	7

	2.5 Noise	7
	2.5.1 Types of image noise	8
	2.6 Type of emergency light	9
	2.6.1 Optical type	9
	2.6.2 Mounting type	9
	2.7 Threshold	10
3	METHODOLOGY	
	3.1 Interfacing MIL with Visual Basic	13
	3.2 Software structure	14
	3.3 GUI development	16
	3.3.1 Picture Box	20
	3.3.2 Display and System	21
	3.3.3 Image control	22
	3.3.4 Command Button	24
	3.3.5 Text Box	26
	3.3.6 List Box	26
	3.3.7 Image Processing Control	31
	3.4 Blob analysis identification	32
	3.5 Blob Analysis Property setting	36
	3.6 Binarize Method	39
	3.7 Filtering Method	39
	3.8 Analyzing Image	40
	3.9 Operation	42
4	RESULT AND DISCUSSION	
	4.1 Introduction	46
	4.2 Probability variable of emergency light	47
	4.2.1 Old model of emergency light	48

4.2.2	New model of emergency light	51
4.2.3	Emergency light exceed range	54
4.3	Binarized result	57
4.3.1	Old model	57
4.3.2	New model	59
4.4	Filtering Result	60
4.4.1	Old model	61
4.4.2	New model	63
4.5	Performance of image processing	65
4.6	Summary of the result	65
5	CONCLUSION AND RECOMENDATION	
5.1	Summary of the work	66
5.2	Recommendations	67
	REFERENCES	68
	APPENDICE	69

LIST OF FIGURES

FIGURE	TITLE	PAGE
1.1	Block Diagram of Camera System	2
2.1	Type of noise	8
2.2	The triangle algorithm is based on finding the value of b that gives the maximum distance d .	12
3.1	Software structure	14
3.2	New Project dialog	16
3.3	IDE with a Standard EXE project open	17
3.4	Properties window with (a) alphabetic and (b) categorized tabs	18
3.5	Form layout window	19
3.6	Component dialog	20
3.7	Load Picture Box into Form1	20
3.8	Load Display into Form1	21
3.9	Image control property	23
3.10	Property of image (a) Parent Image and (b) Child Image	24
3.11	A Command Button	25
3.12	An example of load command	25
3.13	A text box	26
3.14	A List Box icon	27
3.15	Image processing property (a) General and (b) Image	31
3.16	The size of blob with foreground value = zero	33
3.17	The size of blob with foreground value = Non-zero	33
3.18	The size of blob with foreground value = zero	34
3.19	Example of Labeled blob	34

3.20	Blob analysis control property	35
3.21	Original Image	40
3.22	Analyze image	41
3.23	Binarize image	41
3.24	Filtering image	41
3.25	First calculations of blob	41
3.26	Second calculation of blob	42
3.27	Non emergency light filtered image	42
3.28	Flow chart of the operation	44
3.29	Software window	45
4.1	Original image of non-emergency light	48
4.2	Result of figure 4.1- non-emergency light	48
4.3	Original image of non-emergency light- blob left	49
4.4	Result of figure 4.3- emergency light	49
4.5	Original image of emergency light- blob right	50
4.6	Result of figure 4.5- emergency light	50
4.7	Original image non-emergency light	51
4.8	Result of figure 4.7- non-emergency light	52
4.9	Original image of emergency light- blob left	52
4.10	Result of figure 4.9- emergency light	53
4.11	Original image of emergency light- blob right	53
4.12	Result of figure 4.11- emergency light	54
4.13	Original image exceed range	55
4.14	Result of figure 4.13- non-emergency	55
4.15	Original image of emergency light	56
4.16	Result of figure 4.15- non emergency light	56
4.17	Non emergency light	57
4.18	emergency light on the left side	58
4.19	Emergency light on the right side	58
4.20	Non-emergency light	59
4.21	Emergency light on the left side	59
4.22	Emergency light on the right side	60

4.23	Non emergency light	61
4.24	Emergency light on the left side	61
4.25	Emergency light on the right side	62
4.26	Non emergency light	63
4.27	Emergency light on the left side	63
4.28	Emergency light on the right side	64

LIST OF TABLE

TABLE	TITLE	PAGE
3.1	Control Description	15
3.2	Image property	23
3.3	Listbox	27
3.4	Probability Value of blob	28
3.5	Property listing for the program	29
3.6	Image processing property setting	31
3.7	Description of Identifier Blob Type	35
3.8	Attributes property setting	36
3.9	Feature list property setting	37
3.10	Feature property setting	38
4.1	Data of analyze image	65
4.2	Result of analyze image	65

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the daily life we always see that there is always a problem to an emergency vehicle such as ambulance, patrol car and fire-fighter to pass through traffic light because it was red and it's disturbing the drive. This situation is often happen because lack of cooperation from civilian, sometimes driver do not had an experience of this situation so they will waited the traffic light to turn green. There is also a conflict with driver from another traffic light where when emergency light past through traffic light on red condition the car from other traffic light did not notice the emergency vehicle and it will cause accidence. The situation can cause a death if the situation appeared. The top priority of this project is to make sure that emergency vehicle past through a traffic light smoothly and safely.

To achieve this target, camera system that consists of camera, frame grabber, computer and hardware to activate the traffic light could solve the problem from blocked by another vehicle. A camera at a certain distance is use as a sensor to take an image in a real time where it needs to use a frame grabber to process the real time image. From the frame grabber image will be loaded to computer where all processing of image will be done, result of image processing will be display by using Visual Basic 6 where GUI is implement in this project. After the result is display a software will activated the hardware to change the traffic light to green until the

emergency vehicle past through. Then the traffic light will operate in a normalized operation.

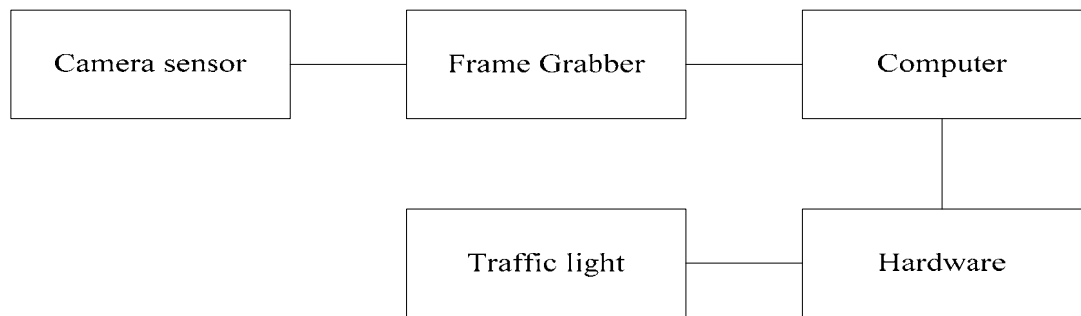


Figure 1.1: Block Diagram of Camera System

1.2 Problem statement

In this project there are some main problems that must inquire, which is will be used to developing the project. The questioning problem is show below:-

- i. How the image will be taken?
- ii. How to process the image?
- iii. What is required software to process the image taken?
- iv. What the best method to analyze image?

1.3 Research Objective

Research of this project must take account concerning list below: -

- i. To recognize suitable technique to captured image.
- ii. To recognize method of process the image
- iii. To recognize suitable software to process the captured image.
- iv. To identify technique to analyze image.

1.4 Significance Of study (Main Objectives)

The main objective of this project is:

- i. To study on image processing using Visual Basic 6 (VB6) and Matrox. It is because to analyze and process the image will perform by using Visual Basic 6 and Matrox (MIL). So it is needed to familiarize with the software, where it will help in developing it.
- ii. To develop software that can determine between emergency light and non-emergency light, where it can be used as an application to solve problem regarding traffic blocking the emergency vehicle when there is an emergency. The emergency light of emergency vehicle will be analyzed by the refraction color of the emergency light.

1.5 Scope of work

This project would focus on the image processing of police petrol car emergency light for the software development. The result would be displayed on computer screen to indicate the processed image is emergency light or non-emergency light. This project would use a Visual Basic 6 and Matrox Imaging Library 7 (MIL) to process and analyze the capture image between emergency and non-emergency light.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter will summarize the content of the paper, report and article that were studied related to the project. The basic concept is illustrated in this chapter.

2.2 Visual Basic

Visual Basic is an application that uses to create a graphical user interface (GUI), where ActiveX control is one of the applications in Visual Basic. The form is created by using drag and drop techniques, the control can be place in the form by drag the control into the toolbox. By inserting code in command button control the program can automatically translate the case of the text being entered, or even prevent certain characters from being inserted [1].

VB presents a 3-step approach for creating programs:

- i. Design the appearance of your application.
- ii. Assign property settings to the objects of your program.
- iii. Write the code to direct specific tasks at runtime.

2.3 Matrox imaging library (MIL)

The Matrox imaging library (MIL) is a hardware-independent library divided into different modules based on functionality. Where is has lot of function of image processing such as blob analysis, OCR, pattern matching and lot more. Matrox imaging library is an extended toolbox for Visual basic where the command language for Matrox is Visual Basic and Visual C++ [2].

2.4 Image Processing

Image processing is form of information processing from the image, there a various image processing operation [3]:

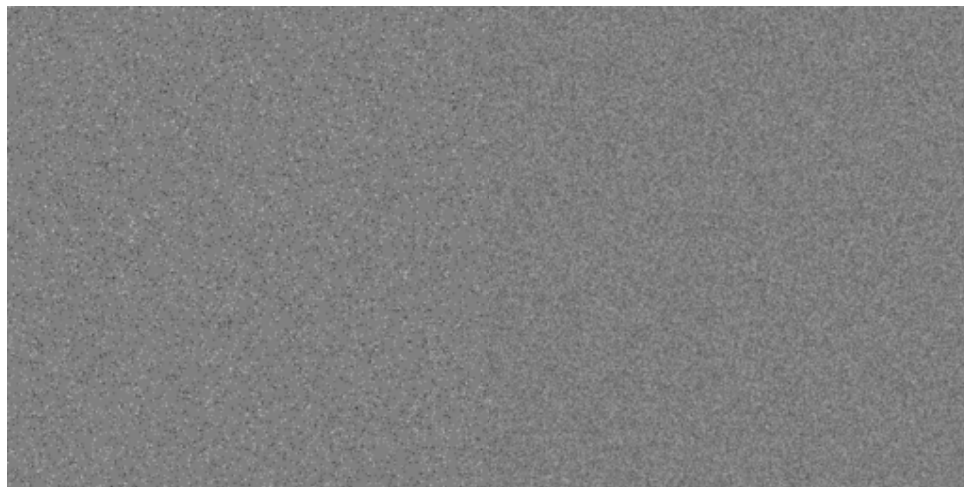
- i. Combination of two or more images
- ii. Image editing and digital retouching
- iii. Image restoration to increase the quality of a digital image
- iv. Segmentation of the image into regions

2.5 Noise

Image noise is an unwanted condition where it is because fluctuation of pixel in an image, where mostly appeared in image regions with low signal level, such as shadow regions or underexposed images. Removing noise cannot be done without loss of some information in the image [4].

2.5.1 Types of image noise

- i. In salt and pepper noise (also known as random noise or independent noise), pixels in the image are vastly different in color from their surrounding pixels. Generally this type of noise will only affect a small number of image pixels. When viewed, the image contains dark and white dots, hence the term salt and pepper noise. Typical sources include flecks of dust on the lens or inside the camera.
- ii. In Gaussian noise (dependent noise), an amount of noise is added to every part of the picture. Each pixel in the image will be changed from its original value by a (usually) small amount [5].



(a) Salt and paper noise

(b) Gaussian noise

Figure 2.1: Type of noise

2.6 Type of emergency light

2.6.1 Optical type

In optical type there is 3 type of light, where rotating light contain a single bulb around which a curved mirror is spun, creating a rotating beam of light, which appears to flash when viewed from a stationary position. Rotating lights often use a quartz-halogen or conventional incandescent bulb, though some rotating beacons are now made with LEDs rather than bulbs. The second type is strobe light, based on strobe lights similar to those used in flash photography. It produced a bright flash by ionizing and then discharging a large current through the gas. Others optical type is LED light, they can be programmed with a wider variety of flash patterns because of their ability to be switched directly by electronics, as opposed to discharging a capacitor through a gas-filled tub.

2.6.2 Mounting type

In mounting type is has verities of type, Roof-mounted single beacon is one of the type that is widely used before Light bar were introduce, Light bars that may contain fixed, rotating, strobe, or LED-based lights offering programmable flash patterns where the flash pattern can be control. Other than that Body work mounted is also one type the light where can be mounted on to the outside of the vehicle and these can be used to provide directional lighting in key areas, such as in front for clearing traffic, or to the rear for scene protection. Common places to mount such beacons include on or in the grill of the vehicle and on the front of the rear view mirrors, where they can gain maximum visibility, or as additional lighting such as on the side of the bonnet which helps increase the warning for oncoming traffic when pulling out of junctions. The last type is Interior type that be used in the interior of a vehicle, generally on the dashboard, visor area, or rear deck [6].

2.7 Threshold

Threshold is used to segment an image by setting all pixels whose intensity values are above a threshold to a foreground value and all the remaining pixels to a background value. Threshold is use change image to a binary state where the pixel is 1 and 0. Thresholding is the simplest method of image segmentation. Several different methods for choosing a threshold exist.

Fixed threshold

One alternative is to use a threshold that is chosen independently of the image data. If it is known that one is dealing with very high-contrast images where the objects are very dark and the background is homogeneous and very light, then a constant threshold of 128 on a scale of 0 to 255 might be sufficiently accurate. By accuracy we mean that the number of falsely-classified pixels should be kept to a minimum.

Istogram-derived thresholds

In most cases the threshold is chosen from the brightness histogram of the region or image that we wish to segment. A variety of techniques have been devised to automatically choose a threshold starting from the gray-value histogram, $\{h[b] \mid b = 0, 1, \dots, 2^B-1\}$. Many of these algorithms can benefit from a smoothing of the raw histogram data to remove small fluctuations but the smoothing algorithm must not shift the peak positions. This translates into a zero-phase smoothing algorithm given below where typical values for W are 3 or 5:

$$h_{\text{smooth}}[b] = \frac{1}{W} \sum_{w=-(W-1)/2}^{(W-1)/2} h_{\text{raw}}[b-w] \quad W \text{ odd}$$

Isodata algorithm

The histogram is initially segmented into two parts using a starting threshold value such as $\theta_0 = 2^{B-1}$, half the maximum dynamic range. The sample mean ($m_{f,0}$) of the gray values associated with the foreground pixels and the sample mean ($m_{b,0}$) of the gray values associated with the background pixels are computed. A new threshold value θ_1 is now computed as the average of these two sample means. The process is repeated, based upon the new threshold, until the threshold value does not change any more. In formula:

$$\theta_k = (m_{f,k-1} + m_{b,k-1}) / 2 \quad \text{until } \theta_k = \theta_{k-1}$$

Triangle algorithm

A line is constructed between the maximum of the histogram at brightness b_{max} and the lowest value $b_{\text{min}} = (p=0) \%$ in the image. The distance d between the line and the histogram $h[b]$ is computed for all values of b from $b = b_{\text{min}}$ to $b = b_{\text{max}}$. The brightness value b_o where the distance between $h[b_o]$ and the line is maximal is the threshold value, that is, $\theta = b_o$. This technique is particularly effective when the object pixels produce a weak peak in the histogram [7].

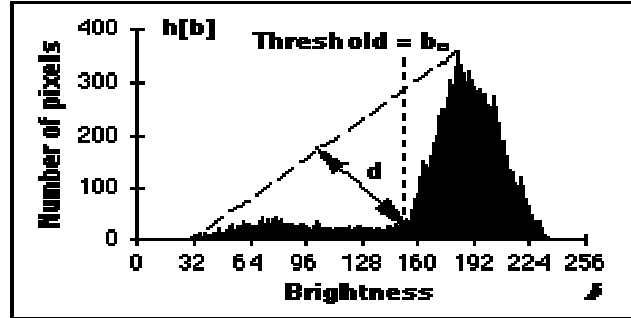


Figure 2.2: The triangle algorithm is based on finding the value of b that gives the maximum distance d .

CHAPTER 3

METHODOLOGY

3.1 Interfacing MIL with Visual Basic

Firstly the Visual Basic 6 software must be installed by following the instruction step. The important setting for this MIL installation show below:-

- i. Development - Matrox imaging library, Active Mil and intelicam
- ii. Hardware device - corono.II, VGA

Full version of MIL has no time limit where hardware like frame grabber or digitizer can be use to interface with MIL. For evaluation software, user can only use it for 30 day and after the duration MIL tools cannot be use anymore and the software must be uninstall. The command of MIL tool is use a Visual Basic language and the ActiveX will be automatically added to the Visual Basic. Installation of MIL must be after the Visual Basic has been installed because if not the ActiveX from MIL will not automatically add. MIL is also can be install at a Visual C++. After the installation Active MIL user guide can be locate at ActiveMIL help where all guided is provided.

3.2 Software structure

A graphical user interface (GUI) is an icon for a program, interaction between user and program is through GUI where all the programming writes in the GUI icon. Use of GUI application will help the beginner user to familiarize with a programming. Consistent GUI also enable user to learn new application faster. The control icon can be added by pointing and clicking the mouse where the icon will be appeared on the form, then the icon can be rearrange the placement by drag the icon in the toolbox. The basic function of toolbox is an important knowledge that needs to understand to develop software. There is a basic description of toolbox in table 3.1 where the control is used to develop the software structure in figure 3.1.

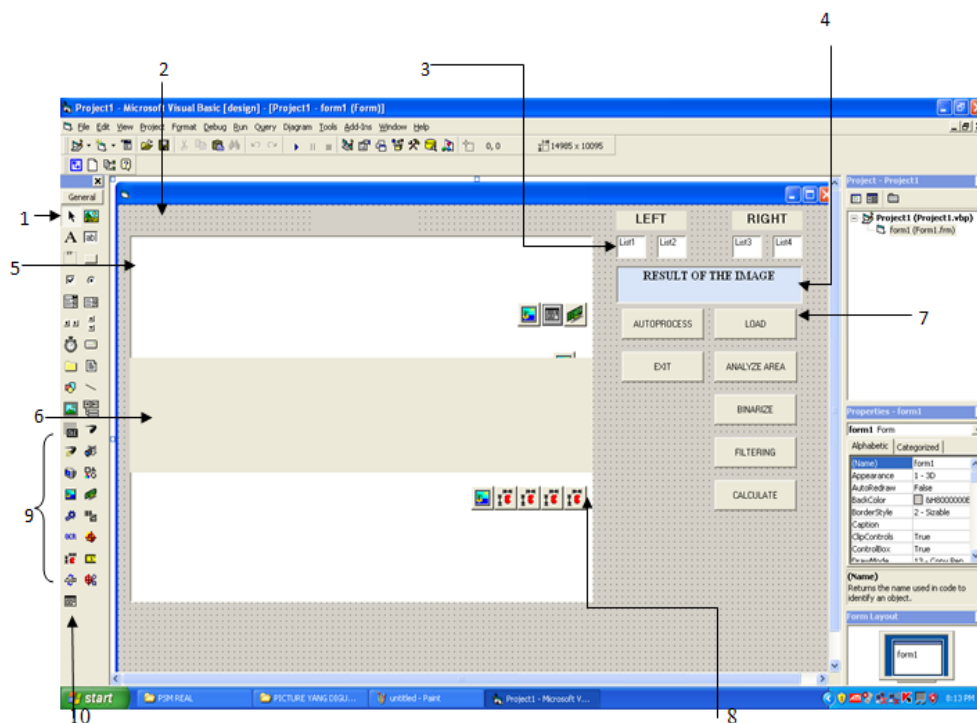


Figure 3.1: Software structure

Table 3.1: Control Description

No.	Control	Description
1	Toolbox	Contain control used to customize forms.
2	Form	Where the program's Graphical User Interface (GUI) will be displayed.
3	ListBox	A control that provide a list of items.
4	TextBox	A control for accepting user input. TextBox can only display text.
5	PictureBox	A control that display image.
6	Display	A control that display image from PictureBox.
7	CommandButton	A control that represents a button. The user presses or clicks to initiate an action.
8	Blob analysis control	A control that Identify connected regions of pixels within an image.
9	Matrox toolbox control	Contain control used to analyze image.
10	CommonDialog	Provide several common dialog boxes such as open and save.

3.3 GUI development

Visual Basic's Integrated Development Environment (IDE) allows the programmers to create and run window program in one application (Visual Basic) without need to open additional program. Where program to create and execute the program is in one application. When Visual Basic is loaded New Project dialog shown in table 3.2 is displayed. The New Project dialog allows the programmer to choose type of Visual Basic program to create. Standard EXE, which is highlighted by default, allow the programmer to create a standard executable. Software development in this project must be build by following step order. Firstly Standard EXE is selected and Open button is press.



Figure 3.2: New Project dialog

Figure 3.3 shows the IDE after Standard EXE is selected. A Standard EXE project contains following windows:

- i. Project1-Form1 (Form)
- ii. Form layout
- iii. Properties – Form1
- iv. Project – Project1
- v. Toolbox

The Project1-Form1 contains a form that use to display a GUI. Control icon is a portion for program where the programming is written in the icon and the result were displayed on form. The form size must be resized by modify the height and width where the scale mode is in a pixel sized at the property form.

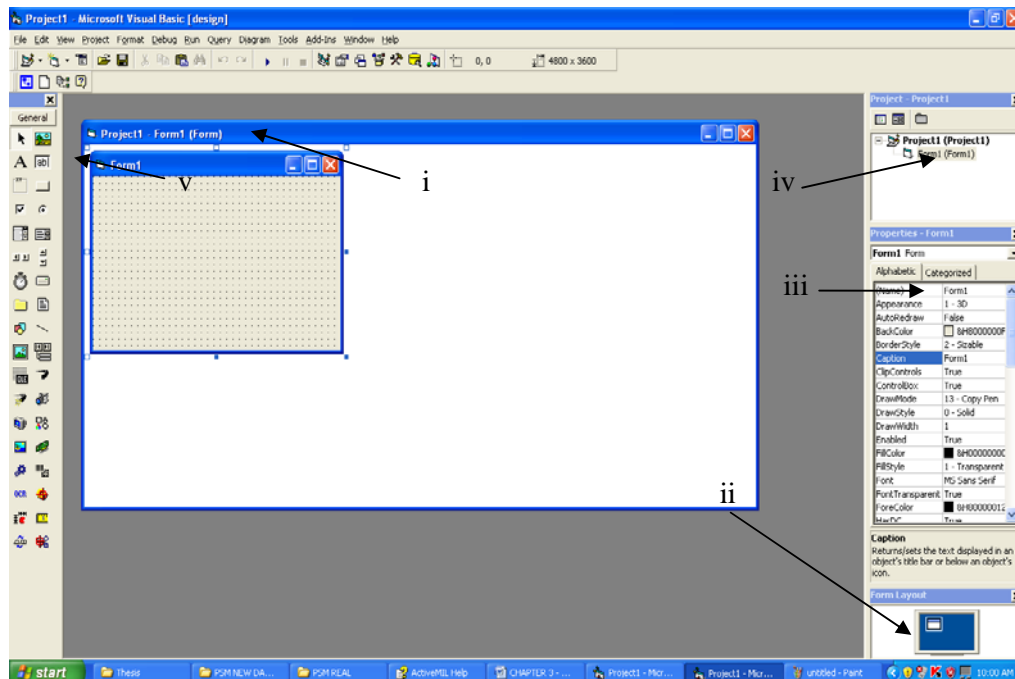


Figure 3.3: IDE with a Standard EXE project open

The properties window (Figure 3.4) shows the properties for a form or control. Properties are attributes such as appearance, back color, size and lot more. Properties are listed either alphabetic or categorized. Alphabetic is a default condition where list of properties can be scroll by using a scrollbar.

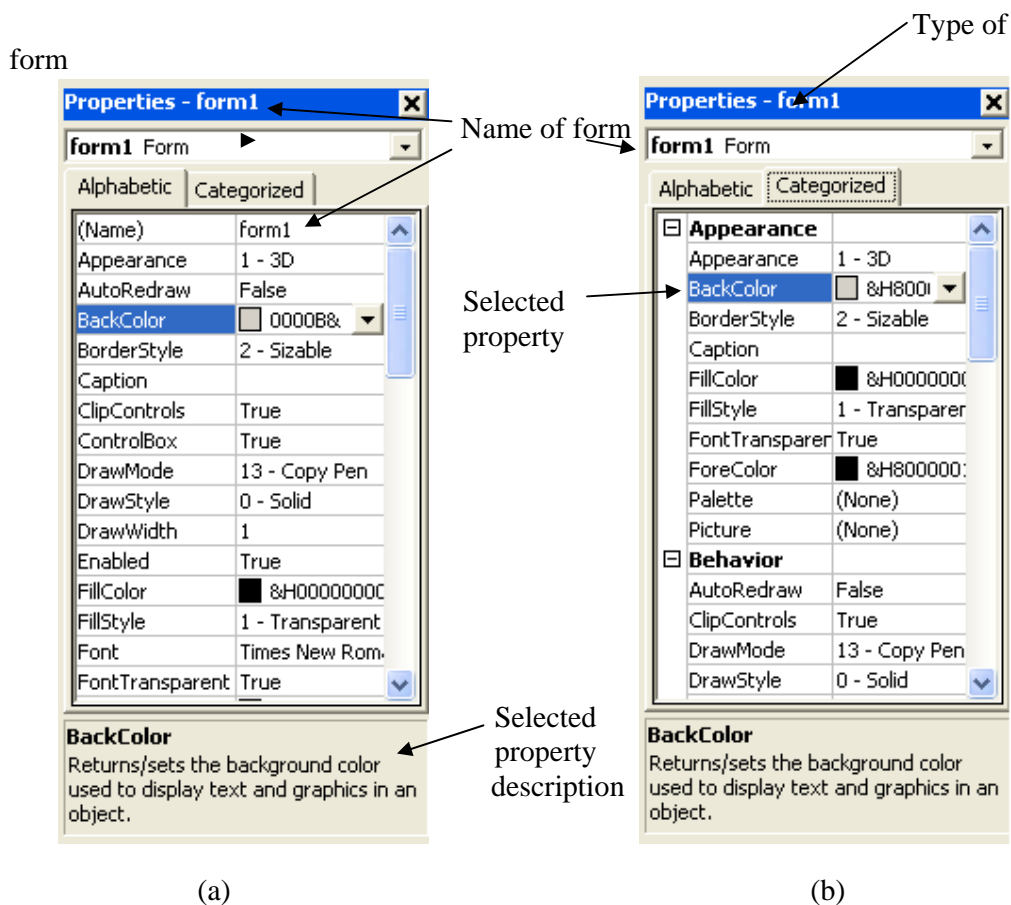


Figure 3.4: Properties window with (a) alphabetic and (b) categorized tabs.

The Form layout window (Figure 3.5) shows a form position on the screen at run time. It consists of an image representing the form relative position on the screen and screen. The form position on the run time can be changed by dragging the form image to the required position.

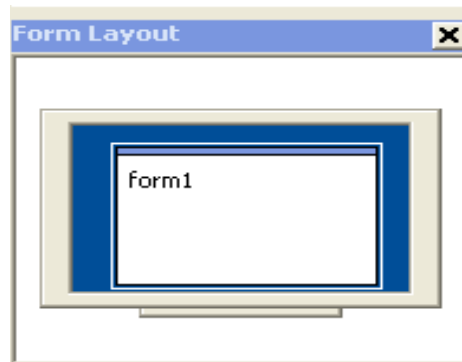


Figure 3.5: Form layout window

Visual Basic control exists in two varieties, intrinsic control and ActiveX control. Intrinsic control or called standard controls are the default control provided in the Visual Basic toolbox.

Secondly after the form has been resized ActiveX control must be loaded into the Visual Basic with the Components dialog (Figure 3.6). In this project some ActiveX control must be added to the toolbox by selecting Components menu's in the Project menu. ActiveX control must be selected by pressing OK to load the control icon to the toolbox. There are list of ActiveX control component that added in this project.

- i. ActiveMil Base Control
- ii. ActiveMil Blob Analysis Control
- iii. ActiveMil Image Processing Control
- iv. Microsoft Common Dialog Control 6

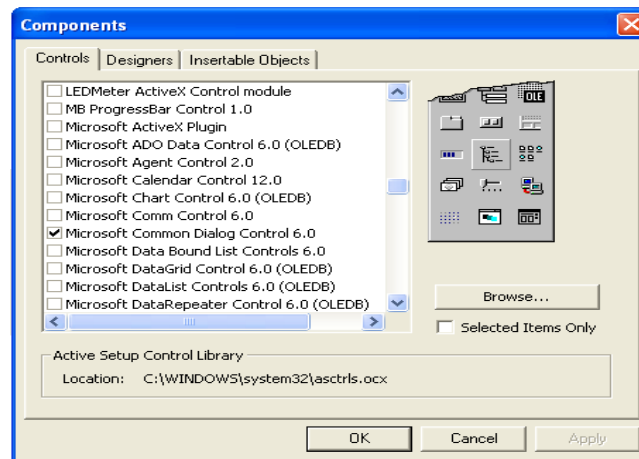


Figure 3.6: Component dialog

3.3.1 Picture Box

After the ActiveX control is added, the Picture Box control is added to the form by double clicking the Picture Box control and it will automatically added to the form1. The property setting is illustrated in table 3.5.

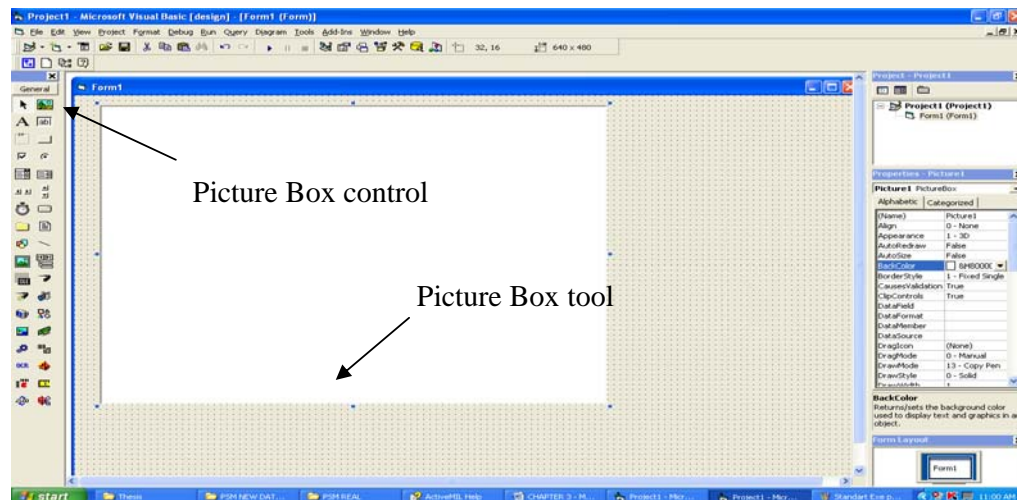


Figure 3.7: Load Picture Box into Form1

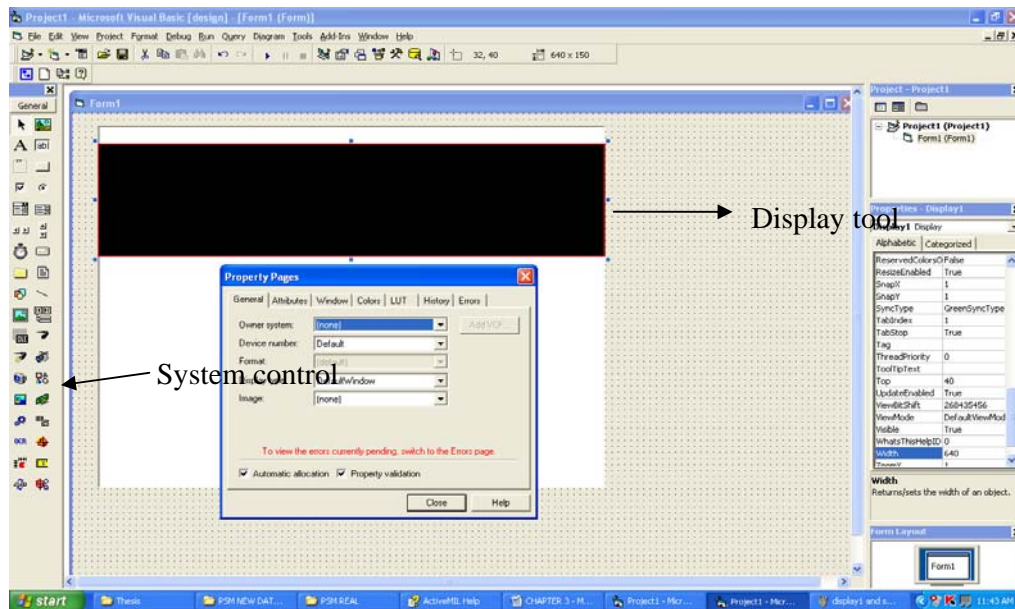


Figure 3.8: Load Display into Form1

3.3.2 Display and System

Third step is to added the display to the form, by double clicking the Display control it will inserted to the form where the property were shows at table 3.5. The ActiveX control must have a System to control the operation. If System control is not attach, the Display1 will surrounded by red line (Figure 3.8).The error can be solve by insert the System control and owner system property for Display1 has to be named by the System tool name (System1). System type that is use is VGA where it can be setting at system tool property.

3.3.3 Image control

The Image control is used to store an image that has been processed where the programming is written in the command button. Figure 3.9 shows an image tool that has been inserted by double-clicking the control icon. Image property (Figure 3.10) is important to setting the analyzed image, parent image is an image that is used as a reference for the child image. Size of image needs to be specified to get a full size of the image and child image can crop the image by setting the child region by calculating distance of pixel image using the formula below. Band is a number of components in the image and for a grayscale image band is one and three for a color image because it has red, green, blue components in the image. Image properties of the project were described in Figure 3.2.

Child region calculation

- i. $\text{Size}(X, Y) = (X = X_{\max} - X_{\min}, Y = Y_{\max} - Y_{\min})$
- ii. $\text{Offset}(X, Y) = (X = X_{\min}, Y = Y_{\min})$

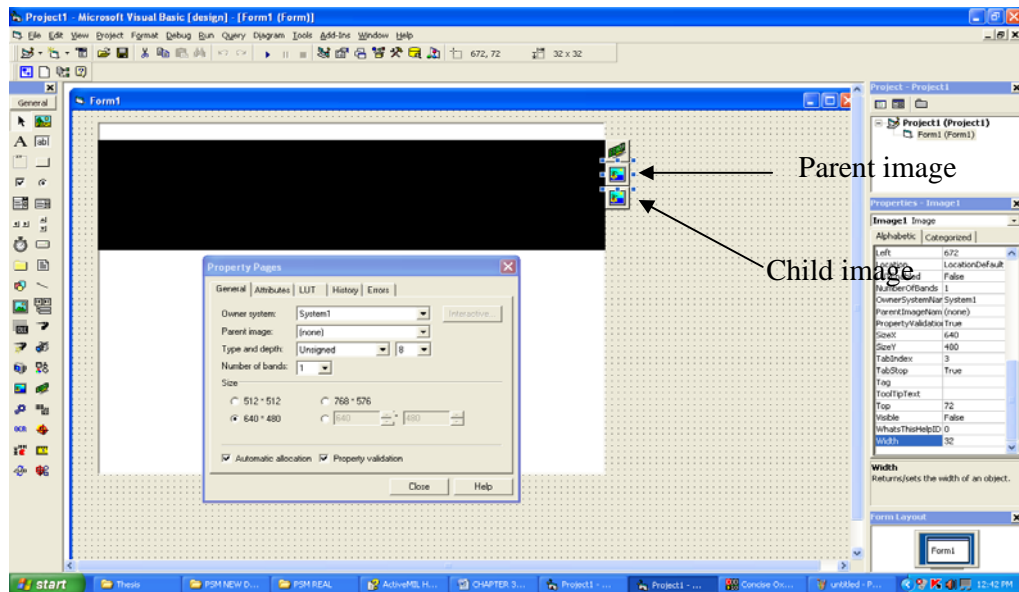
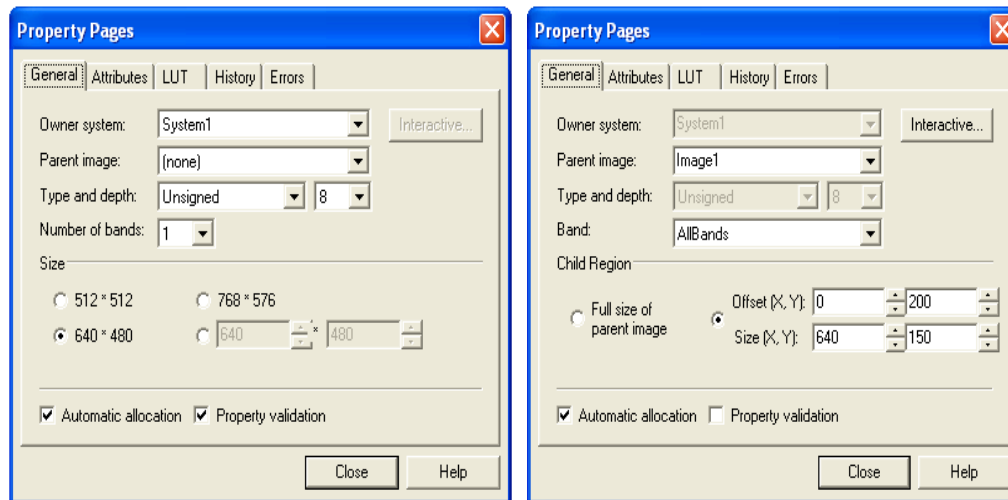


Figure 3.9: Image control property

Table 3.2: Image property

Image Name	Parent Image	Child region (pixel)			
		size		Offset	
		X	Y	X	Y
Image1	-	640	480	0	0
Image2	Image1	640	150	0	200
Image3	Image2	Full size of parent image			
Image4	Image3	Full size of parent image			



(a)

(b)

Figure 3.10: Property of image (a) Parent Image and (b) Child Image

3.3.4 Command Button

A Command Button (Figure 3.11) is create in this form is operated in two order, manually and automatically. Manual operation is use to illustrate how the software operate and the automatic operation demonstrate the full operation by just load the picture and result of analyze image is displayed. In each common button have own programming command and property of Command Button is illustrate in table 3.5. Command Button dialog tool is needed to open and save file. Figure 3.12 shows an example of load command.

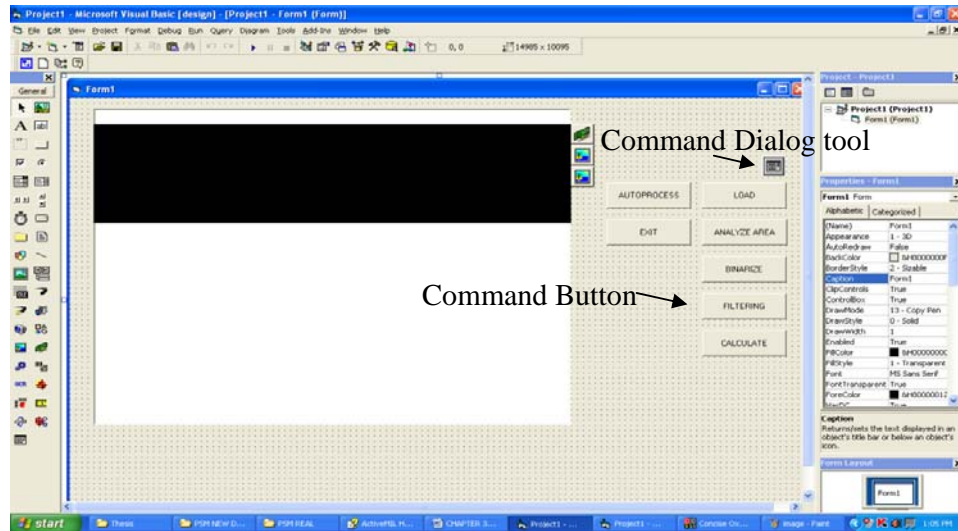


Figure 3.11: A Command Button

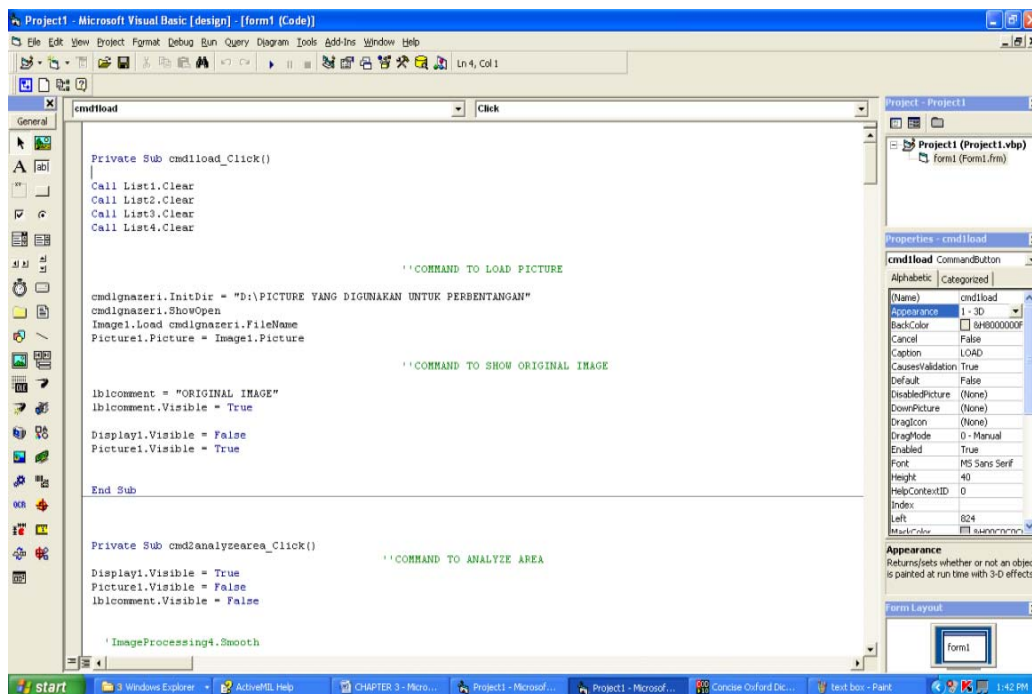


Figure 3.12: An example of load command

3.3.5 Text Box

The text box is use to show the result of the image where the result image is depend on the value of List box that will be explain in the List Box part. The result shows the image is in the emergency light vehicle of non-emergency. Property of the text box is described in the table 3.5.

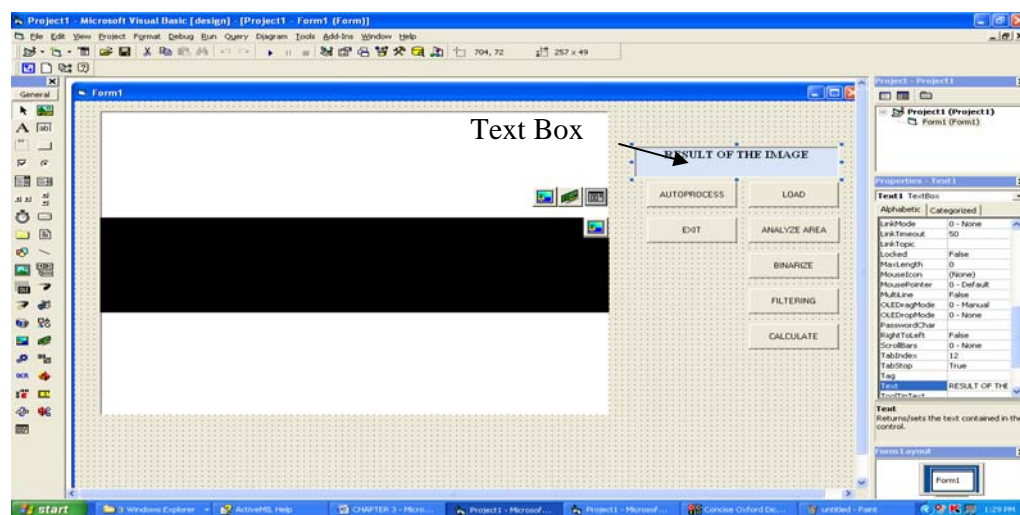


Figure 3.13: A text box

3.3.6 List Box

In project form (Figure 3.14) there is four type of list box where it is divided into two parts, Left (List1, List2) and Right (List3, List4). The list box show result of blob calculation in each blob where table 3.3 shows list of blob that is count. The list box name cannot be change randomly because it has individual command operation. Table 3.4 demonstrates probability of blob appeared on list.

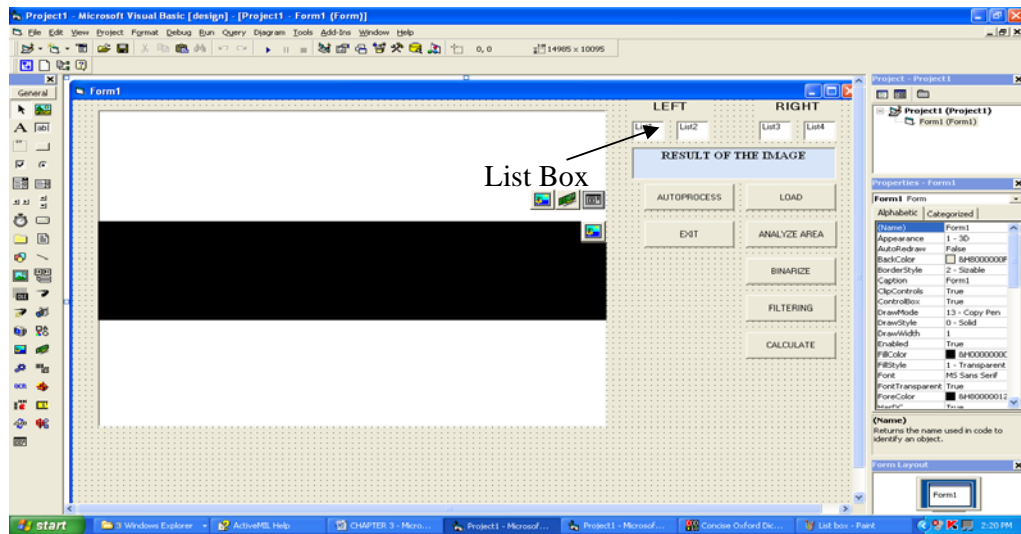


Figure 3.14: A List Box icon


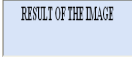

Table 3.3: Table of Listbox


Area	Name of List box	Name of Blob
Left	List1	BLOB
	List2	BLOB3
Right	List3	BLOB1
	List4	BLOB2

Table 3.4: Probability Value of blob

No.	Left 1 List1 (L1)	Left 2 List2 (L2)	Right 1 List3 (R1)	Right 2 List4 (R2)	Result
1	0	0	0	x	Non-emergency
2	0	0	1	x	Emergency
3	0	0	>1	0	Non-emergency
4	0	0	>11	1	Emergency
5	0	0	>1	>1	Non-emergency
6	1	x	0	x	Emergency
7	1	x	1	x	Emergency
8	1	x	>1	0	Emergency
9	1	x	>1	1	Emergency
10	>1	x	0	x	Non-emergency
11	>11	x	1	x	Emergency
12	>1	x	>1	0	Non-emergency
13	>1	x	>1	1	Emergency
14	>1	x	>1	>1	Non-emergency

Table 3.5: Table of property listing for the program

Object	Icon	Property	Property setting	Property description
Form		Name	Form1	Identifies the form
		Caption		Form title-bar display
		Height	10095	Size of the form
		Width	14985	
		Back color	&H80000013&	Color of background
Command Button		Name	Cmd1load	Identifies the CommandButton
		Caption	Load	Text that appear on button
		Height	40	Size of command button
		Width	113	
		Fond	MS Sans Serif Regular ,8pt	Commandbutton text fond
TextBox		Name	Text1	Identifies text box
		Text	RESULT OF THE IMAGE	Text that appears on text1.
		Fond	Times New Roman Bold ,12pt	Fond for text
ListBox		Name	List1	Identifies the List Box

Label		Name Caption Fond	Label1 LEFT MS Sans Serif Regular,12pt	Identifies the Label1 Text Label1 display Label1 text font
Picture Box		Name Height Width Scale Mode Back Color	Picture1 480pixel 640pixel 3- pixel Active title bar text	Identifies the picture box Size of the picture box Scale of the picture box Color of the picture box
Display		Name Height Width Owner system Back color	Display1 150pixel 200pixel System1 &H00597900&	Identifies the display Size of the display Background color of Display
System		System type	VGA	System control the whole ActiveX control

3.3.7 Image Processing Control

In this property (Figure 3.15), owner system that use is system1 where the control must be inserted into the form. Control system in each form must have at least one, because system will control the whole operation in this software. Image processing methods support images of any data type or depth and all the process of image processing will use image from source and will be stored in the image at the destination that has been specified in the property pages. By adjusting the result size it will increase speed of calculation. Image processing in this development use two image processing where it use to change the image into binary state and filtering the image by remove noise and smoothing the image. Table 3.6 illustrates the property setting of this software.

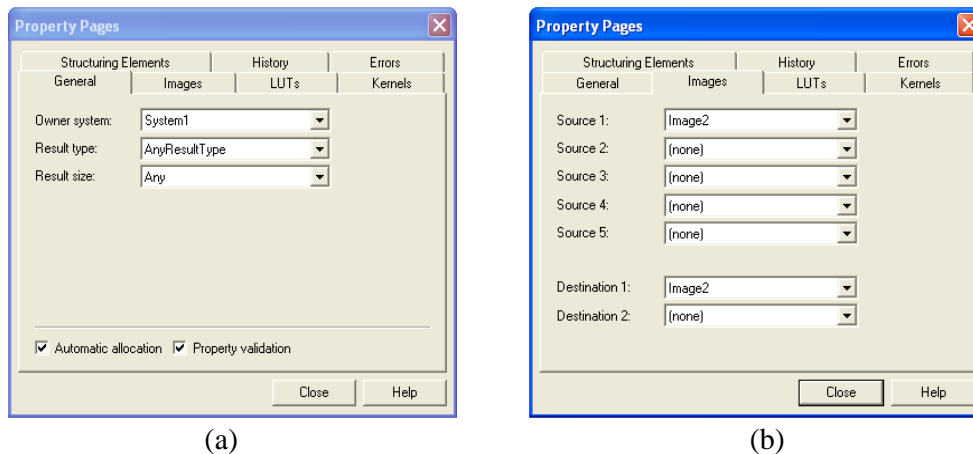


Figure 3.15: Image processing property (a) General and (b) Image

Table 3.6: Image processing property setting

Name	Owner system	Source 1	Destination 1	Destination 2
Image Processing 1	System1	Image 2	Image 2	-
Image Processing 2	System 1	Image 3	Image 3	Image 4

3.4 Blob analysis identification

Blob analysis can be used to identify connected region pixel within an image and calculated the selected feature of those region. Blob analysis use information of blob to make a calculation where blob is defined as areas of touching pixel that are in the same logical pixel state. The pixel state is called the foreground state where it has zero and non-zero, while the alternate state is called the background state. Foreground zero considered black area as a blob while non-zero assuming white area as a blob, whereas figure below shows example of emergency light blob that use a foreground zero (Figure 3.16) and non-zero (Figure 3.17).

As shown in Figure A and Figure C there is information that use to considered the blob. FeretY is measured by using maximum length of the blob and FeretX is use maximum breadth show in Figure C. Area and perimeter can be determined by using formula below which the distances were calculated using pixel value:-

- i. $\text{Area} = \text{Length} \times \text{Breadth}$
- ii. $\text{Perimeter} = 2(\text{Length} + \text{Breadth})$

The area and perimeter is calculate by assuming all probably that has in the image of emergency light that been measured. In this project police petrol car is use as a reference where there is two type of emergency light that using an old version the new version. This 2 type of model has a difference size and refraction color which will affect the calculation complexity.

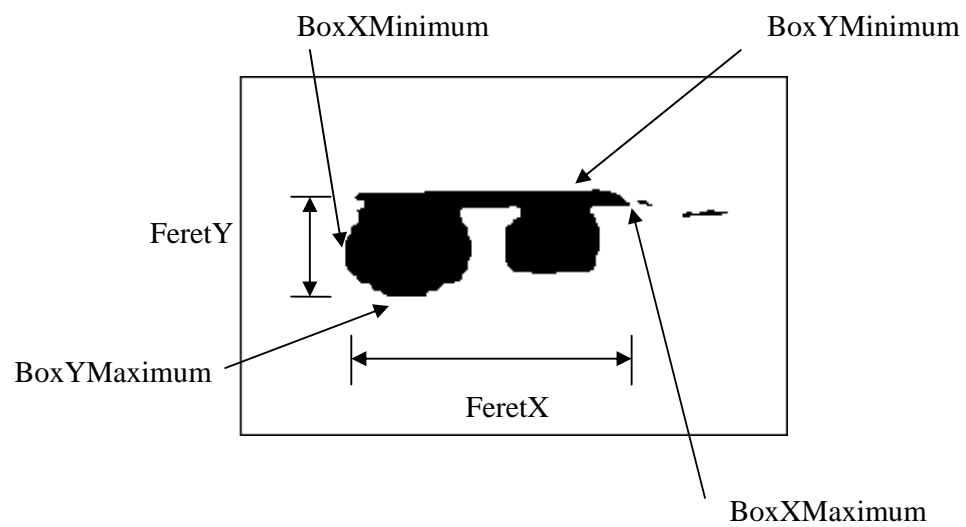


Figure 3.16: The size of blob with foreground value = zero

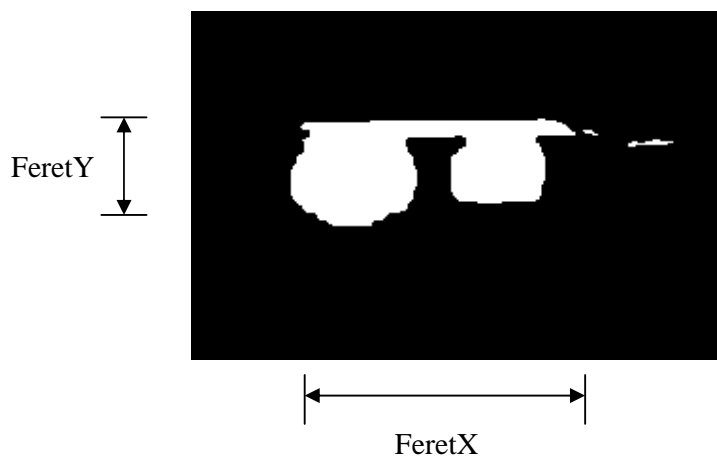


Figure 3.17: The size of blob with foreground value = Non-zero

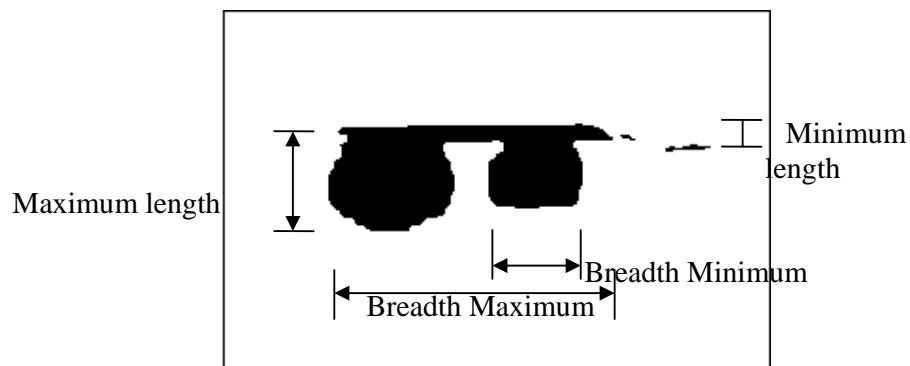


Figure 3.18: The size of blob with foreground value = zero

Identifier blob type is use to identify the condition of blob that want to be counted using the given information. Once the blob has been calculated the result will be counted and displayed on the ListBox. Table 3.7 show the description of identifier blob type where figure 3.19 an example of labeled blob.

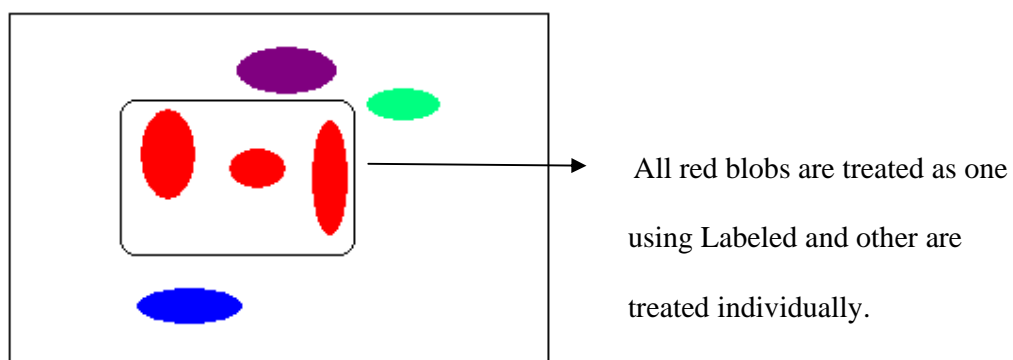


Figure 3.19: Example of Labeled blob

Table 3.7: Description of Identifier Blob Type

Condition	Description
Individual	Feature calculations are performed on each blob individually.
Entire image	All blobs are grouped together and treated as one blob, where feature calculations are performed on this blob.
Labeled	Feature calculations are performed on blobs with the same label.

Figure 3.20 show property that must be taken into account where all the features that we selected can be exclude, include and delete by selecting filter that we want to use. The setting for the blob could be set inside the Blob Analysis Control tool's property.

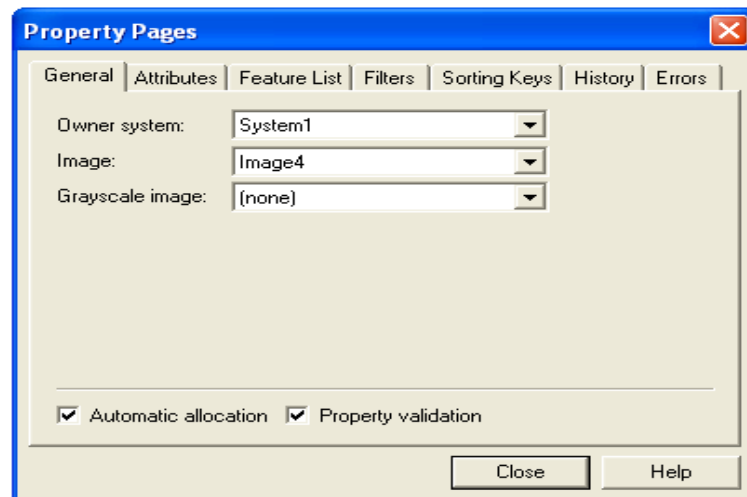


Figure 3.20: Blob analysis control property

3.5 Blob Analysis Property setting

The step to open blob analysis property is by right clicking at the Blob Analysis Tool and then Blob Analysis form will be displayed where properties must be clicked to view the content. In the table 3.8 shows the attributes property that used in the software development. The foreground value is zero where the black area use is assume as a blob, pixel type of image processing is binary state where the image is convert to binary a BINARIZE command button. For number of feret angle minimum number is 2 and the maximum is 64, in a default condition number of feret angle is 8.

Table 3.8: Attributes property setting

Name	Setting
Identifier blob type	Individual
Identifier pixel type	Binary
Lattice	8connected
Number of feret angle	8
Pixel Characteristics	
Foreground value	Zero
Aspect ratio	1

Table 3.9 show the setting for a feature list where in can be find in Blob Analysis tool property, feature list setting is to initiate program to calculate the selected features only where the setting must be set to true. Filter that set in table 3.10 is use to filter the information given, the blob can be include, exclude or delete. The value of blob is calculated outside of the program where image that has been filtered is saved as an image file. Image file is calculated manually by using Adopt Photoshop CS2 and the image is analyzed by using various conditions where the maximum and minimum value of each blob must be considered.

Table 3.9: Feature list property setting

Blob name	Feature list	Feature state
BlobAnalysisFrame	-	-
BLOB	Area Perimeter	True
BLOB1	Area Box	True
BLOB2	Area Box	True
BLOB3	Area Box Perimeter	True

Table 3.10: Feature property setting

Blob Name	Filter Type	Feature	Condition	Low Value	High Value
BlobAnalysisFrame	-	-	-	-	-
BLOB	Exclude	Area	LessThan	1225	-
	Exclude	BoxXMinimum	LessThan	32	-
	Exclude	BoxXMaximum	GreaterThan	320	-
	Exclude	BoxYMinimum	LessThan	18	-
	Exclude	BoxYMaximum	GreaterThan	150	-
	Exclude	FeretX	InRange	0	5
	Exclude	FereyY	OutOfRange	21	58
	Exclude	Perimeter	LessThan	80	80
BLOB1	Exclude	BoxXMinimum	LessOrEqualTo	350	-
	Exclude	BoxXMaximum	GreaterThan	632	-
	Exclude	BoxYMinimum	LessThan	4	-
	Exclude	BoxYMaximum	GreaterThan	150	-
	Exclude	FeretX	OutOfRange	36	261
	Exclude	FeretY	OutOfRange	27	122
	Exclude	Area	Area	1225	-
BLOB2	Exclude	BoxXMinimum	GreaterThan	350	-
	Exclude	BoxXMaximum	LessThan	632	-
	Exclude	BoxYMinimum	LessThan	4	-
	Exclude	BoxYMaximum	GreaterThan	115	-
	Exclude	Area	GreaterThan	1700	-
	Exclude	FeretX	OutOfRange	28	32
	Exclude	FeretY	OutOfRange	43	49
BLOB3	Exclude	Area	LessThan	1225	-
	Exclude	FeretX	GreaterThan	246	-
	Exclude	FeretY	GreaterThan	122	-
	Exclude	BoxXMinimum	LessThan	35	-
	Exclude	BoxXMaximum	GreaterOrEqual	260	-
	Exclude	BoxYMimum	LessThan	9	-
	Exclude	BoxYMaximum	GreaterThan	115	-
	Exclude	Perimeter	LessThan	330	-

3.6 Binarize Method

In this Binarize method image from the analyze area is in a grayscale type where the calculation time duration for grayscale image is increase and the calculation is much complex. Binarize method is changing the image to a 0, 1 state where the image is black and white. The images were change to binary state by threshold the image using threshold value 120.

3.7 Filtering Method

Filtering method is use to remove the noise and smoothing the image where the image must be in a good condition where it will decrease the calculation length. In this step the image use rank and smooth as a filtering device. In this operation the threshold value that been used is 120. Where Adopt Photoshop CS2 is used by try each of the threshold value to get the best threshold image.

3.8 Analyzing Image

Image that wish to be calculate must be analyzed where Figure 3.21 show example of original image. The image size is 640x480 pixel and from that size analyze area image (Figure 3.22) is selected. The picture must be change into binary state (Figure 3.23) and than the image is filtered (Figure 3.24). From the image result of filtering, image is being analyzed and the area of blob is identified .If the area have a value of blob greater than 1 at BLOB1 then calculation will use BLOB2 filtering. While if result of blob count at BLOB equal to 0 then blob analysis will use BLOB3 to calculate the selected area. After the whole calculation the result will be display at textbox. Figure 3.25 is an example image of non-emergency light show where the blob is not counted as an emergency light blob.



Figure 3.21: Original Image



Figure 3.22: Child image for analyze area



Figure 3.23: Binarize image

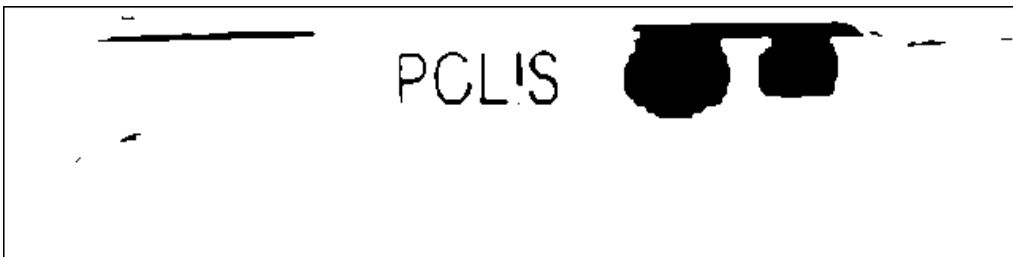


Figure 3.24: Filtering image

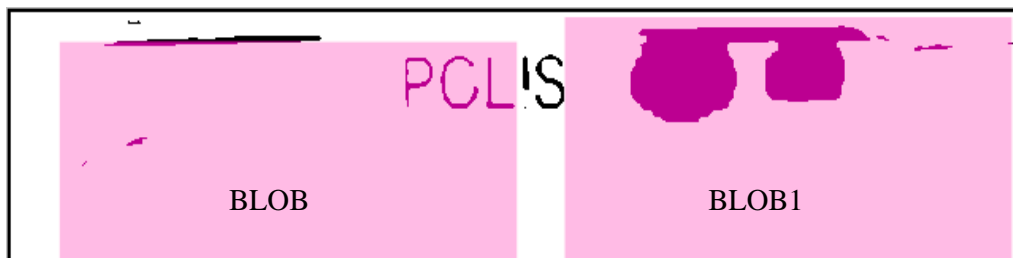


Figure 3.25: First calculations of blob

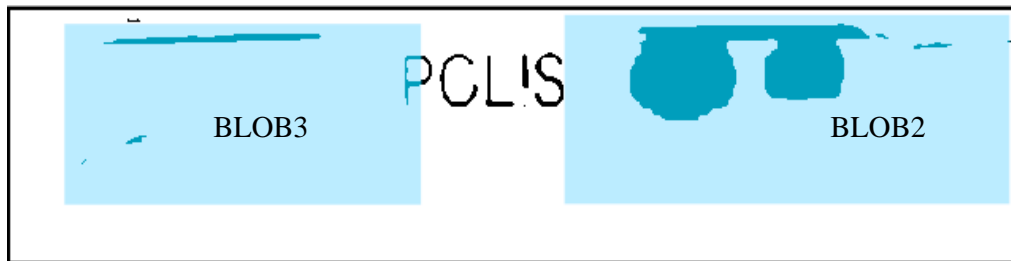


Figure 3.26: Second calculation of blob



Figure 3.27: Non emergency light filtered image

3.9 Operation

In this software development Visual Basic 6 were use to make a GUI and Matrox is an extended toolbox where all image analyzing is done by using Matrox. The image were taken by using SONY video camera to take the image, emergency vehicle image can not be taken by using camera because the speed of grabbing image is lower than video camera which can cause lot of information from the emergency vehicle can be lost. Video image that has been taken then were convert into a static image by using a Power Video Converter where the picture is extract into 5 pictures per second. Figure 3.28 shows the full operation of the software where the operation is start by load the image from specified folder where the picture of emergency light is save. The image that been use then were analyze to find the specified analyze area, where the emergency vehicle must be in the range of analyze area. If the emergency light outer the range then it will not take into account.

Image then were convert into a binary state where the image at a zero (black) or non-zero (white) state it is because grayscale image cause the calculation much more complex and it will increase calculation duration. Selected image must be filtered to remove noise in the image where it can cause disturbance to calculation operation. After all the process, image were analyze by using blob control that has been provide in the Matrox Imaging Library where it were identify connection regions of pixel within the image. Filter setting will be used to instruct computer to either include or exclude specified blob dimensions. By doing so unnecessary blob will not be taken into calculation. In the calculation the first calculation is calculate using BLOB and BLOB1 where if the counted blob at BLOB is 1 then the image is in an emergency state. If BLOB is equal to 0 then the filter will use BLOB3 to calculate the selected area and the result counted show 1 then it is a emergency light if not the text show opposite. For BLOB1 text show emergency if counted result equal to 1 and if not BLOB2 is use as filter replacing BLOB1 for the calculation. The result will be display on a computer screen whether the emergency vehicle in an emergency of non emergency state by calculating the result of each list box. Figure 3.28 show the full window of developed software.

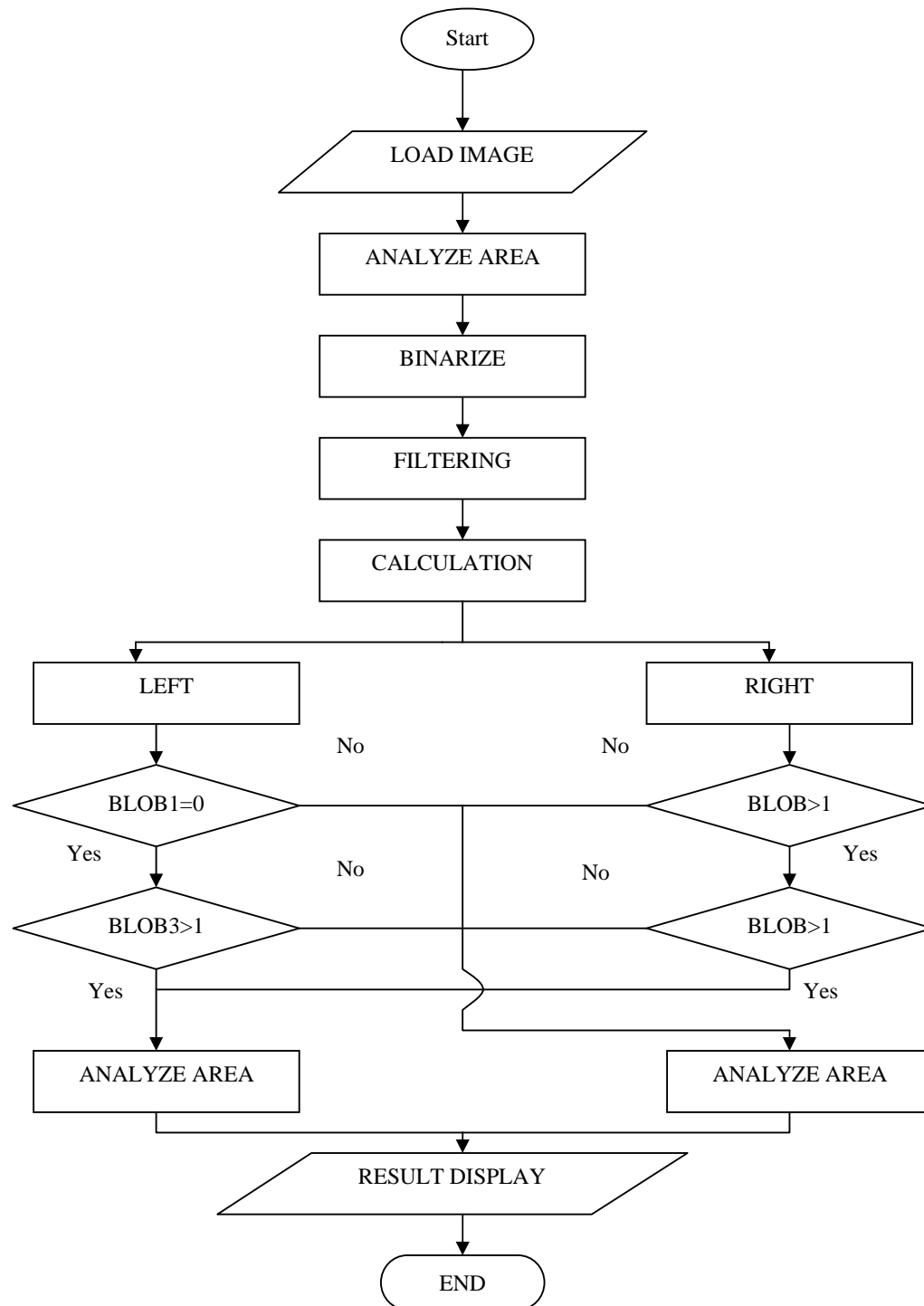


Figure 3.28: Flow chart of the operation

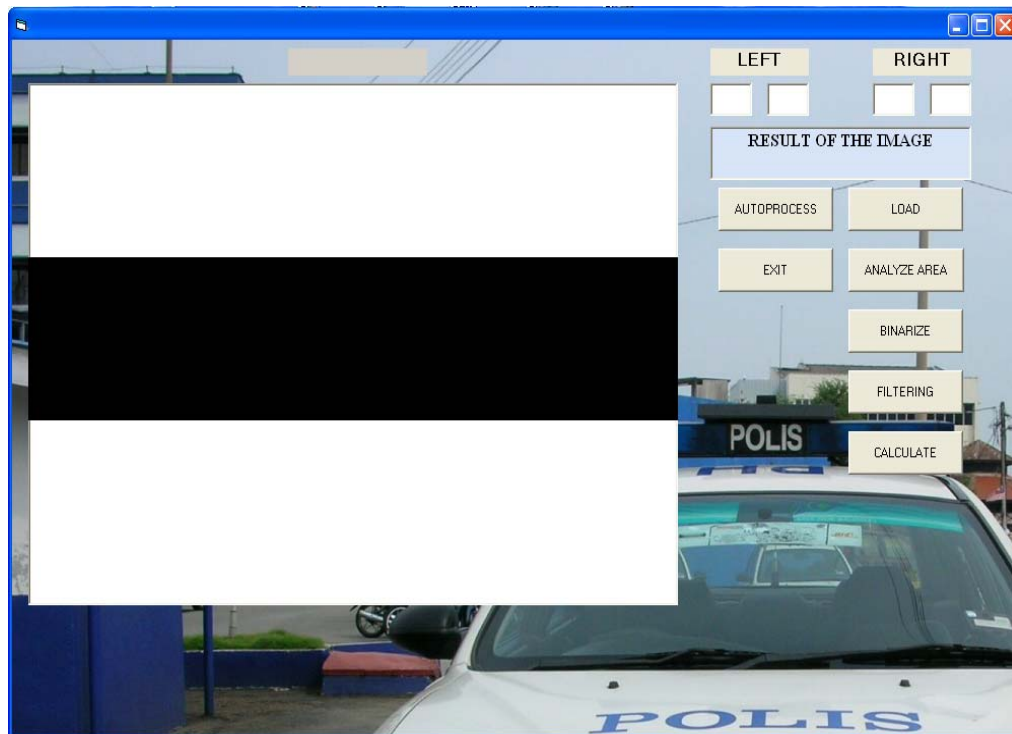


Figure 3.29: Software window

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter will discuss on the result and discussion of the image of police petrol car for the software development.

This chapter is explaining the result of image processing which covering those part:-

- i. Probability variable of emergency light
- ii. Filtration result
- iii. Binarized result
- iv. Performance of image processing

4.2 Probability variable of emergency light

Development of the project uses 2 model of emergency light for a police car, which old model and new model. The old model has 3 conditions of emergency light shown below:-

- i. Non emergency
- ii. Emergency light left side
- iii. Emergency light right side

An emergency light appeared on the left side is using BLOB and BLOB3 to calculated the blob on the left side. Where in the left side given the last result of the calculation is 1. While for emergency light on the right side is calculated with BLOB1 and BLOB2, the result is use the same calculation method as on the left side

4.2.1 Old model of emergency light



Figure 4.1: Original image of non-emergency light



Figure 4.2: Result of figure 4.1- non-emergency light



Figure 4.3: Original image of non-emergency light- blob left



Figure 4.4: Result of figure 4.3- emergency light



Figure 4.5: Original image of emergency light- blob right



Figure 4.6: Result of figure 4.5- emergency light

4.2.2 New model of emergency light

Calculation use in this new model of emergency light is the same with old model of emergency light. Whereas it has 3 condition, but in this emergency light the blob has a difference size of blob and the refraction color of emergency it not touching each other because it will cause information that need to be inserted need to be considered the maximum length of the old model emergency light. The consideration of maximum area and minimum area must be taken because the filtration without taken probably of blob size can cause losing blob that is search.



Figure 4.7: Original image non-emergency light



Figure 4.8: Result of figure 4.7- non-emergency light



Figure 4.9: Original image of emergency light- blob left



Figure 4.10: Result of figure 4.9- emergency light



Figure 4.11: Original image of emergency light- blob right



Figure 4.12: Result of figure 4.11- emergency light

4.2.3 Emergency light exceed range

In this condition the emergency light vehicle is considered non-emergency light, it is because blob that outer of the range is touching or outside the box limit.



Figure 4.13: Original image exceed range



Figure 4.14: Result of figure 4.13- non-emergency



Figure 4.15: Original image of emergency light



Figure 4.16: Result of figure 4.15- non emergency light

4.3 Binarized result

In this filtration method the threshold condition is replaces pixel values less than or equal to the low value with the highest image value possible, while other pixels are set to zero. The whole image in this filtration method is shown a comparison between original (top) and filtered image (bottom).

4.3.1 Old model



Figure 4.17: Non emergency light



Figure 4.18: emergency light on the left side



Figure 4.19: Emergency light on the right side

4.3.2 New model

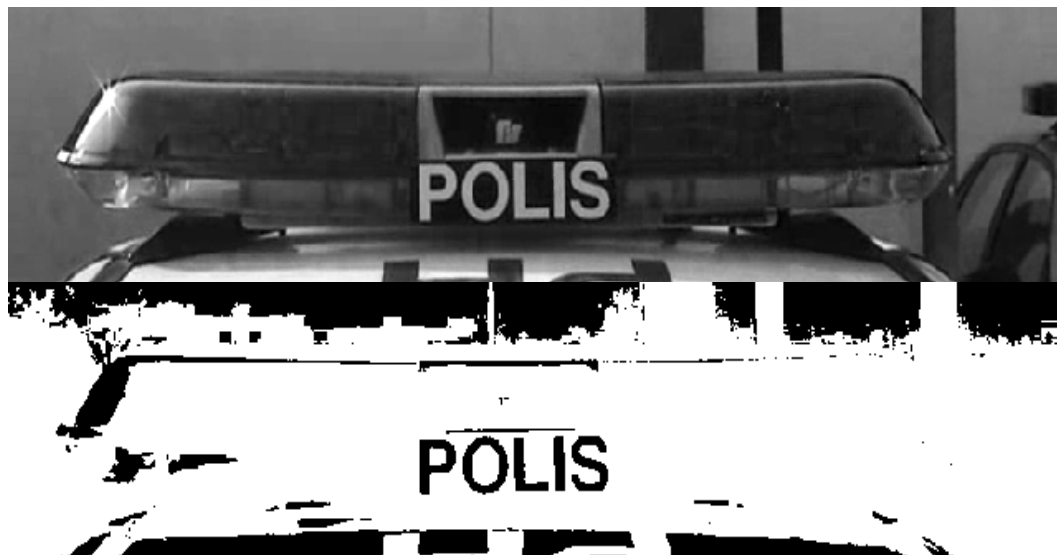


Figure 4.20: Non-emergency light

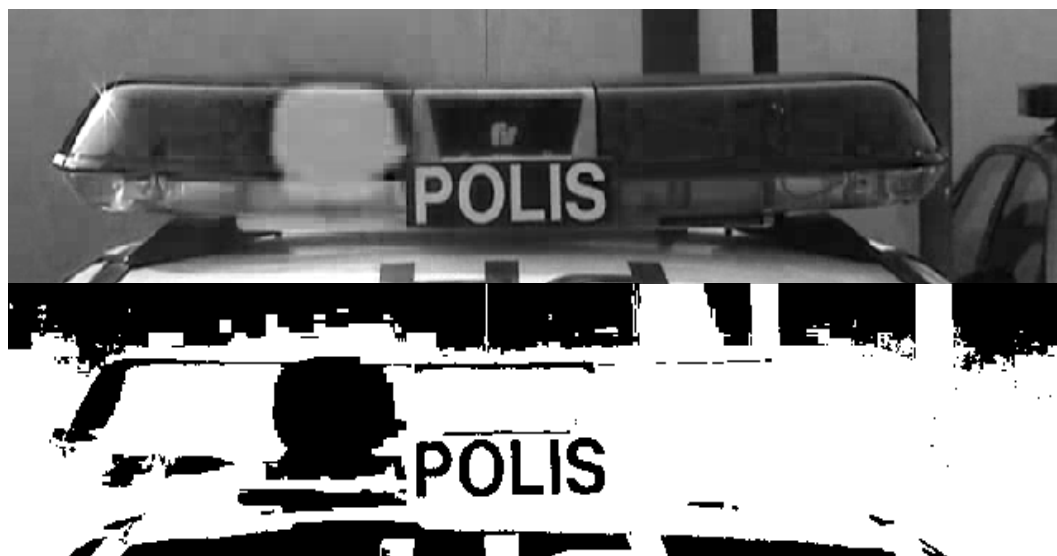


Figure 4.21: Emergency light on the left side

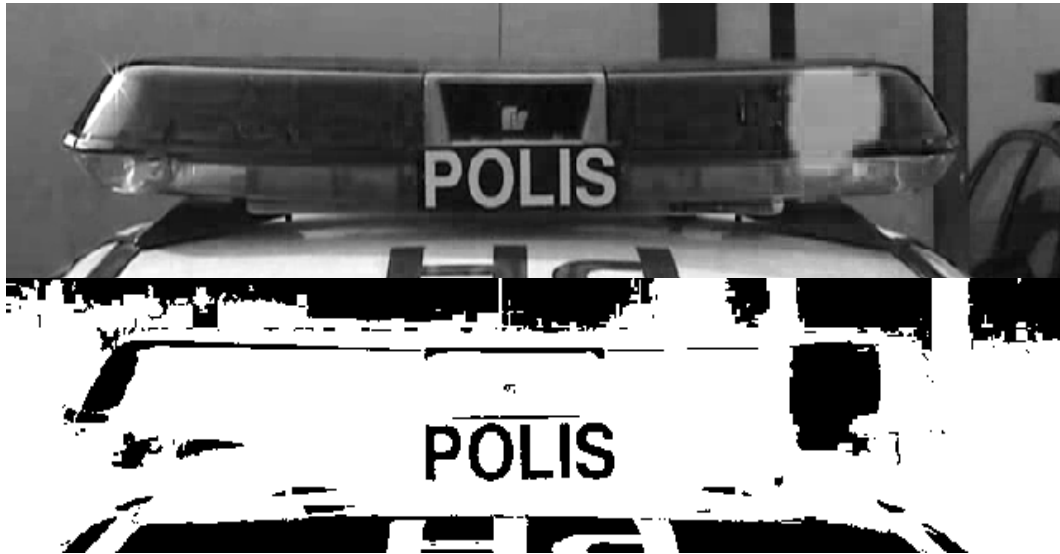


Figure 4.22: Emergency light on the right side

4.4 Filtering Result

Filtering is used to remove noise and smoothing the image, where the operation is done to get the best image. Therefore filtering is an important part before the image is being analyzed using Matrox tools.

4.4.1 Old model

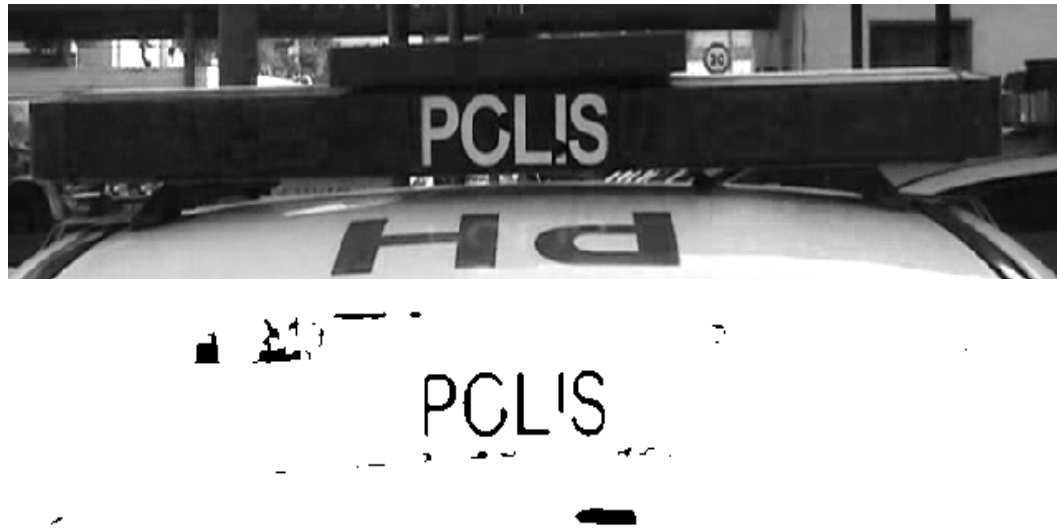


Figure 4.23: Non emergency light



Figure 4.24: Emergency light on the left side



Figure 4.25: Emergency light on the right side

4.4.2 New model



Figure 4.26: Non emergency light



Figure 4.27: Emergency light on the left side



Figure 4.28: Emergency light on the right side

4.5 Performance of image processing

Performance of analyzing image is shown in the table below, the program is able to detect all condition of analyze image which shown that the software has none error the program. Image that exceeds the limit is considered as a non-emergency is proved where the images that exceed the range have been counted as an on-emergency.

Table 4.1: Data of analyze image

Model	Emergency Light	Non-emergency light	Total
New	13	11	24
Old	19	28	47
Exceed range	1	1	2
Total	33	40	73

Table 4.2: Result of analyze image

Model	Emergency Light	Non-emergency light	Total
New	13	11	24
Old	19	28	47
Exceed range		2	2
Total	32	41	73

4.6 Summary of the result

The performance results prove that the software that has been developed is operated accurately where program can analyze police patrol car that use the same model of emergency light. All steps that are implement in this project must be analyze correctly to prevent from error result in the project.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Summary of the work

Various condition of emergency light is taken from police petrol car, which the system is able to differentiate between emergency light and non emergency light. Therefore the software is fully operated can be use without error on analyzing police petrol car. Furthermore the objective of this project is achieved, where to study image processing using Visual Basic 6 and Matrox in developing the software to recognize emergency light and non-emergency light which is successfully differentiate the image.

5.2 Recommendations

For future plan of this project, it is recommended to develop the full camera system which the software can be operated in a real time where it has to be interface with the frame grabber. The system should be apply to the traffic light to recognize police petrol car which will be helpful to them to doing their job without has to worried about traffic problem. The system has an advantages where the traffic light will be operated with a computer and the software can be modified functioning with lot more function to be implement to the software, where all the image analyzing use only one camera system. Therefore by using Matrox user can develop software to read car registration number and if it been implement to this software it surely helping police by have an extra function where wanted car can be found by finding the car registration number at the traffic light. This Matrox software is also a useful utility where it should be implement to the industrial because Matrox is an image processing tool that has been simplified to help people to analyzing image.

REFERENCES

- [1] Wikipedia Org. Visual Basic. http://en.wikipedia.org/wiki/Visual_Basic, Accessed on 16th Jun 2007.
- [2] Matrox Electronic System Ltd., Matrox Imaging Library Version 7, User Guide, Canada, 2002
- [3] Wikipedia Org. Image Processing. http://en.wikipedia.org/wiki/Image_processing, Accessed on 16th Jun 2007.
- [4] Wikipedia Org. Noise. <http://en.wikipedia.org/wiki/Noise>, Accessed on 11th July 2007.
- [5] Wikipedia Org. Image Noise. http://en.wikipedia.org/wiki/Image_noise, Accessed on 11th July 2007.
- [6] Wikipedia Org. Emergency Vehicle Lighting. http://en.wikipedia.org/wiki/Emergency_vehicle_lighting, Accessed on 3rd August 2007.
- [7] Fundamental segmentation. <http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip-Segmenta.html#Heading118>, Accessed on 29th July 2007.

APPENDICE

Program code

```
Private Sub cmd1load_Click()
```

```
Call List1.Clear  
Call List2.Clear  
Call List3.Clear  
Call List4.Clear
```

```
"COMMAND TO LOAD PICTURE
```

```
cmd1gnazeri.InitDir = "D:\PICTURE YANG DIGUNAKAN UNTUK  
PERBENTANGAN"  
cmd1gnazeri.ShowOpen  
Image1.Load cmd1gnazeri.FileName  
Picture1.Picture = Image1.Picture
```

```
"COMMAND TO SHOW ORIGINAL IMAGE
```

```
lblcomment = "ORIGINAL IMAGE"  
lblcomment.Visible = True
```

```
Display1.Visible = False  
Picture1.Visible = True
```

```
End Sub
```

```
Private Sub cmd2analyzearea_Click()
```

```
"COMMAND TO ANALYZE AREA
```

```
Display1.Visible = True  
Picture1.Visible = False  
lblcomment.Visible = False
```

```
Picture1.Picture = Image2.Picture  
Call SavePicture(Picture1, "D:\PICTURE YANG DIGUNAKAN UNTUK  
PERBENTANGAN/Analyzehaha.BMP")
```

```
'ImageProcessing4.Smooth  
'ImageProcessing4.Rank imp3x3Rect, impMedian, impGrayscale
```

End Sub

Private Sub cmd3binarize_Click()

"COMMAND TO BINARIZE

Const IMAGE_THRESHOLD_VALUE = 120

ImageProcessing1.Binarize impLessOrEqualTo, IMAGE_THRESHOLD_VALUE

Picture1.Picture = Image3.Picture

Call SavePicture(Picture1, "D:\PICTURE YANG DIGUNAKAN UNTUK
PERBENTANGAN/BINARIZE.BMP")

End Sub

Private Sub cmd4filtering_Click()

"REMOVE NOISE

ImageProcessing2.Rank imp3x3Rect, impMedian, impBinary

ImageProcessing2.Smooth

"REMOVE NOISE

'ImageProcessing2.Rank imp3x3Rect, impMedian, impBinary

'ImageProcessing2.Smooth

"REMOVE NOISE

'ImageProcessing2.Rank imp3x3Rect, impMedian, impBinary

'ImageProcessing2.Smooth

'Const MIN_BLOB_RADIUS = 5

'OPEN\CLOSE

'ImageProcessing2.Open MIN_BLOB_RADIUS, impBinary

'ImageProcessing2.Close MIN_BLOB_RADIUS, impBinary

" REMOVE ALL BLOB THAT TOUCHING THE

FRAME

BlobAnalysisFrame.EraseBorderBlobs Image3, Image3, blobBinary, blobZero,
blob8Connected

"SAVE PICTURE

Picture1.Picture = Image3.Picture

Call SavePicture(Picture1, "D:\PICTURE YANG DIGUNAKAN UNTUK
PERBENTANGAN/FILTERED.bmp")

End Sub

Private Sub cmd5calculate_Click()

"COMMAND TO CALCULATE

'BLOB.ForegroundPixelValue = blobZero

'OPERATION BLOB

BLOB.Calculate

BLOB.ApplyFilter (1)

'BLOB.ApplyFilter (2)

BLOB.ApplyFilter (3)

BLOB.ApplyFilter (4)

BLOB.ApplyFilter (5)

'BLOB.ApplyFilter (6)

'BLOB.ApplyFilter (7)

'BLOB.ApplyFilter (8)

L1 = BLOB.Results.Count

List1.AddItem Str(L1)

"If L1 = 1 Then

"List1.AddItem 1

"End If

'OPERATION BLOB1

BLOB1.Calculate

BLOB1.ApplyFilter (1)

BLOB1.ApplyFilter (2)

BLOB1.ApplyFilter (3)

BLOB1.ApplyFilter (4)

BLOB1.ApplyFilter (5)

BLOB1.ApplyFilter (6)

BLOB1.ApplyFilter (7)

R1 = BLOB1.Results.Count

List3.AddItem Str(R1)

```
If R1 > 1 Then  
BLOB2.Calculate
```

```
BLOB2.ApplyFilter (1)  
BLOB2.ApplyFilter (2)  
BLOB2.ApplyFilter (3)  
BLOB2.ApplyFilter (4)  
BLOB2.ApplyFilter (5)  
BLOB2.ApplyFilter (6)  
BLOB2.ApplyFilter (7)
```

```
R2 = BLOB2.Results.Count  
List4.AddItem Str(R2)  
End If
```

```
If L1 = 0 Then  
'OPERATION BLOB3  
BLOB3.Calculate
```

```
BLOB3.ApplyFilter (1)  
BLOB3.ApplyFilter (2)  
BLOB3.ApplyFilter (3)  
BLOB3.ApplyFilter (4)  
BLOB3.ApplyFilter (5)  
BLOB3.ApplyFilter (6)  
BLOB3.ApplyFilter (7)  
BLOB3.ApplyFilter (8)
```

```
L2 = BLOB3.Results.Count  
List2.AddItem Str(L2)  
End If
```

```
End Sub
```

```
Private Sub cmd6result_Click()
```

```
End Sub
```

```
Private Sub cmd9exit_Click()
```

```
End
```

```
End Sub
```

```
Private Sub cmdautoprocess_Click()
```

```
Call List1.Clear
```

```
Call List2.Clear
```

```
Call List3.Clear
```

```
Call List4.Clear
```

```
"COMMAND TO LOAD PICTURE
```

```
cmdlgnazeri.InitDir = "D:\PICTURE YANG DIGUNAKAN UNTUK  
PERBENTANGAN"
```

```
cmdlgnazeri.ShowOpen
```

```
Image1.Load cmdlgnazeri.FileName
```

```
Picture1.Picture = Image1.Picture
```

```
"COMMAND TO SHOW ORIGINAL IMAGE
```

```
lblcomment = "ORIGINAL IMAGE"
```

```
lblcomment.Visible = True
```

```
Display1.Visible = False
```

```
Picture1.Visible = True
```

```
"COMMAND TO ANALYZE AREA
```

```
Display1.Visible = True
```

```
Picture1.Visible = False
```

```
lblcomment.Visible = False
```

```
ImageProcessing4.Smooth
```

```
ImageProcessing4.Rank imp3x3Rect, impMedian, impGrayscale
```

```
Const IMAGE_THRESHOLD_VALUE = 120
```


"COMMAND TO BINARIZE

```
ImageProcessing1.Binarize impLessOrEqualTo, IMAGE_THRESHOLD_VALUE
Picture1.Picture = Image3.Picture
Call SavePicture(Picture1, "D:\PICTURE YANG DIGUNAKAN UNTUK
PERBENTANGAN/BINARIZE.BMP")
```

"COMMAND TO FILTERING

"REMOVE NOISE

```
ImageProcessing2.Rank imp3x3Rect, impMedian, impBinary
ImageProcessing2.Smooth
```

"REMOVE NOISE

```
'ImageProcessing2.Rank imp3x3Rect, impMedian, impBinary
'ImageProcessing2.Smooth
```

"REMOVE NOISE

```
'ImageProcessing2.Rank imp3x3Rect, impMedian, impBinary
'ImageProcessing2.Smooth
```

```
'Const MIN_BLOB_RADIUS = 5
```

```
'OPEN\CLOSE
```

```
'ImageProcessing2.Open MIN_BLOB_RADIUS, impBinary
```

```
'ImageProcessing2.Close MIN_BLOB_RADIUS, impBinary
```

" REMOVE ALL BLOB THAT TOUCHING THE

FRAME

```
BlobAnalysisFrame.EraseBorderBlobs Image3, Image3, blobBinary, blobZero,
blob8Connected
```

"SAVE PICTURE

```
Picture1.Picture = Image3.Picture
```

```
Call SavePicture(Picture1, "D:\PICTURE YANG DIGUNAKAN UNTUK
PERBENTANGAN/FILTERED.bmp")
```

```
'BLOB.ForegroundPixelValue = blobZero
```

```
'OPERATION BLOB  
BLOB.Calculate
```

```
BLOB.ApplyFilter (1)  
'BLOB.ApplyFilter (2)  
BLOB.ApplyFilter (3)  
BLOB.ApplyFilter (4)  
BLOB.ApplyFilter (5)  
'BLOB.ApplyFilter (6)  
'BLOB.ApplyFilter (7)  
'BLOB.ApplyFilter (8)
```

```
L1 = BLOB.Results.Count  
List1.AddItem Str(L1)
```

```
"If L1 = 1 Then  
"List1.AddItem 1  
"End If
```

```
'OPERATION BLOB1  
BLOB1.Calculate
```

```
BLOB1.ApplyFilter (1)  
BLOB1.ApplyFilter (2)  
BLOB1.ApplyFilter (3)  
BLOB1.ApplyFilter (4)  
BLOB1.ApplyFilter (5)  
BLOB1.ApplyFilter (6)  
BLOB1.ApplyFilter (7)  
R1 = BLOB1.Results.Count  
List3.AddItem Str(R1)
```

```
If R1 > 1 Then  
BLOB2.Calculate
```

```
BLOB2.ApplyFilter (1)  
BLOB2.ApplyFilter (2)  
BLOB2.ApplyFilter (3)  
BLOB2.ApplyFilter (4)  
BLOB2.ApplyFilter (5)  
BLOB2.ApplyFilter (6)  
BLOB2.ApplyFilter (7)
```

```

R2 = BLOB2.Results.Count
List4.AddItem Str(R2)
End If

```

```

If L1 = 0 Then
'OPERATION BLOB3
BLOB3.Calculate

```

```

BLOB3.ApplyFilter (1)
BLOB3.ApplyFilter (2)
BLOB3.ApplyFilter (3)
BLOB3.ApplyFilter (4)
BLOB3.ApplyFilter (5)
BLOB3.ApplyFilter (6)
BLOB3.ApplyFilter (7)
BLOB3.ApplyFilter (8)

```

```

L2 = BLOB3.Results.Count
List2.AddItem Str(L2)
End If

```

"COMMAND TO SHOW RESULT

```

RESULT1 = "EMERGENCY LIGHT"
RESULT2 = "NON EMERGENCY LIGHT"

```

```

'1
If L1 = 0 And L2 = 0 And R1 = 0 Then
Text1.Text = RESULT2
End If

```

```

'2
If L1 = 0 And L2 = 0 And R1 = 1 Then
Text1.Text = RESULT1
End If

```

```

'3
If L1 = 0 And L2 = 0 And R1 > 1 And R2 = 0 Then
Text1.Text = RESULT2
End If

```

```

'4
If L1 = 0 And L2 = 0 And R1 > 1 And R2 = 1 Then
Text1.Text = RESULT1

```

End If

'5

If L1 = 0 And L2 = 0 And R1 > 1 And R2 > 1 Then

Text1.Text = RESULT2

End If

'6

If L1 = 1 And R1 = 0 Then

Text1.Text = RESULT1

End If

'7

If L1 = 1 And R1 = 1 Then

Text1.Text = RESULT1

End If

'8

If L1 = 1 And R1 > 1 And R2 = 0 Then

Text1.Text = RESULT1

End If

'9

If L1 = 1 And R1 > 1 And R2 = 1 Then

Text1.Text = RESULT1

End If

'10

If L1 > 1 And R1 = 0 Then

Text1.Text = RESULT2

End If

'11

If L1 > 1 And R1 = 1 Then

Text1.Text = RESULT1

End If

'12

If L1 > 1 And R1 > 1 Then

Text1.Text = RESULT2

End If

'13

If L1 > 1 And R2 > 1 And R2 = 1 Then

Text1.Text = RESULT1

End If

'14

If L1 > 1 And R1 > 1 And R2 > 1 Then

Text1.Text = RESULT2

End If

End Sub

Private Sub Form_Load()

'CONST FOR BLOB

Const EXCLUDE_AREA_LESS_THAN_1225BLOB = 1

Const EXCLUDE_BOXXMIN_LESS_THAN_35BLOB = 2

Const EXCLUDE_BOXXMAX_GREATER_EQUAL_246BLOB = 3

Const EXCLUDE_BOXYMIN_LESS_THAN_18BLOB = 4

Const EXCLUDE_BOXYMAX_GREATER_THAN_115BLOB = 5

Const EXCLUDE_FERETX_IN_RANGE_0_5BLOB = 6

Const EXCLUDE_FERETY_GREATER_THAN_72BLOB = 7

Const BLOBP = 8

'CONST FOR BLOB1

Const EXCLUDE_BOXXMIN_LESS_EQUAL_320BLOB = 1

Const EXCLUDE_BOXXMAX_GREATER_THAN_632BLOB1 = 2

Const EXCLUDE_BOXYMIN_LESS_THAN_18BLOB1 = 3

Const EXCLUDE_BOXYMAX_GREATER_THAN_115BLOB1 = 4

Const EXCLUDE_FERETX_GREATER_THAN_261BLOB1 = 5

Const EXCLUDE_FERETY_GREATER_THAN_122BLOB1 = 6

Const EXCLUDE_AREA_LESS_THAN_1225BLOB1 = 7

'CONST FOR BLOB2

Const BLOB2XMIN = 1

Const BLOB2XMAX = 2

Const BLOB2YMIN = 3

Const BLOB2YMAX = 4

Const BLOB2A = 5

Const BLOB2FX = 6

Const BLOB2FY = 7

```
'CONST FOR BLOB3
```

```
Const BLOB3A = 1
```

```
Const BLOB3FX = 2
```

```
Const BLOB3FY = 3
```

```
Const BLOB3BXMIN = 4
```

```
Const BLOB3BXMAX = 5
```

```
Const BLOB3BYMIN = 6
```

```
Const BLOB3BYMAX = 7
```

```
Const BLOB3P = 8
```

```
Dim L1 As Variant
```

```
Dim L2 As Variant
```

```
Dim R1 As Variant
```

```
Dim R2 As Variant
```

```
Const IMAGE_THRESHOLD_VALUE = 120
```

```
End Sub
```