Lampiran 1: Petunjuk Instalasi *open source computer vision library*

Instalasi OpenCV (*Open source computer vision library*) pada Windows 98/2000/XP
(summary Resmana Lim, 16 Mei 2002)

1. Dahului dengan instalasi Microsoft Visual C++ ver 6.0 (MSVC 6.0)
2. Instal DirecX 8.1 SDK dan taruh/arahkan instalasi pada directory c:\mssdk (ganti directory default c:\dxsdk menjadi c:\mssdk). Jawab Yes ketika ditanya apakah ingin menambahkan path directory pada MSVC. DirecX SDK ini dibutukan untuk akses informasi multimedia (video, sound, streaming dll).
3. Pastikan pada windows path terdapat path "c:/windows/system" dan "c:/windows".
4. Jalankan ipl25.exe untuk instlasi Intel Image Processing Library. Library ini dibutuhkan pada banyak aplikasi OpenCV.
5. Instal OpenCV ver 2.0 dengan menjalankan opencv_core_b2_0.exe (library dasar), lalu lanjutkan dengan menjalankan opencv_apps_b2_0.exe untuk instalasi aplikasi OpenCV.
6. Upgrade OpenCV dengan versi 2.1
    - Extrak opencv_core_b2.1.zip dan taruh pada directory c:\Program File\Intel\
    - Extrak opencv_apps_b2.1.zip dan taruh pada directory c:\Program File\Intel\
    - Extrak opencv_patch_b2.1.1.zip dan taruh pada directory c:\Program File\Intel\
    - Extrak opencv_dll_addon_b2.1.1.zip dan taruh pada directory c:\Program File\Intel\
7. Pastikan apakah setelah diinstall path untuk ipl dan open CV telah terpasang dengan baik bila tidak tambahkan secara manual.
8. Perhatikan path directory pada MSVC (klik Tools>Options>Directories), dan tambahkan include dan library path:

Pada include path:

Dari MS DirectShow SDK:

C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\BASECLASSES
C:\MSSDK\SAMPLES\MULTIMEDIA\COMMON\INCLUDE
C:\MSSDK\INCLUDE

Dari OpenCV list:

C:\PROGRAM FILES\INTEL\OPENCV\CVAUX\INCLUDE
C:\PROGRAM FILES\INTEL\OPENCV\CV\INCLUDE
C:\PROGRAM FILES\INTEL\OPENCV\APPS\COMMON
C:\PROGRAM FILES\INTEL\OPENCV\OTHERLIBS\VLGRFMTS

Dari IPL:

C:\PROGRAM FILES\INTEL\PLSUITE\INCLUDE

Libraries yang perlu disertakan untuk linking:

Dari MS DirectShow SDK:

C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\
BASECLASSES\RELEASE
C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\
BASECLASSES\debug
C:\MSSDK\LIB

Dari OpenCV:

C:\PROGRAM FILES\INTEL\OPENCV\LIB

Dari Ipl:

C:\PROGRAM FILES\INTEL\PLSUITE\LIB\MSVC

9. Copykan directory \Intel-OpenCV2.1\aplikasi-tambahan\CvlGrFrmts (dari CDRom) menuju directory C:\Program Files\Intel\opencv\otherlibs\

10. Build library tambahan pada:
    - C:\Program Files\Intel\opencv\otherlibs\highgui
    - C:\Program Files\Intel\opencv\otherlibs\CvlGrFrmts

    Lantas tambahkan pada path MSVC untuk include dan library dengan isian directory diatas.

11. Proses instalasi selesai, dan selanjutnya perhatikan struktur directory OpenCV pada c:\Program File\Intel\OpenCV

|

+ **_DSW** // Workspace file for Microsoft Visual Studio and a few Perl utilities for statistics

|

+ **CV** // The library itself

| + Include // External library interface

| + _Include // Internal library interface

| + Make // Project file

| + Src // Source files

|

+ **CVAux** // Additional (experimental) stuff.

|

+ **Docs** // documentation

| + HTML // overview documentatation in HTML format

|

+ **Bin** // all the pre-built binaries

|

+ **Lib** // pre-built import and static libraries

|

+ **Apps** // Demo Applications

| |

| + CamShiftDemo // Application - Wrapper for CamShift Tracker Filter

| + VMDemo // View Morphing demo

| + LkDemo // Lucas-Kanade Pyramid-based Point Tracker

| + HMMDemo // Hidden Markov Models Face Recognition Demo

| + StereoGR // Stereo-based Gesture Recognition App for PointGrey Stereo Cameras

| + Hawk // Scripting Environment

| + Common // CImage and CCamera classes

|

+ **Filters** // Direct Show filters

| + CalibFilter // Camera Calibration filter

| + CamShift // CamShift tracker

| + Condens // ConDensation based tracker

| + Kalman // Kalman filter based tracker

| + ProxyTrans // Proxy DirectShow filter

|

+ **OtherLibs**

| + _Mkl // Math Kernel Library - used for tests on matriks functions

| + _IPL // Image Processing Library - base library for the OpenCV

| + HighGUI // Simple GUI library with platform-independed interface

| + VlGrFmts // Library for reading/writing raster images

| + GestRec // Experimental gesture recognition module

| + PtGrey // Interface Module for PointGrey Stereo Camera

|

+ **Tests**// sources for algorithmic tests

Untuk aplikasi-aplikasi menggunakan video/webcam, membutuhkan library DirectShow. Build baseclasses dari directshow, dengan open project pada MSVC yang ada pada:
C:\mssdk\samples\Multimedia\DirectShow\BaseClasses\baseclasses.dsw

➔ lalu build secara batch baik untuk versi release maupun debug

Bermainlah dengan contoh-contoh aplikasi bawaan OpenCV, dan buat anda familiar dengan struktur OpenCV. Baca Manual pdf yang telah disediakan, baik untuk OpenCV Reference Manual maupun IPL Reference Manual.

Lampiran 2: Listing Listing Program Metode ROI untuk pergerakan dan *event-event* khusus

```
 void CCamCapDlg::CallBackCam1(IplImage *frame)
{
          IplImage*dest_maze_hsv,*dest_maze_h,*dest_maze_s, *dest_maze_v;

        CvSize size_mobil;
        CvRect size_roi_sensorl;
        CvSize size_image_rois;
        CvPoint pt1,pt2;

        size_mobil.width  = frame->width;
        size_mobil.height = frame->height;
   // 2. Create a temporary grey image the same size, flip it and set the origin to
        //          Bottom left (DIB's are upside down to normal images!)

        dest_maze_hsv = cvCreateImage(size_mobil, 8, 3);
        iplMirror(dest_maze_hsv, dest_maze_hsv, 0);

        dest_maze_hsv->origin = IPL_ORIGIN_BL;


// frame di konvert menjadi HSV

   iplRGB2HSV(frame, dest_maze_hsv);

        dest_maze_h=cvCreateImage(size_mobil, 8, 1);
   iplMirror(dest_maze_h, dest_maze_h, 0);
        dest_maze_h->origin = IPL_ORIGIN_BL;

        dest_maze_s=cvCloneImage(dest_maze_h);
        dest_maze_v=cvCloneImage(dest_maze_h);

//cv 14-80
        cvCvtPixToPlane(dest_maze_hsv,          dest_maze_h,          dest_maze_s,
dest_maze_v, NULL);
        if (m_track==0)
        {
                pt1.x = 60;//double klik box
                pt1.y = 40;
                pt2.x = 100;
                pt2.y = 80;
                cvRectangle(frame,pt1,pt2,CV_RGB(255,0,0),2);
        }
        else if (m_track==1)
        {
```

```
                IplImage        *src_warna,     *src_warna_hsv,     *src_warna_h,
*src_warna_s, *src_warna_v;
                int histh_dim[1] = {255};
                int hists_dim[1] = {255};
                int histv_dim[1] = {255};
                CvSize   size;

                                size_roi_sensorl = cvRect(60,40,40,40);

                                size_image_rois = cvSize(40,40);

                                src_warna                                       =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,3);
                                iplMirror(src_warna, src_warna, 0);

                                src_warna->origin = IPL_ORIGIN_BL;

                                cvSetImageROI(dest_maze_hsv, size_roi_sensorl);
                                cvCopyImage(dest_maze_hsv,src_warna);

                size.width  = src_warna->width;
                size.height = src_warna->height;

        //membuat image baru (cv 14-8)
                src_warna_hsv = cvCloneImage(src_warna);

        //merubah RGB ke HSV (ipl 9-12)
                src_warna_h=cvCreateImage(size, IPL_DEPTH_8U, 1);
                src_warna_s=cvCreateImage(size, IPL_DEPTH_8U, 1);
                src_warna_v=cvCreateImage(size, IPL_DEPTH_8U, 1);

        //pisahkan masing-masing channel (cv 14-80)
                cvCvtPixToPlane(src_warna_hsv,     src_warna_h,     src_warna_s,
src_warna_v, NULL);
                float thresh_h[1][2] = { {0, 255} };
                float* pthresh_h[1] = { thresh_h[0] };
        //cv 10-41
                hist_h = cvCreateHist ( 1, histh_dim, CV_HIST_ARRAY, 0, 1);
        //cv 10-50
                cvSetHistBinRanges( hist_h, pthresh_h, 1);
        //cv 10-50
                cvCalcHist(&src_warna_h, hist_h, 0, 0);

        //hitung nilai histogram dari sampel warna src_warna_s
                float thresh_s[1][2]= { {0, 255} };
                float* pthresh_s[1] = { thresh_s[0] };
                hist_s = cvCreateHist ( 1, hists_dim, CV_HIST_ARRAY, 0, 1);
```

```
        //cv 10-50
                cvSetHistBinRanges( hist_s, pthresh_s, 1);
        //cv 10-50
                cvCalcHist(&src_warna_s, hist_s, 0, 0);

        //hitung nilai histogram dari sampel warna src_warna_v
                float thresh_v[1][2] = { {0, 255} };
                float* pthresh_v[1] = { thresh_v[0] };
                hist_v = cvCreateHist ( 1, histv_dim, CV_HIST_ARRAY, 0, 1);
        //cv 10-50
                cvSetHistBinRanges( hist_v, pthresh_v, 1);
        //cv 10-50
                cvCalcHist(&src_warna_v, hist_v, 0, 0);

        cvReleaseImage(&src_warna_v);
        cvReleaseImage(&src_warna_s);
                cvReleaseImage(&src_warna_h);
                cvReleaseImage(&src_warna_hsv);
                cvReleaseImage(&src_warna);
                m_track=2;
        }

        else
        {
//calback
        IplImage  *img_calback_h,*img_calback_and_sv, *semen,*display;

//calback
        img_calback_h = cvCloneImage(dest_maze_h);

        //cv 10-51
        cvCalcBackProject(&dest_maze_h, img_calback_h, hist_h);


// memperbaiki hasil tracking
// masukan konstrain s dan v
        cvThreshold(dest_maze_s,        dest_maze_s,        90,        255,
CV_THRESH_BINARY);
        cvThreshold(dest_maze_v,        dest_maze_v,        90,        255,
CV_THRESH_BINARY);

//SV di and kan ipl 5-15
        img_calback_and_sv = cvCloneImage(dest_maze_h);
        iplAnd(dest_maze_s, dest_maze_v, img_calback_and_sv);

//
        IplImage *temp, *temp1;
        temp=cvCloneImage(dest_maze_h);
```

```
iplAnd(img_calback_h, img_calback_and_sv, temp); //

temp1=cvCloneImage(temp);
iplMedianFilter(temp1, temp, 3, 3, 1, 1); //lowpass filter
cvThreshold(temp,temp,10,255,CV_THRESH_BINARY);
iplMirror(temp,temp,1);
int rx,ry;
CvPoint pt1,pt2;
display=cvCloneImage(temp);
for (rx=32;rx<128;rx=rx+32)
{
        for (ry=24;ry<96;ry=ry+24)
        {

                        pt1.x = rx;
                        pt1.y = ry;
                        pt2.x = rx+32;
                        pt2.y = ry+24;

cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
        }
}
pt1.x = 0;//klik kiri box
pt1.y = 0;
pt2.x = 32;
pt2.y = 120;
cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
pt1.x = 32;//double klik box
pt1.y = 96;
pt2.x = 128;
pt2.y = 120;
cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
pt1.x = 128;//klik kanan box
pt1.y = 0;
pt2.x = 160;
pt2.y = 120;
cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
pt1.x = 32;//drag drop box
pt1.y = 0;
pt2.x = 128;
pt2.y = 24;
if (dr)
cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),8);
else
cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);

cvvShowImage("Probability Image",display);
cvReleaseImage(&display);
```

```
//cek
int jx,jy,cx,cy,xs,ys,valws,valw;
        CvRect size_roi_sensorl;
        CvSize size_image_rois;
        valws = cvCountNonZero (temp);
        if (valws>=50)
        {
                xs=1;
                ys=120;
                jx=0;
                cy=0;
                for (cx=0;cx<(160-xs);cx=cx+xs)
        //linex
                {
                        size_roi_sensorl = cvRect(cx,cy,xs,ys);

                        size_image_rois = cvSize(xs,ys);

                        semen                                                   =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,1);
                        iplMirror(semen, semen, 0);

                        semen->origin = IPL_ORIGIN_BL;

                        cvSetImageROI(temp, size_roi_sensorl);
                        cvCopyImage(temp,semen);

                        valw = cvCountNonZero (semen);
                        cvReleaseImage(&semen);
                        jx=jx+cx*valw;
                }
                xs=160;
                ys=1;
                jy=0;
                cx=0;
                for (cy=0;cy<(120-ys);cy=cy+ys)
        //liney
                {
                        size_roi_sensorl = cvRect(cx,cy,xs,ys);

                        size_image_rois = cvSize(xs,ys);

                        semen                                                   =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,1);
                        iplMirror(semen, semen, 0);

                        semen->origin = IPL_ORIGIN_BL;
```

```
                cvSetImageROI(temp, size_roi_sensorl);
                cvCopyImage(temp,semen);

                valw = cvCountNonZero (semen);
                cvReleaseImage(&semen);
                jy=jy+cy*valw;
        }
        cx=jx/valws;
        cy=jy/valws;

//hitung titik berat dari rect mouse

        if (cx<32)//klik kiri
        {
                if (ste!=1)
                {
                        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
                        mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
                        dr=false;
                }
                ste=1;
        }
        else if (cx>=128)//klik kanan
        {
                if (ste!=2)
                {
                        mouse_event(MOUSEEVENTF_RIGHTDOWN,sx,sy,0,0);
                        mouse_event(MOUSEEVENTF_RIGHTUP,sx,sy,0,0);
                }
                ste=2;
        }
        else
        {
                if (cy<24)//drag drop
                {
                        if (ste!=3 && ste!=4)
                        {
                                if (!dr)
                                {
                                        dr=true;

        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
                                        ste=3;
                                }
                                else
                                {
                                        dr=false;
```

```
mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
                                        ste=4;
                                }
                        }
                }
                else if (cy>=96)
                {
                        if (ste!=5)
                        {

mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);

mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);

mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);

mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
                                dr=false;
                        }
                        ste=5;
                }
                else if (cy<48)//bawah
                {
                        if (cx<64)//kiri
                        {
                                sx=sx-ac;
                                sy=sy+ac;
                                if (ste==6)
                                        ac++;
                                else
                                        ac=1;
                                ste=6;
                        }
                        else if (cx<96)//tengah
                        {
                                //sx=sx+ac;
                                sy=sy+ac;
                                if (ste==7)
                                        ac++;
                                else
                                        ac=1;
                                ste=7;
                        }
                        else//kanan
                        {
                                sx=sx+ac;
                                sy=sy+ac;
```

```
                    if (ste==8)
                        ac++;
                    else
                        ac=1;
                    ste=8;
                }
            }
            else if (cy<72)//tengah
            {
                if (cx<64)//kiri
                {
                    sx=sx-ac;
                    //sy=sy+ac;
                    if (ste==9)
                        ac++;
                    else
                        ac=1;
                    ste=9;
                }
                else if (cx<96)//tengah
                {
                    //sx=sx+ac;
                    //sy=sy+ac;
                    ste=0;
                }

                else//kanan
                {
                    sx=sx+ac;
                    //sy=sy+ac;
                    if (ste==10)
                        ac++;
                    else
                        ac=1;
                    ste=10;
                }
            }
            else//atas
            {
                if (cx<64)//kiri
                {
                    sx=sx-ac;
                    sy=sy-ac;
                    if (ste==11)
                        ac++;
                    else
                        ac=1;
                    ste=11;
```

```
                        }
                        else if (cx<96)//tengah
                        {
                                //sx=sx+ac;
                                sy=sy-ac;
                                if (ste==12)
                                        ac++;
                                else
                                        ac=1;
                                ste=12;
                        }

                        else//kanan
                        {
                                sx=sx+ac;
                                sy=sy-ac;
                                if (ste==13)
                                        ac++;
                                else
                                        ac=1;
                                ste=13;
                        }
                }
        }
        if (ste>=6)
        SetCursorPos(sx,sy);
        else if (ste==3)
        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
}
                cvReleaseImage(&temp);
                cvReleaseImage(&temp1);
                cvReleaseImage(&img_calback_h);
                cvReleaseImage(&img_calback_and_sv);
                cvReleaseImage(&display);

}
//cv 14-9
        cvReleaseImage(&dest_maze_v);
        cvReleaseImage(&dest_maze_s);
        cvReleaseImage(&dest_maze_h);
        cvReleaseImage(&dest_maze_hsv);
}
```

Lampiran 3: Listing Program Metode tracking untuk pergerakan ROI untuk *event-event* khusus

```
void CCamCapDlg::CallBackCam1(IplImage *frame)
{

        IplImage *dest_maze_hsv, *dest_maze_h, *dest_maze_s,
*dest_maze_v;

        CvSize size_mobil;
        CvRect size_roi_sensorl;
        CvSize size_image_rois;
        CvPoint pt1,pt2;

        size_mobil.width  = frame->width;
        size_mobil.height = frame->height;
   // 2. Create a temporary grey image the same size, flip it and set the origin to
        //          Bottom left (DIB's are upside down to normal images!)

        dest_maze_hsv = cvCreateImage(size_mobil, 8, 3);
        iplMirror(dest_maze_hsv, dest_maze_hsv, 0);

        dest_maze_hsv->origin = IPL_ORIGIN_BL;

// frame di konvert menjadi HSV

   iplRGB2HSV(frame, dest_maze_hsv);

        dest_maze_h=cvCreateImage(size_mobil, 8, 1);
   iplMirror(dest_maze_h, dest_maze_h, 0);
        dest_maze_h->origin = IPL_ORIGIN_BL;

        dest_maze_s=cvCloneImage(dest_maze_h);
        dest_maze_v=cvCloneImage(dest_maze_h);

//cv 14-80
        cvCvtPixToPlane(dest_maze_hsv,          dest_maze_h,          dest_maze_s,
dest_maze_v, NULL);
        if (m_track==0)
        {
                pt1.x = 60;//double klik box
                pt1.y = 40;
                pt2.x = 100;
                pt2.y = 80;
                cvRectangle(frame,pt1,pt2,CV_RGB(255,0,0),2);
        }
        else if (m_track==1)
```

```
{

        IplImage *src_warna, *src_warna_hsv, *src_warna_h,
*src_warna_s, *src_warna_v;
        int histh_dim[1] = {255};
        int hists_dim[1] = {255};
        int histv_dim[1] = {255};
        CvSize  size;

                size_roi_sensorl = cvRect(60,40,40,40);

                size_image_rois = cvSize(40,40);

                src_warna =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,3);
                iplMirror(src_warna, src_warna, 0);

                src_warna->origin = IPL_ORIGIN_BL;

                cvSetImageROI(dest_maze_hsv, size_roi_sensorl);
                cvCopyImage(dest_maze_hsv,src_warna);

        size.width  = src_warna->width;
        size.height = src_warna->height;

    //membuat image baru (cv 14-8)
        src_warna_hsv = cvCloneImage(src_warna);

    //merubah RGB ke HSV (ipl 9-12)
        src_warna_h=cvCreateImage(size, IPL_DEPTH_8U, 1);
        src_warna_s=cvCreateImage(size, IPL_DEPTH_8U, 1);
        src_warna_v=cvCreateImage(size, IPL_DEPTH_8U, 1);

    //pisahkan masing-masing channel (cv 14-80)
        cvCvtPixToPlane(src_warna_hsv, src_warna_h, src_warna_s,
src_warna_v, NULL);
        float thresh_h[1][2] = { {0, 255} };
        float* pthresh_h[1] = { thresh_h[0] };
    //cv 10-41
        hist_h = cvCreateHist ( 1, histh_dim, CV_HIST_ARRAY, 0, 1);
    //cv 10-50
        cvSetHistBinRanges( hist_h, pthresh_h, 1);
    //cv 10-50
        cvCalcHist(&src_warna_h, hist_h, 0, 0);

    //hitung nilai histogram dari sampel warna src_warna_s
        float thresh_s[1][2]= { {0, 255} };
        float* pthresh_s[1] = { thresh_s[0] };
```

```
            hist_s = cvCreateHist ( 1, hists_dim, CV_HIST_ARRAY, 0, 1);
        //cv 10-50
            cvSetHistBinRanges( hist_s, pthresh_s, 1);
        //cv 10-50
            cvCalcHist(&src_warna_s, hist_s, 0, 0);

        //hitung nilai histogram dari sampel warna src_warna_v
            float thresh_v[1][2] = { {0, 255} };
            float* pthresh_v[1] = { thresh_v[0] };
            hist_v = cvCreateHist ( 1, histv_dim, CV_HIST_ARRAY, 0, 1);
        //cv 10-50
            cvSetHistBinRanges( hist_v, pthresh_v, 1);
        //cv 10-50
            cvCalcHist(&src_warna_v, hist_v, 0, 0);

        cvReleaseImage(&src_warna_v);
        cvReleaseImage(&src_warna_s);
            cvReleaseImage(&src_warna_h);
            cvReleaseImage(&src_warna_hsv);
            cvReleaseImage(&src_warna);
            m_track=2;
        }

        else
        {
//calback
        IplImage  *img_calback_h,*img_calback_and_sv, *semen,*display;


//calback
        img_calback_h = cvCloneImage(dest_maze_h);

        //cv 10-51
        cvCalcBackProject(&dest_maze_h, img_calback_h, hist_h);

// memperbaiki hasil tracking
// masukan konstrain s dan v
        cvThreshold(dest_maze_s,        dest_maze_s,        90,        255,
CV_THRESH_BINARY);
        cvThreshold(dest_maze_v,        dest_maze_v,        90,        255,
CV_THRESH_BINARY);

//SV di and kan ipl 5-15
        img_calback_and_sv = cvCloneImage(dest_maze_h);
        iplAnd(dest_maze_s, dest_maze_v, img_calback_and_sv);

//
        IplImage *temp, *temp1;
```

```
        temp=cvCloneImage(dest_maze_h);;
        cvvShowImage("Hue",img_calback_h);
        iplAnd(img_calback_h, img_calback_and_sv, temp); //

        temp1=cvCloneImage(temp);
        iplMedianFilter(temp1, temp, 3, 3, 1, 1); //lowpass filter
        cvThreshold(temp,temp,10,255,CV_THRESH_BINARY);
        iplMirror(temp,temp,1);

        CvPoint pt1,pt2;
        display=cvCloneImage(temp);
        pt1.x = 0;//klik kiri box
        pt1.y = 0;
        pt2.x = 30;
        pt2.y = 120;
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
        pt1.x = 30;//double klik box
        pt1.y = 100;
        pt2.x = 130;
        pt2.y = 120;
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
        pt1.x = 130;//klik kanan box
        pt1.y = 0;
        pt2.x = 160;
        pt2.y = 120;
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
        pt1.x = 30;//drag drop box
        pt1.y = 0;
        pt2.x = 130;
        pt2.y = 20;
        if (dr)
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),8);
        else
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);

        cvvShowImage("Probability Image",display);
        cvReleaseImage(&display);
//cek

int jx,jy,cx,cy,xs,ys,valws,valw;
        CvRect size_roi_sensorl;
        CvSize size_image_rois;
        valws = cvCountNonZero (temp);
        if (valws>=10)
        {
                xs=1;
                ys=120;
                jx=0;
```

```
                cy=0;
                for (cx=0;cx<(160-xs);cx=cx+xs)
        //linex
                {
                        size_roi_sensorl = cvRect(cx,cy,xs,ys);

                        size_image_rois = cvSize(xs,ys);

                        semen                                                   =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,1);
                        iplMirror(semen, semen, 0);

                        semen->origin = IPL_ORIGIN_BL;

                        cvSetImageROI(temp, size_roi_sensorl);
                        cvCopyImage(temp,semen);

                        valw = cvCountNonZero (semen);
                        cvReleaseImage(&semen);
                        jx=jx+cx*valw;
                }
                xs=160;
                ys=1;
                jy=0;
                cx=0;
                for (cy=0;cy<(120-ys);cy=cy+ys)
        //liney
                {
                        size_roi_sensorl = cvRect(cx,cy,xs,ys);

                        size_image_rois = cvSize(xs,ys);

                        semen                                                   =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,1);
                        iplMirror(semen, semen, 0);

                        semen->origin = IPL_ORIGIN_BL;

                        cvSetImageROI(temp, size_roi_sensorl);
                        cvCopyImage(temp,semen);

                        valw = cvCountNonZero (semen);
                        cvReleaseImage(&semen);
                        jy=jy+cy*valw;
                }
                cx=jx/valws;
                cy=jy/valws;
```

```
//hitung titik berat dari rect mouse
        if (cx<30)//klik kiri
        {
                if (ste!=1)
                {
                        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
                        mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
                        dr=false;
                }
                ste=1;
        }
        else if (cx>=130)//klik kanan
        {
                if (ste!=2)
                {
                        mouse_event(MOUSEEVENTF_RIGHTDOWN,sx,sy,0,0);
                        mouse_event(MOUSEEVENTF_RIGHTUP,sx,sy,0,0);
                }
                ste=2;
        }
        else
        {
                if (cy<20)//drag drop
                {
                        if (ste!=3 && ste!=4)
                        {
                                if (!dr)
                                {
                                        dr=true;

        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
                                        ste=3;
                                }
                                else
                                {
                                        dr=false;

        mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
                                        ste=4;
                                }
                        }
                }
                else if (cy>=100)
                {
                        if (ste!=5)
                        {

        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
```

```
		mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);

		mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);

		mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
					dr=false;
				}
				ste=5;
			}
			else
			{
				xl=cx-xl;//xl negatif ke kiri
				if (labs(xl)>4)
					sx=640/160*cx;
				xl=cx;
				yl=yl-cy;//yl negatif ke atas
				if (labs(yl)>3)
					sy=480/120*cy;
				yl=cy;
				ste=6;
			}
		}
		if (ste==6)
		SetCursorPos(sx,sy);
		else if (ste==3)
		mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
}
			cvReleaseImage(&temp);
			cvReleaseImage(&temp1);
			cvReleaseImage(&img_calback_h);
			cvReleaseImage(&img_calback_and_sv);
			cvReleaseImage(&display);

}
//cv 14-9
		cvReleaseImage(&dest_maze_v);
		cvReleaseImage(&dest_maze_s);
		cvReleaseImage(&dest_maze_h);
		cvReleaseImage(&dest_maze_hsv);
}
```

Lampiran 4: Listing Program Metode tracking berlanjut untuk pergerakan dan ROI untuk *event-event* khusus

```
void CCamCapDlg::CallBackCam1(IplImage *frame)
{

        IplImage        *dest_maze_hsv,         *dest_maze_h,         *dest_maze_s,
*dest_maze_v;

        CvSize size_mobil;
        CvRect size_roi_sensorl;
        CvSize size_image_rois;
        CvPoint pt1,pt2;

        size_mobil.width  = frame->width;
        size_mobil.height = frame->height;
   // 2. Create a temporary grey image the same size, flip it and set the origin to
        //          Bottom left (DIB's are upside down to normal images!)

        dest_maze_hsv = cvCreateImage(size_mobil, 8, 3);
        iplMirror(dest_maze_hsv, dest_maze_hsv, 0);

        dest_maze_hsv->origin = IPL_ORIGIN_BL;


// frame di konvert menjadi HSV

   iplRGB2HSV(frame, dest_maze_hsv);

        dest_maze_h=cvCreateImage(size_mobil, 8, 1);
   iplMirror(dest_maze_h, dest_maze_h, 0);
        dest_maze_h->origin = IPL_ORIGIN_BL;

        dest_maze_s=cvCloneImage(dest_maze_h);
        dest_maze_v=cvCloneImage(dest_maze_h);

//cv 14-80
        cvCvtPixToPlane(dest_maze_hsv,          dest_maze_h,          dest_maze_s,
dest_maze_v, NULL);
        if (m_track==0)
        {
                pt1.x = 60;//double klik box
                pt1.y = 40;
                pt2.x = 100;
                pt2.y = 80;
                cvRectangle(frame,pt1,pt2,CV_RGB(255,0,0),2);
        }
```

```
        else if (m_track==1)
        {

                IplImage *src_warna, *src_warna_hsv, *src_warna_h,
*src_warna_s, *src_warna_v;
                int histh_dim[1] = {255};
                int hists_dim[1] = {255};
                int histv_dim[1] = {255};
                CvSize   size;

                                size_roi_sensorl = cvRect(60,40,40,40);

                                size_image_rois = cvSize(40,40);

                                src_warna                                       =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,3);
                                iplMirror(src_warna, src_warna, 0);

                                src_warna->origin = IPL_ORIGIN_BL;

                                cvSetImageROI(dest_maze_hsv, size_roi_sensorl);
                                cvCopyImage(dest_maze_hsv,src_warna);

                size.width  = src_warna->width;
                size.height = src_warna->height;

        //membuat image baru (cv 14-8)
                src_warna_hsv = cvCloneImage(src_warna);

        //merubah RGB ke HSV (ipl 9-12)
                src_warna_h=cvCreateImage(size, IPL_DEPTH_8U, 1);
                src_warna_s=cvCreateImage(size, IPL_DEPTH_8U, 1);
                src_warna_v=cvCreateImage(size, IPL_DEPTH_8U, 1);

        //pisahkan masing-masing channel (cv 14-80)
                cvCvtPixToPlane(src_warna_hsv,  src_warna_h,  src_warna_s,
src_warna_v, NULL);
                float thresh_h[1][2] = { {0, 255} };
                float* pthresh_h[1] = { thresh_h[0] };
        //cv 10-41
                hist_h = cvCreateHist ( 1, histh_dim, CV_HIST_ARRAY, 0, 1);
        //cv 10-50
                cvSetHistBinRanges( hist_h, pthresh_h, 1);
        //cv 10-50
                cvCalcHist(&src_warna_h, hist_h, 0, 0);

        //hitung nilai histogram dari sampel warna src_warna_s
                float thresh_s[1][2]= { {0, 255} };
```

```
                float* pthresh_s[1] = { thresh_s[0] };
                hist_s = cvCreateHist ( 1, hists_dim, CV_HIST_ARRAY, 0, 1);
        //cv 10-50
                cvSetHistBinRanges( hist_s, pthresh_s, 1);
        //cv 10-50
                cvCalcHist(&src_warna_s, hist_s, 0, 0);

        //hitung nilai histogram dari sampel warna src_warna_v
                float thresh_v[1][2] = { {0, 255} };
                float* pthresh_v[1] = { thresh_v[0] };
                hist_v = cvCreateHist ( 1, histv_dim, CV_HIST_ARRAY, 0, 1);
        //cv 10-50
                cvSetHistBinRanges( hist_v, pthresh_v, 1);
        //cv 10-50
                cvCalcHist(&src_warna_v, hist_v, 0, 0);

        cvReleaseImage(&src_warna_v);
        cvReleaseImage(&src_warna_s);
                cvReleaseImage(&src_warna_h);
                cvReleaseImage(&src_warna_hsv);
                cvReleaseImage(&src_warna);
                m_track=2;
        }

        else
        {
//calback
        IplImage  *img_calback_h,*img_calback_and_sv, *semen,*display;


//calback
        img_calback_h = cvCloneImage(dest_maze_h);

        //cv 10-51
        cvCalcBackProject(&dest_maze_h, img_calback_h, hist_h);


// memperbaiki hasil tracking
// masukan konstrain s dan v
        cvThreshold(dest_maze_s,        dest_maze_s,        90,        255,
CV_THRESH_BINARY);
        cvThreshold(dest_maze_v,        dest_maze_v,        90,        255,
CV_THRESH_BINARY);

//SV di and kan ipl 5-15
        img_calback_and_sv = cvCloneImage(dest_maze_h);
        iplAnd(dest_maze_s, dest_maze_v, img_calback_and_sv);
```

```
//
        IplImage *temp, *temp1;
        temp=cvCloneImage(dest_maze_h);;
        cvvShowImage("Hue",img_calback_h);
        iplAnd(img_calback_h, img_calback_and_sv, temp); //

        temp1=cvCloneImage(temp);
        iplMedianFilter(temp1, temp, 3, 3, 1, 1); //lowpass filter
        cvThreshold(temp,temp,10,255,CV_THRESH_BINARY);
        iplMirror(temp,temp,1);

        CvPoint pt1,pt2;
        display=cvCloneImage(temp);
        pt1.x = 0;//klik kiri box
        pt1.y = 0;
        pt2.x = 30;
        pt2.y = 120;
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
        pt1.x = 30;//double klik box
        pt1.y = 100;
        pt2.x = 130;
        pt2.y = 120;
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
        pt1.x = 130;//klik kanan box
        pt1.y = 0;
        pt2.x = 160;
        pt2.y = 120;
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);
        pt1.x = 30;//drag drop box
        pt1.y = 0;
        pt2.x = 130;
        pt2.y = 20;
        if (dr)
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),8);
        else
        cvRectangle(display,pt1,pt2,CV_RGB(255,255,255),2);

        cvvShowImage("Probability Image",display);
        cvReleaseImage(&display);
//cek

int jx,jy,cx,cy,xs,ys,valws,valw;
        CvRect size_roi_sensorl;
        CvSize size_image_rois;
        valws = cvCountNonZero (temp);
        if (valws>=10)
        {
                xs=1;
```

```
                ys=120;
                jx=0;
                cy=0;
                for (cx=0;cx<(160-xs);cx=cx+xs)
        //linex
                {
                        size_roi_sensorl = cvRect(cx,cy,xs,ys);

                        size_image_rois = cvSize(xs,ys);

                        semen                                                   =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,1);
                        iplMirror(semen, semen, 0);

                        semen->origin = IPL_ORIGIN_BL;

                        cvSetImageROI(temp, size_roi_sensorl);
                        cvCopyImage(temp,semen);

                        valw = cvCountNonZero (semen);
                        cvReleaseImage(&semen);
                        jx=jx+cx*valw;
                }
                xs=160;
                ys=1;
                jy=0;
                cx=0;
                for (cy=0;cy<(120-ys);cy=cy+ys)
        //liney
                {
                        size_roi_sensorl = cvRect(cx,cy,xs,ys);

                        size_image_rois = cvSize(xs,ys);

                        semen                                                   =
cvCreateImage(size_image_rois,IPL_DEPTH_8U,1);
                        iplMirror(semen, semen, 0);

                        semen->origin = IPL_ORIGIN_BL;

                        cvSetImageROI(temp, size_roi_sensorl);
                        cvCopyImage(temp,semen);

                        valw = cvCountNonZero (semen);
                        cvReleaseImage(&semen);
                        jy=jy+cy*valw;
                }
                cx=jx/valws;
```

```
                cy=jy/valws;

//hitung titik berat dari rect mouse
        if (cx<30)//klik kiri
        {
                if (ste!=1)
                {
                        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
                        mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
                        dr=false;
                }
                ste=1;
        }
        else if (cx>=130)//klik kanan
        {
                if (ste!=2)
                {
                        mouse_event(MOUSEEVENTF_RIGHTDOWN,sx,sy,0,0);
                        mouse_event(MOUSEEVENTF_RIGHTUP,sx,sy,0,0);
                }
                ste=2;
        }
        else
        {
                if (cy<20)//drag drop
                {
                        if (ste!=3 && ste!=4)
                        {
                                if (!dr)
                                {
                                        dr=true;

        mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
                                        ste=3;
                                }
                                else
                                {
                                        dr=false;

        mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
                                        ste=4;
                                }
                        }
                }
                else if (cy>=100)
                {
                        if (ste!=5)
                        {
```

```
		mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);

		mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);

		mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);

		mouse_event(MOUSEEVENTF_LEFTUP,sx,sy,0,0);
					dr=false;
				}
				ste=5;
			}
			else
			{
				xl=cx-xl;//xl negatif ke kiri
				if (labs(xl)>1 && labs(xl)<15)
					sx=sx+xl*3;
				xl=cx;
				yl=yl-cy;//yl negatif ke atas
				if (labs(yl)>1 && labs(yl)<10)
					sy=sy+yl*3;
				yl=cy;
				ste=6;
			}
		}
		if (ste==6)
		SetCursorPos(sx,sy);
		else if (ste==3)
		mouse_event(MOUSEEVENTF_LEFTDOWN,sx,sy,0,0);
}
			cvReleaseImage(&temp);
			cvReleaseImage(&temp1);
			cvReleaseImage(&img_calback_h);
			cvReleaseImage(&img_calback_and_sv);
			cvReleaseImage(&display);

}
//cv 14-9
		cvReleaseImage(&dest_maze_v);
		cvReleaseImage(&dest_maze_s);
		cvReleaseImage(&dest_maze_h);
		cvReleaseImage(&dest_maze_hsv);   }
```

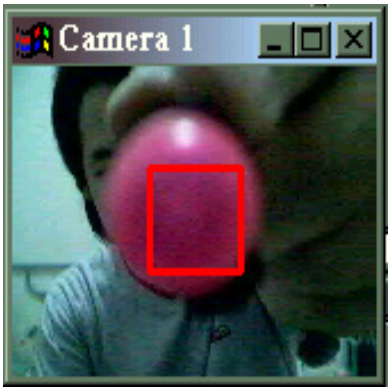
MANUAL BOOK

Pada saat program dijalankan maka tampilan program akan tampak seperti pada gambar ini:

Setelah itu yang harus dilakukan adalah :

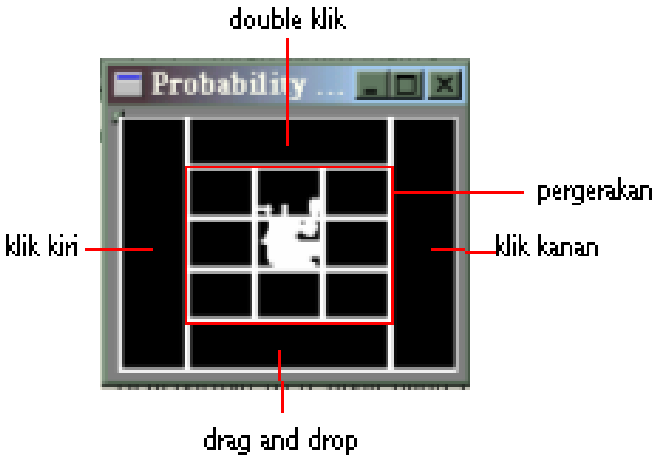
1. Tekan tombol pilihan untuk memilih sekaligus mengaktifkan kamera. Tampilan pada kamera akan nampak seperti gambar berikut:

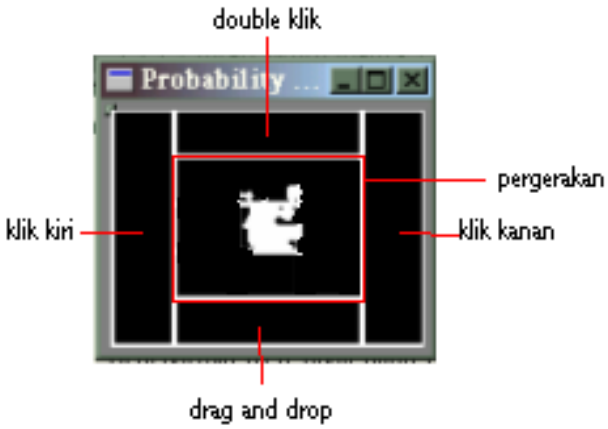
2. Pengecekan dan penyesuaian resolusi pada kamera dengan cara menekan tombol change resolution. Resolusi pada kamera harus berada pada 160x120 pixel.
3. Pengecekan dan penyesuaian apakah brightness dari kamera sehingga tidak terlalu gelap ataupun terang.
4. Setelah selesai dekatkan obyek yang warnanya akan ditracking ke kamera sampai warna obyek menutup kotak merah seperti pada gambar berikut:



5. Tombol init histogram ditekan untuk mengambil sampel warna dari obyek. Letaknya ditunjukkan gambar dibawah ini

6. Setelah itu akan ada 3 tampilan yang berbeda-beda untuk tiap metode, umtuk metode roi akan muncul tampilan seperti pada gambar berikut:





7. untuk metode tracking dan tracking berlanjut sama tampilannya yaitu seperti pada gambar diatas.
8. Setelah tampilan diatas muncul maka untuk metode roi kursor mouse bergerak mengikuti daerah dimana obyek berada, misalkan bila obyek berada di daerah klik kiri maka klik kiri yang terjadi, bila obyek berada di daerah pergerakan mouse yaitu bergerak keatas maka kursor mouse akan bergerak keatas.
9. Untuk metode tracking maka kursor mouse akan selalu bergerak kearah manapun obyek berada.
10. Sedangkan untuk metode trackin berlanjut kursor mouse akan bergerak seperti gerakan pada mouse noteped