

Widget.h

```
#ifndef WIDGET_H
#define WIDGET_H

#include <QWidget>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include "opencv2/features2d/features2d.hpp"
#include "opencv2/calib3d/calib3d.hpp"
#include "opencv2/nonfree/nonfree.hpp"
#include <QTimer>
#include <QImage>
#include <QLineEdit>
#include <QLabel>
#include <QPushButton>
#include <QGridLayout>
#include <iostream>
#include <conio.h>

class Widget : public QWidget
{
    Q_OBJECT

public:
    Widget(QWidget *parent = 0);
    ~Widget();

    cv::VideoCapture capture;
    cv::Mat original;
    cv::Mat mask;

    QLabel *InputLabel;
    QPushButton *PushButton;
    QPushButton *ResetPushbutton;
    QPushButton *ClosePushbutton;
    QLineEdit *Nama;
    QLineEdit *Umur;
    QGridLayout *Layout;
    QTimer *tmrTimer;

    bool test;

private:

public slots:
    void processFrameAndUpdateGUI();
    void on_btnPauseResume_clicked();
    //void KelilingLing(int);
    void Reset();
    void Close();

};

#endif // WIDGET_H

#include "widget.h"
```

Widget.cpp

```
Widget::Widget(QWidget *parent)
    : QWidget(parent)
{
    capture.open(0);
    // Check if successful
    if(!capture.isOpened() == true){
        return;
    }

    InputLabel = new QLabel();
    PushButton = new QPushButton("Pause");
    PushButton->setFixedSize(80,30);
    ResetPushbutton = new QPushButton("Reset");
    ResetPushbutton->setFixedSize(80,30);
    ClosePushbutton = new QPushButton("Close");
    ClosePushbutton->setFixedSize(80,30);
    Nama = new QLineEdit();
    Nama->setStyleSheet("*{color:yellow;background-color:
rgba(0,0,0,0);border:1px transparant;}");
    Nama->setPlaceholderText("Enter Your Name");
    Umur = new QLineEdit();
    Umur->setStyleSheet("*{color:yellow;background-color:
rgba(0,0,0,0);border:1px transparant;}");
    Umur->setPlaceholderText("Enter Your Age");

    Layout = new QGridLayout(this);
    Layout->addWidget(InputLabel,0,0,50,50);
    Layout->addWidget(PushButton, 1,0);
    Layout->addWidget(ResetPushbutton,2,0);
    Layout->addWidget(ClosePushbutton,3,0);
    Layout->addWidget(Nama,1,1);
    Layout->addWidget(Umur,2,1);

    //processFrameAndUpdateGUI();
    tmrTimer = new QTimer();

    connect(tmrTimer,SIGNAL(timeout()),this,SLOT(processFrameAndUpdateGUI()));
    if(test == true)
        tmrTimer->start(0);
    else
        tmrTimer->stop();

    connect(PushButton,SIGNAL(clicked()),this,
SLOT(on_btnPauseResume_clicked()));
    connect(ResetPushbutton,SIGNAL(clicked()),this,SLOT(Reset()));
    connect(ClosePushbutton,SIGNAL(clicked()),this,SLOT(close()));

}

Widget::~~Widget()
{

}

void Widget::processFrameAndUpdateGUI()
{
    capture.read(original);
```

```

    if(original.empty() == true)
        return;

    //cv::cvtColor(original,original,CV_BGR2HLS);//CV_BGR2RGB

    QImage qimgOriginal((uchar*)
original.data,original.cols,original.rows,original.step,QImage::Format_RGB8
88); // for color images

    InputLabel->setPixmap(QPixmap::fromImage(qimgOriginal));
}

void Widget::on_btnPauseResume_clicked()
{
    if(test==true) test = false;
    else test = true;
    if (test == true)
        tmrTimer->start(0);
    else
        tmrTimer->stop();
    int Result=1;
}

void Widget::Reset()
{
    Umur->clear();
    Nama->clear();
}

void Widget::Close()
{
    ClosePushbutton->close();
}

```