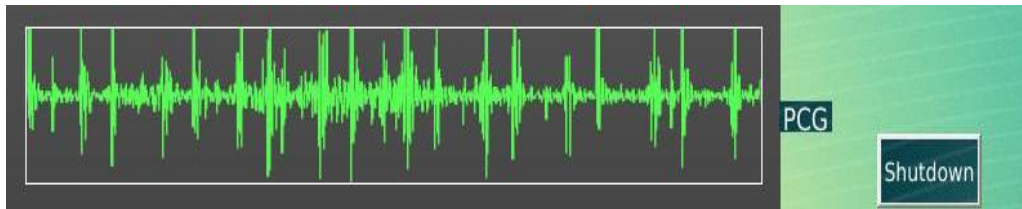
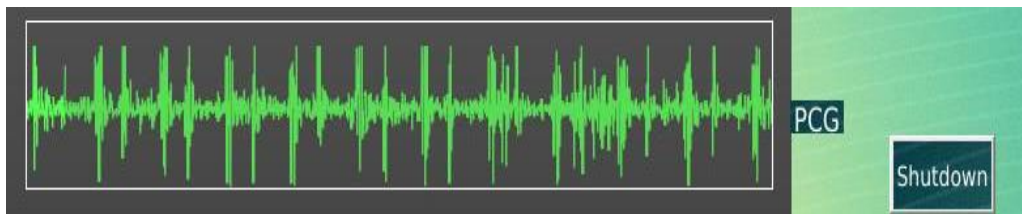


LAMPIRAN

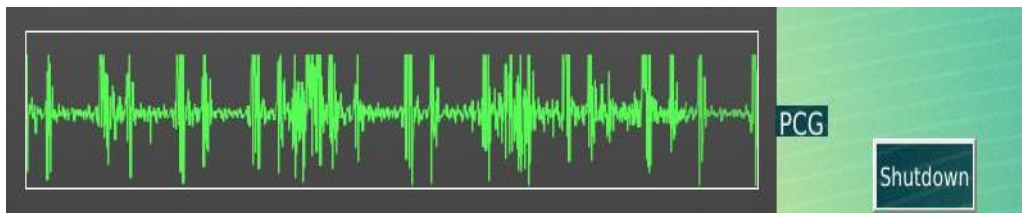
Gambar grafik pasien pertama



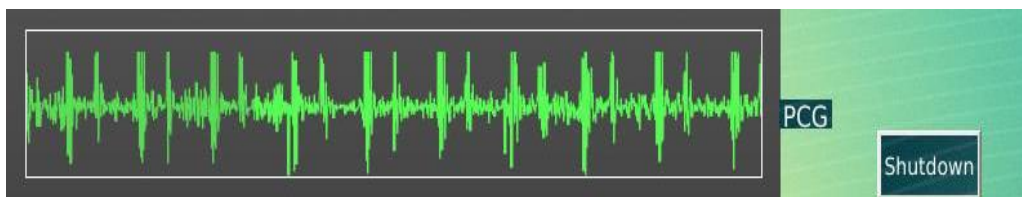
Gambar grafik pasien kedua



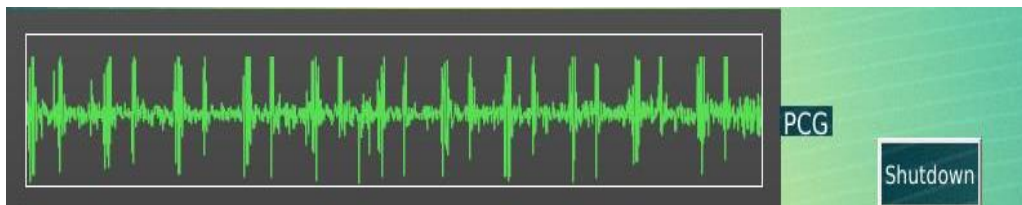
Gambar grafik pasien ketiga



Gambar grafik pasien keempat



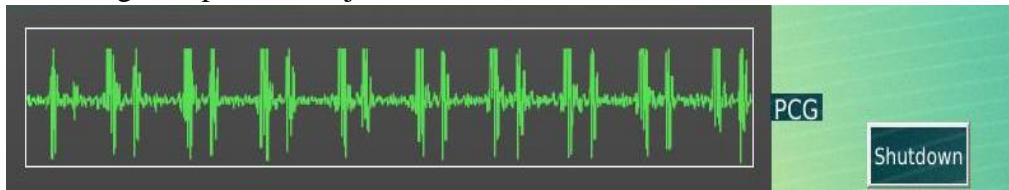
Gambar grafik pasien kelima



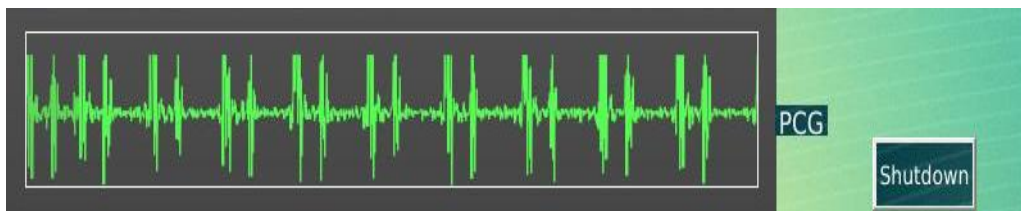
Gambar grafik pasien keenam



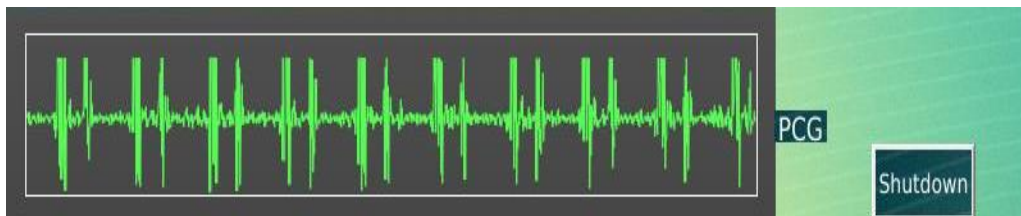
Gambar grafik pasien ketujuh



Gambar grafik pasien kedelapan

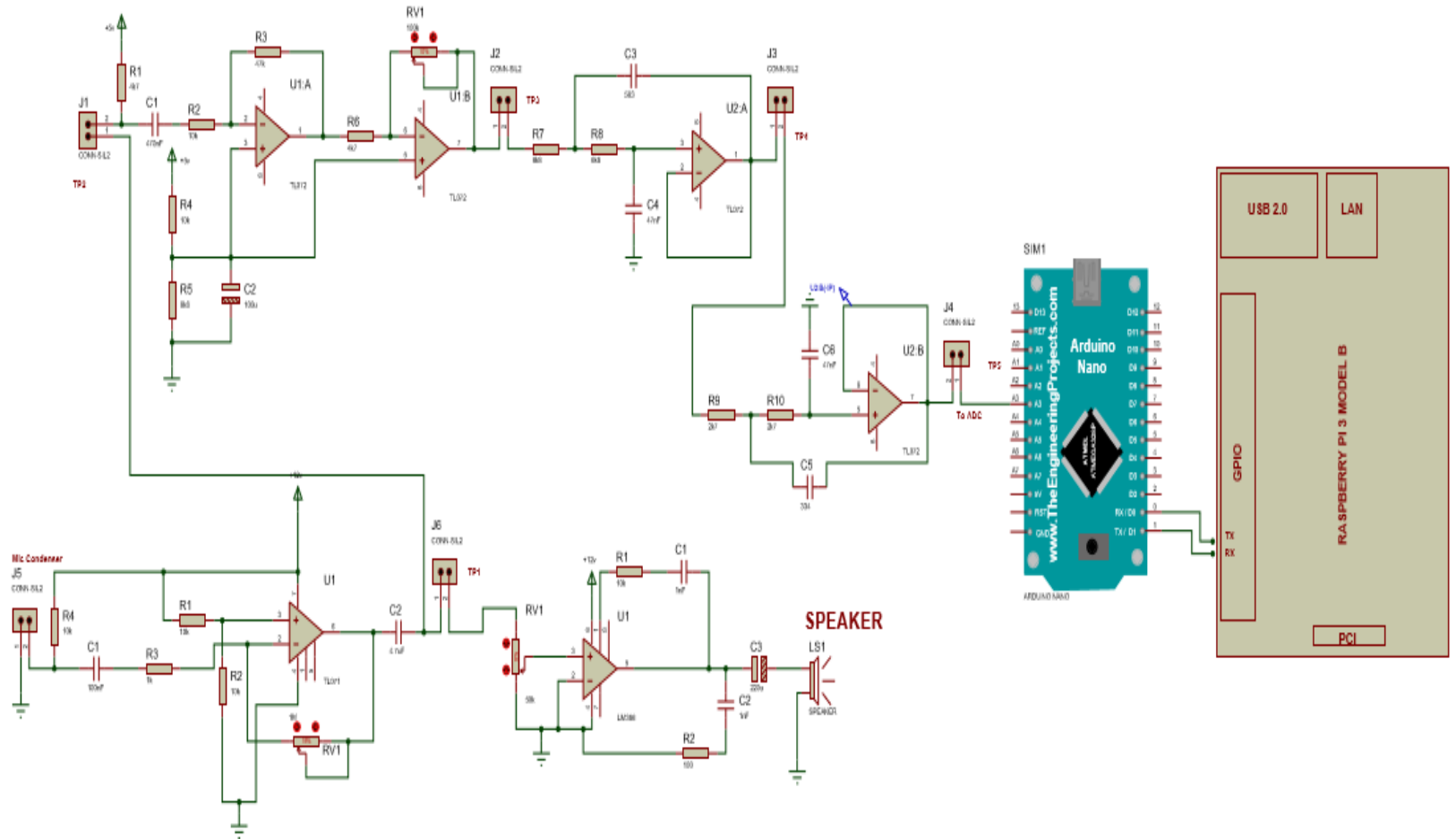


Gambar grafik pasien kesembilan

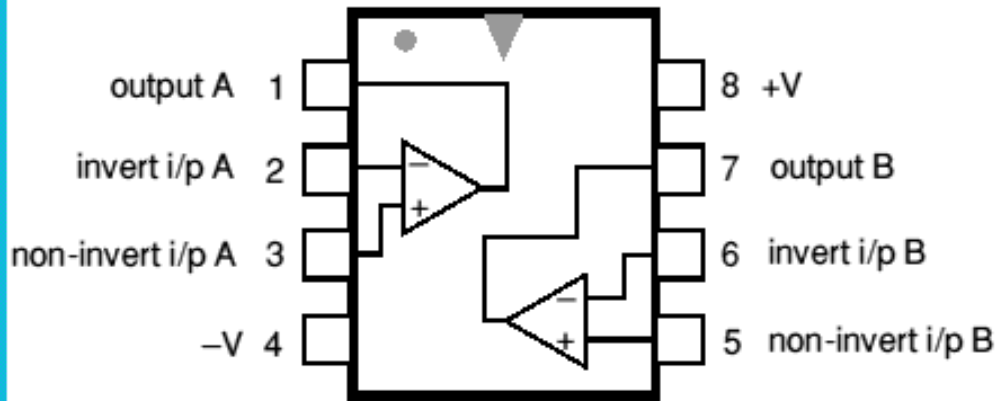


Gambar grafik pasien kesepuluh

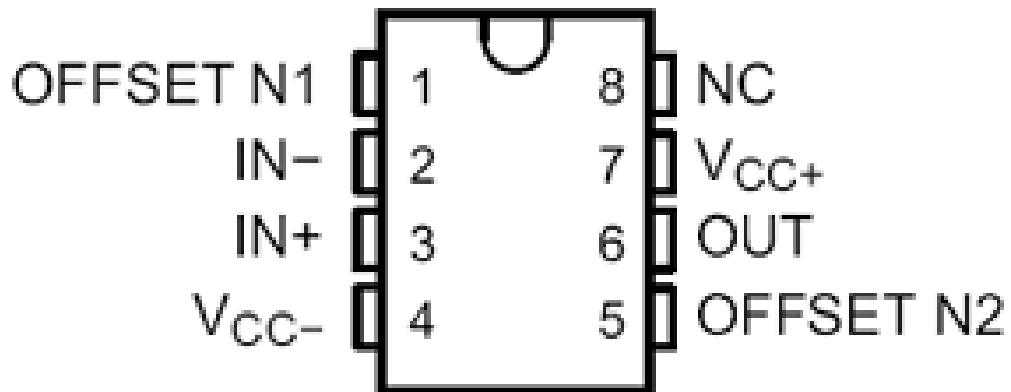


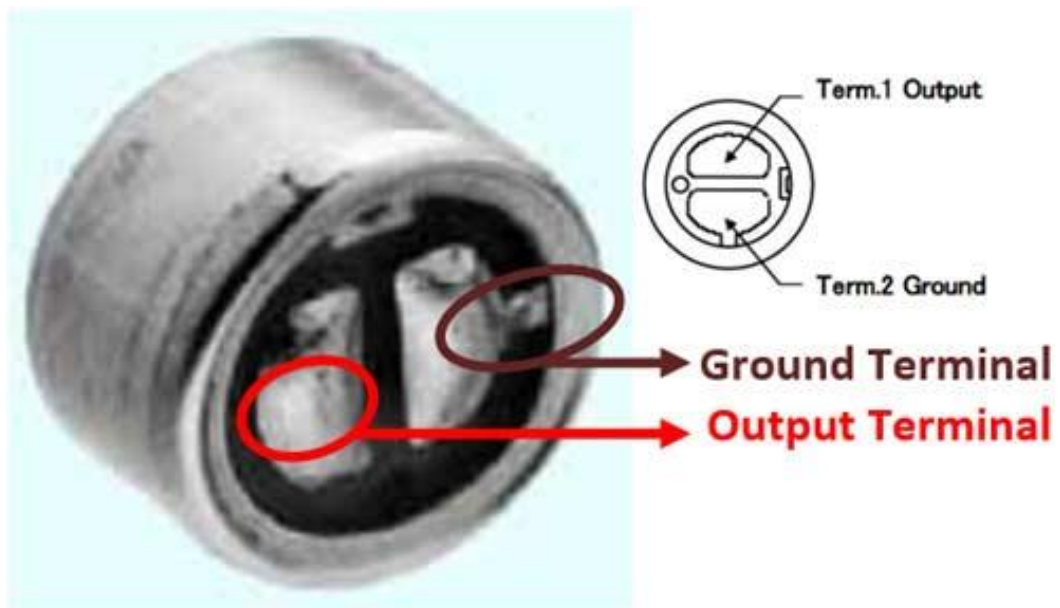
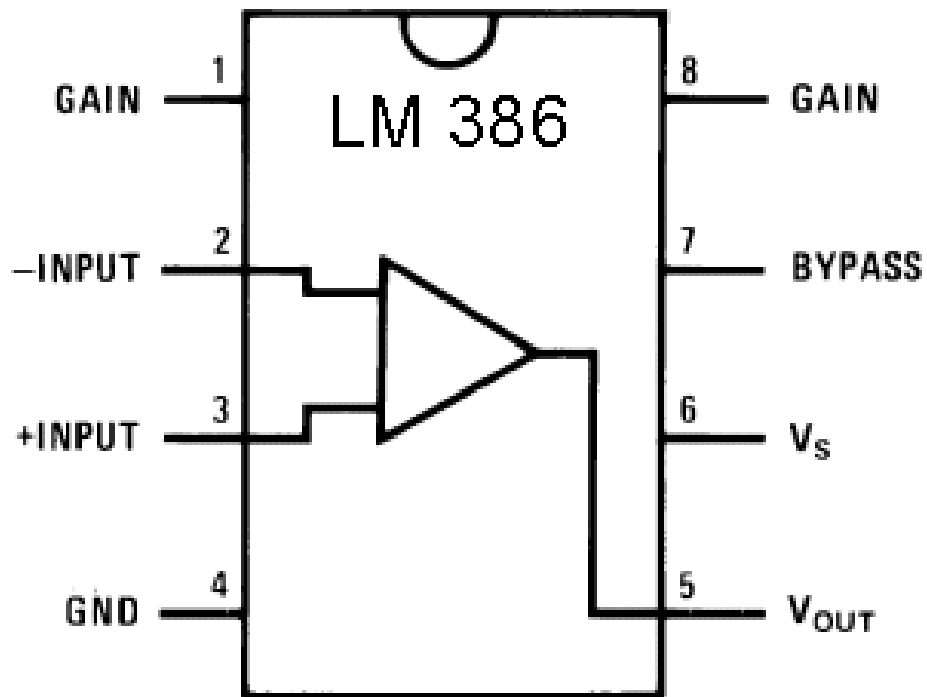


TL072



TL071





Program Arduino Nano

```
int command;
void setup() {
  // put your setup code here, to run once:
  Serial.begin (115200);
}
```

```

void loop() {
  // put your main code here, to run repeatedly:
  double lead_1 = analogRead(A0);
  double lead_2 = analogRead(A1);
  double lead_3 = analogRead(A2);
  double PCG = analogRead(A3);

  if(Serial.available() > 0)
  {
    if(Serial.peek() == 'c')
    {
      Serial.read();
      command = Serial.parseInt();
    }
    while(Serial.available() > 0)
    {
      Serial.read();
    }
  }
  if(command == 1)
  {
    Serial.print(lead_1);
    Serial.println("A");
    Serial.print(lead_2);
    Serial.println("B");
    Serial.print(lead_3);
    Serial.println("C");
    Serial.print(PCG);
    Serial.println("D");
  }
}

```

Program QT

Arduino.CPP

```

#include "arduino.h"

QString saving;
QString serialBuffer;
QStringList list;

Arduino::Arduino(QWidget *parent) : QMainWindow(parent)
{
    qDebug() << "Number of ports: "
    << QSerialPortInfo::availablePorts().length() << "\n";
    //menampilkan port yang terpakai

```

```

        foreach (const QSerialPortInfo &serialPortInfo,
QSerialPortInfo::availablePorts())
        {
            qDebug()<<"Description: " <<serialPortInfo.description()
<< "\n"; //menampilkan deskripsi port
            qDebug()<<"Has vendor id?: "
<<serialPortInfo.hasVendorIdentifier()<<"\n"; //menampilkan
vendor
            qDebug()<<"Vendor ID: "
<<serialPortInfo.vendorIdentifier() <<"\n"; //menampilkan nomor
vendor
            qDebug()<<"Has product id?: "
<<serialPortInfo.hasProductIdentifier() <<"\n";
            qDebug()<<"Product ID: "
<<serialPortInfo.productIdentifier()<<"\n"; //menampilkan nomor
product
        }

        bool arduino_is_available = true; //available arduino
kondisi awal "false"
        QString arduino_uno_port_name; //variabel dengan type data
string

        foreach (const QSerialPortInfo &serialPortInfo,
QSerialPortInfo::availablePorts())
        {
            if(serialPortInfo.hasProductIdentifier() &&
serialPortInfo.hasVendorIdentifier()) //apabila arduino sudah
diketahui nomor produk dan nomor vendor
            {
                if((serialPortInfo.productIdentifier() ==
arduino_uno_product_id) //apabila nomor produk yang diketahui
dan yang telah di setting sesuai
                    &&(serialPortInfo.vendorIdentifier() ==
arduino_uno_vendor_id)) //apabila nomor vendor yang diketahui
dan yang telah disetting sesuai
                {
                    arduino_is_available = true; //apabila
kondisinya sesuai maka available arduino menjadi "true"
                    arduino_uno_port_name =
serialPortInfo.portName(); //variabel arduino_uno_port_name akan
diganti dengan yang telah diketahui
                }
            }
        }
        arduinoserial = new
QextSerialPort(QextSerialPort::EventDriven);
        if(arduino_is_available) //apabila available arduino ==
true?
        {
            qDebug()<<"Found the arduino port...\n";
            arduinoserial->setPortName("ttyS0");//ttyS0
            arduinoserial->open(QextSerialPort::ReadWrite);
            arduinoserial->setBaudRate((BaudRateType(BAUD115200)));
            arduinoserial->setDataBits(DATA_8);

```

```

        arduinoserial->setFlowControl(FLOW_OFF);
        arduinoserial->setParity(PAR_NONE);
        arduinoserial->setStopBits(STOP_1);
        // arduinoserial->waitForReadyRead(5000);

QObject::connect(arduinoserial, SIGNAL(readyRead()), this, SLOT(serialReceiver()));
        arduinoserial->write("c1");
        // arduinoserial->setRts(true);
    }
    else
    {
        qDebug() << "Couldn't find correct port for the
arduino.\n"; //apabila arduino beravaliabile "false" maka akan
ditampilkan
    }
}

void Arduino::serialReceiver() //fungsi
{
    QString readData;
    QByteArray serialData;

    if(arduinoserial->bytesAvailable())
    {
        serialData = arduinoserial->readAll();
        QString dataserial ;
        dataserial
=QString::fromStdString(serialData.toStdString());
        emit alldata(dataserial);
        serialBuffer =
QString::fromStdString(serialData.toStdString());
        emit data_realtime(serialBuffer);
        //serialBuffer="";
        //arduinoserial->waitForReadyRead(1000);
    }
}

```

Data Save

```

#include "formpatientdatasave.h"
#include "ui_formpatientdatasave.h"

QString value="";
FormPatientDataSave::FormPatientDataSave(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::FormPatientDataSave)
{
    ui->setupUi(this);
}

FormPatientDataSave::~FormPatientDataSave()
{
    delete ui;
}

```



```
void FormPatientDataSave::on_pushButton_clicked()
{
    this->close();
}

void FormPatientDataSave::on_pushButton_12_clicked()
{
    value=value+"0";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_3_clicked()
{
    value=value+"1";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_4_clicked()
{
    value=value+"2";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_5_clicked()
{
    value=value+"3";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_6_clicked()
{
    value=value+"4";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_7_clicked()
{
    value=value+"5";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_8_clicked()
{
    value=value+"6";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_9_clicked()
{
    value=value+"7";
    ui->lineEdit->setText(value);
}

void FormPatientDataSave::on_pushButton_10_clicked()
{
    value=value+"8";
```

```

        ui->lineEdit->setText (value);
    }

void FormPatientDataSave::on_pushButton_11_clicked()
{
    value=value+"9";
    ui->lineEdit->setText (value);
}

void FormPatientDataSave::on_pushButton_13_clicked()
{
    ui->lineEdit->clear();
    value="";
}

void FormPatientDataSave::on_pushButton_2_clicked()
{
    if (value=="")
    {
        QMessageBox::warning(this,"title","file not open");
    }
    else
    {
        emit sendnumberpatient (value);
        this->close();
    }
}

```

Open Data Save

```

#include "formopenfile.h"
#include "ui_formopenfile.h"

QStringList listsavedata,listdataopen;
int i=0;
int plot1,plot2,plot3,plot4;
int hitung=0;
double plottertime;
int second_number_file;
int average_plot_open;
bool pulse_high_open=false;
bool pulse_low_open=false;
int a_open;
int search_open;
int nomorgambar_open;

volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int R = 82;
volatile int T = 82;
volatile int thresh = 103;
volatile int amp = 50;
volatile bool firstBeat = true;
volatile bool secondBeat = false;
volatile int Signal;

```

```

volatile int IBI = 600;
volatile bool Pulse = false;
volatile int BPM;

QString readdatasave;
QString number_file_second;
FormOpenFile::FormOpenFile(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::FormOpenFile)
{
    ui->setupUi(this);
    makeplot_openfile_1();
    makeplot_openfile_2();
    makeplot_openfile_3();
    makeplot_openfile_4();
    browserfolder = new FormBrowserFolder(this);

    connect(browserfolder, SIGNAL(sendspath(QString)), this, SLOT(receivepathfile(QString)));

    connect(browserfolder, SIGNAL(sendfilename(QString)), this, SLOT(receivefilename(QString)));

    timerplot = new QTimer(this);

    connect(timerplot, SIGNAL(timeout()), this, SLOT(timerplotting()));
    timerplot->start(0);

    ui->horizontalScrollBar->setRange(0, 200);
    connect(ui->horizontalScrollBar, SIGNAL(valueChanged(int)),
    this, SLOT(horzScrollBarChanged(int)));

    ui->progressBar->setValue(0);
    ui->progressBar->setEnabled(false);

    ui->pushButton_3->setEnabled(false);
}

FormOpenFile::~FormOpenFile()
{
    delete ui;
}

void FormOpenFile::receivefilename(QString y)
{
    if(y.contains(".txt"))
    {
        y=y.replace(".txt", "").trimmed();
    }
    ui->label_2->setAlignment(Qt::AlignCenter);
    ui->label_2->setText(y);
}

```

```

void FormOpenFile::makeplot_openfile_1()
{
    //Untuk memberi warna pada plot
    ui->customPlot_openfile->addGraph();
    ui->customPlot_openfile->graph(0)-
>setPen(QPen(QColor(0,0,0)));

    //Untuk memberi range X dan Y
    ui->customPlot_openfile->xAxis->setRange(0,60);
    ui->customPlot_openfile->yAxis->setRange(0,1100);
    ui->customPlot_openfile->xAxis2->setVisible(true);
    ui->customPlot_openfile->xAxis2->setTickLabels(false);
    ui->customPlot_openfile->yAxis2->setVisible(true);
    ui->customPlot_openfile->yAxis2->setTickLabels(false);

    connect(ui->customPlot_openfile->xAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile->xAxis2,
    SLOT(setRange(QCPRange)));
    connect(ui->customPlot_openfile->yAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile->yAxis2,
    SLOT(setRange(QCPRange)));

    ui->customPlot_openfile->xAxis2-
>setBasePen(QPen(Qt::black,1));
    ui->customPlot_openfile->yAxis2-
>setBasePen(QPen(Qt::black,1));
    ui->customPlot_openfile->xAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_openfile->yAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_openfile->xAxis2->setSubTickPen(Qt::NoPen);
    ui->customPlot_openfile->yAxis2->setSubTickPen(Qt::NoPen);

    ui->customPlot_openfile->xAxis->setBasePen(QPen(Qt::black,
1));
    ui->customPlot_openfile->yAxis->setBasePen(QPen(Qt::black,
1));
    ui->customPlot_openfile->xAxis->setTickPen(Qt::NoPen);
    ui->customPlot_openfile->yAxis->setTickPen(Qt::NoPen);
    ui->customPlot_openfile->xAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_openfile->yAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_openfile->xAxis->setTickLabels(false);
    ui->customPlot_openfile->yAxis->setTickLabels(false);
    ui->customPlot_openfile->xAxis->grid()-
>setPen(QPen(Qt::NoPen));
    ui->customPlot_openfile->yAxis->grid()-
>setPen(QPen(Qt::NoPen));
    ui->customPlot_openfile->xAxis->grid()-
>setSubGridPen(Qt::NoPen);
    ui->customPlot_openfile->yAxis->grid()-
>setSubGridPen(Qt::NoPen);
    ui->customPlot_openfile->xAxis->grid()-
>setSubGridVisible(false);
    ui->customPlot_openfile->yAxis->grid()-
>setSubGridVisible(false);
    ui->customPlot_openfile->xAxis->grid()-
>setZeroLinePen(Qt::NoPen);

```

```

    ui->customPlot_openfile->yAxis->grid() -
>setZeroLinePen(QPen(Qt::NoPen));

    QPixmap pixmap;
    pixmap.load("/home/pi/ECG_Layout_fix.png");
    ui->customPlot_openfile-
>setBackground(pixmap.scaled(599,115),Qt::KeepAspectRatioByExpanding);

    QLinearGradient plotGradient;
    plotGradient.setStart(0, 0);
    plotGradient.setFinalStop(0, 350);
    plotGradient.setColorAt(0, QColor(255,255,255));
    // plotGradient.setColorAt(1, QColor(50, 50, 50));
    ui->customPlot_openfile->setBackground(plotGradient);
}

void FormOpenFile::makeplot_openfile_2()
{
    //Untuk memberi warna pada plot
    ui->customPlot_openfile_2->addGraph();
    ui->customPlot_openfile_2->graph(0)-
>setPen(QPen(QColor(0,0,0)));

    //Untuk memberi range X dan Y
    ui->customPlot_openfile_2->xAxis->setRange(0,60);
    ui->customPlot_openfile_2->yAxis->setRange(0,1055);
    ui->customPlot_openfile_2->xAxis2->setVisible(true);
    ui->customPlot_openfile_2->xAxis2->setTickLabels(false);
    ui->customPlot_openfile_2->yAxis2->setVisible(true);
    ui->customPlot_openfile_2->yAxis2->setTickLabels(false);

    connect(ui->customPlot_openfile_2->xAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile_2-
>xAxis2, SLOT(setRange(QCPRange)));
    connect(ui->customPlot_openfile_2->yAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile_2-
>yAxis2, SLOT(setRange(QCPRange)));

    ui->customPlot_openfile_2->xAxis2-
>setBasePen(QPen(Qt::black,1));
    ui->customPlot_openfile_2->yAxis2-
>setBasePen(QPen(Qt::black,1));
    ui->customPlot_openfile_2->xAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_2->yAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_2->xAxis2->setSubTickPen(Qt::NoPen);
    ui->customPlot_openfile_2->yAxis2->setSubTickPen(Qt::NoPen);

    ui->customPlot_openfile_2->xAxis->setBasePen(QPen(Qt::black,
1));
    ui->customPlot_openfile_2->yAxis->setBasePen(QPen(Qt::black,
1));
    ui->customPlot_openfile_2->xAxis->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_2->yAxis->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_2->xAxis->setSubTickPen(Qt::NoPen);

```

```

    ui->customPlot_openfile_2->yAxis->setSubTickPen(QPen(Qt::NoPen));
    ui->customPlot_openfile_2->xAxis->setTickLabels(false);
    ui->customPlot_openfile_2->yAxis->setTickLabels(false);
    ui->customPlot_openfile_2->xAxis->grid()-
>setPen(QPen(Qt::NoPen));
    ui->customPlot_openfile_2->yAxis->grid()-
>setPen(QPen(Qt::NoPen));
    ui->customPlot_openfile_2->xAxis->grid()-
>setSubGridPen(QPen(Qt::NoPen));
    ui->customPlot_openfile_2->yAxis->grid()-
>setSubGridPen(QPen(Qt::NoPen));
    ui->customPlot_openfile_2->xAxis->grid()-
>setSubGridVisible(false);
    ui->customPlot_openfile_2->yAxis->grid()-
>setSubGridVisible(false);
    ui->customPlot_openfile_2->xAxis->grid()-
>setZeroLinePen(QPen(Qt::NoPen));
    ui->customPlot_openfile_2->yAxis->grid()-
>setZeroLinePen(QPen(Qt::NoPen));

    QPixmap pixmap;
    pixmap.load("/home/pi/ECG_Layout_fix.png");
    ui->customPlot_openfile_2-
>setBackground(pixmap.scaled(599,115),Qt::KeepAspectRatioByExpanding);

    QLinearGradient plotGradient;
    plotGradient.setStart(0, 0);
    plotGradient.setFinalStop(0, 350);
    plotGradient.setColorAt(0, QColor(255, 255, 255));
    // plotGradient.setColorAt(1, QColor(50, 50, 50));
    ui->customPlot_openfile_2->setBackground(plotGradient);
}

void FormOpenFile::makeplot_openfile_3()
{
    //Untuk memberi warna pada plot
    ui->customPlot_openfile_3->addGraph();
    ui->customPlot_openfile_3->graph(0)-
>setPen(QPen(QColor(0,0,0)));

    //Untuk memberi range X dan Y
    ui->customPlot_openfile_3->xAxis->setRange(0,60);
    ui->customPlot_openfile_3->yAxis->setRange(0,850);
    ui->customPlot_openfile_3->xAxis2->setVisible(true);
    ui->customPlot_openfile_3->xAxis2->setTickLabels(false);
    ui->customPlot_openfile_3->yAxis2->setVisible(true);
    ui->customPlot_openfile_3->yAxis2->setTickLabels(false);

    connect(ui->customPlot_openfile_3->xAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile_3-
>xAxis2, SLOT(setRange(QCPRange)));
    connect(ui->customPlot_openfile_3->yAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile_3-
>yAxis2, SLOT(setRange(QCPRange)));
}

```

```

    ui->customPlot_openfile_3->xAxis2-
>setBasePen(QPen(Qt::black,1));
    ui->customPlot_openfile_3->yAxis2-
>setBasePen(QPen(Qt::black,1));
    ui->customPlot_openfile_3->xAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_3->yAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_3->xAxis2->setSubTickPen(Qt::NoPen);
    ui->customPlot_openfile_3->yAxis2->setSubTickPen(Qt::NoPen);

    ui->customPlot_openfile_3->xAxis->setBasePen(QPen(Qt::black,
1));
    ui->customPlot_openfile_3->yAxis->setBasePen(QPen(Qt::black,
1));
    ui->customPlot_openfile_3->xAxis->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_3->yAxis->setTickPen(Qt::NoPen);
    ui->customPlot_openfile_3->xAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_openfile_3->yAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_openfile_3->xAxis->setTickLabels(false);
    ui->customPlot_openfile_3->yAxis->setTickLabels(false);
    ui->customPlot_openfile_3->xAxis->grid()-
>setPen(QPen(Qt::NoPen));
    ui->customPlot_openfile_3->yAxis->grid()-
>setPen(QPen(Qt::NoPen));
    ui->customPlot_openfile_3->xAxis->grid()-
>setSubGridPen(Qt::NoPen);
    ui->customPlot_openfile_3->yAxis->grid()-
>setSubGridPen(Qt::NoPen);
    ui->customPlot_openfile_3->xAxis->grid()-
>setSubGridVisible(false);
    ui->customPlot_openfile_3->yAxis->grid()-
>setSubGridVisible(false);
    ui->customPlot_openfile_3->xAxis->grid()-
>setZeroLinePen(Qt::NoPen);
    ui->customPlot_openfile_3->yAxis->grid()-
>setZeroLinePen(Qt::NoPen);

    QPixmap pixmap;
    pixmap.load("/home/pi/ECG_Layout_fix.png");
    ui->customPlot_openfile_3-
>setBackground(pixmap.scaled(599,115),Qt::KeepAspectRatioByExpan
ding);

    QLinearGradient plotGradient;
    plotGradient.setStart(0, 0);
    plotGradient.setFinalStop(0, 350);
    plotGradient.setColorAt(0, QColor(255, 255, 255));
    // plotGradient.setColorAt(1, QColor(50, 50, 50));
    ui->customPlot_openfile_3->setBackground(plotGradient);
}

void FormOpenFile::makeplot_openfile_4()
{
    //Untuk memberi warna pada plot
    ui->customPlot_openfile_4->addGraph();
    ui->customPlot_openfile_4->graph(0)->setPen(QPen(QColor(88,
255, 85)));

```

```

//Untuk memberi range X dan Y
ui->customPlot_openfile_4->xAxis->setRange(0,60);
ui->customPlot_openfile_4->yAxis->setRange(0,900);
ui->customPlot_openfile_4->xAxis2->setVisible(true);
ui->customPlot_openfile_4->xAxis2->setTickLabels(false);
ui->customPlot_openfile_4->yAxis2->setVisible(true);
ui->customPlot_openfile_4->yAxis2->setTickLabels(false);

connect(ui->customPlot_openfile_4->xAxis,
        SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile_4-
>xAxis2, SLOT(setRange(QCPRange)));
connect(ui->customPlot_openfile_4->yAxis,
        SIGNAL(rangeChanged(QCPRange)),ui->customPlot_openfile_4-
>yAxis2, SLOT(setRange(QCPRange)));

ui->customPlot_openfile_4->xAxis2-
>setBasePen(QPen(Qt::white,1));
ui->customPlot_openfile_4->yAxis2-
>setBasePen(QPen(Qt::white,1));
ui->customPlot_openfile_4->xAxis2->setTickPen(Qt::NoPen);
ui->customPlot_openfile_4->yAxis2->setTickPen(Qt::NoPen);
ui->customPlot_openfile_4->xAxis2->setSubTickPen(Qt::NoPen);
ui->customPlot_openfile_4->yAxis2->setSubTickPen(Qt::NoPen);

ui->customPlot_openfile_4->xAxis->setBasePen(QPen(Qt::white,
1));
ui->customPlot_openfile_4->yAxis->setBasePen(QPen(Qt::white,
1));
ui->customPlot_openfile_4->xAxis->setTickPen(Qt::NoPen);
ui->customPlot_openfile_4->yAxis->setTickPen(Qt::NoPen);
ui->customPlot_openfile_4->xAxis->setSubTickPen(Qt::NoPen);
ui->customPlot_openfile_4->yAxis->setSubTickPen(Qt::NoPen);
ui->customPlot_openfile_4->xAxis->setTickLabels(false);
ui->customPlot_openfile_4->yAxis->setTickLabels(false);
ui->customPlot_openfile_4->xAxis->grid()-
>setPen(QPen(Qt::NoPen));
ui->customPlot_openfile_4->yAxis->grid()-
>setPen(QPen(Qt::NoPen));
ui->customPlot_openfile_4->xAxis->grid()-
>setSubGridPen(Qt::NoPen);
ui->customPlot_openfile_4->yAxis->grid()-
>setSubGridPen(Qt::NoPen);
ui->customPlot_openfile_4->xAxis->grid()-
>setSubGridVisible(false);
ui->customPlot_openfile_4->yAxis->grid()-
>setSubGridVisible(false);
ui->customPlot_openfile_4->xAxis->grid()-
>setZeroLinePen(Qt::NoPen);
ui->customPlot_openfile_4->yAxis->grid()-
>setZeroLinePen(Qt::NoPen);

QLinearGradient plotGradient;
plotGradient.setStart(0, 0);
plotGradient.setFinalStop(0, 350);
plotGradient.setColorAt(0, QColor(80, 80, 80));

```



```

        plotGradient.setColorAt(1, QColor(50, 50, 50));
        ui->customPlot_openfile_4->setBackground(plotGradient);
    }

void FormOpenFile::timerplotting()
{
    QString readData;
    // qDebug()<<listsavedata.count();

    if(hitung==0)
    {
        hitung = listsavedata.count();
    }
    if(i<hitung&&hitung>0)
    {
        ui->progressBar->setEnabled(true);
        ui->progressBar->setRange(0, hitung);
        for(int a =0; a<hitung; a++)
        {
            ui->pushButton_2->setEnabled(false);
            ui->pushButton_3->setEnabled(false);
            i++;
            ui->progressBar->setValue(i);
            plotvertime++;
            readData = listsavedata.value(i);
            if(readData.contains("A"))
            {
                readData = readData.replace("A", "").trimmed();
                double in = readData.toDouble();
                // qDebug()<< in;
                if(in>9&&in<1024)
                {
                    //qDebug()<<"nilai A"<<in<<i<<"/"<<hitung;
                    plot1=in;
                }
            }
            else if (readData.contains("B"))
            {
                readData = readData.replace("B", "").trimmed();
                double in = readData.toDouble();
                // qDebug()<< in;

                if(in>9)
                {
                    //qDebug()<<"nilai B"<<in;
                    plot2=in;
                    // qDebug()<<plot2;
                    Signal = plot2;
                    for(int i=0; i<100; i++)
                    {
                        average_plot_open +=Signal;
                    }
                    average_plot_open = average_plot_open/100;
                    // qDebug()<<average_lead;
                }
            }
        }
    }
}

```

```

if(average_plot_open>630&&pulse_high_open==false)
{
    pulse_high_open=true;
    pulse_low_open=true;
    // qDebug()<<"pulse high";
}
if(average_plot_open<450&&pulse_low_open==true)
{
    pulse_low_open=false;
    pulse_high_open=false;
    for(int i=0;i<1;i++)
    {
        a_open++;
        search_open++;
        qDebug()<<search_open;
    }
}
}
else if (readData.contains("C"))
{
    readData = readData.replace("C","").trimmed();
    double in = readData.toDouble();
    // qDebug()<<in;

    if(in>9)
    {
        //qDebug()<<"nilai C"<<in;
        plot3=in;
    }
}
else if (readData.contains("D"))
{
    readData = readData.replace("D","").trimmed();
    double in = readData.toDouble();
    // qDebug()<<in;
    if(in>9)
    {
        //qDebug()<<"nilai D"<<in;
        plot4=in;
    }
}
double key = plottertime/1190.0; // time elapsed since
start of demo, in seconds
//double input = 1;

static double lastPointKey = 0;
if (key-lastPointKey > 0.002) // at most add point every
2 ms
{
    ui->customPlot_openfile->graph(0)-
>addData(key,plot1);
    ui->customPlot_openfile->xAxis->setRange(key/2.0, 8,
Qt::AlignRight);
}

```

```

        // qDebug()<<"nilai key"<<key;

        ui->customPlot_openfile_2->graph(0)-
>addData(key,plot2);
        ui->customPlot_openfile_2->xAxis->setRange(key/2.0,
8, Qt::AlignRight);

        ui->customPlot_openfile_3->graph(0)-
>addData(key,plot3);
        ui->customPlot_openfile_3->xAxis->setRange(key/2.0,
8, Qt::AlignRight);

        ui->customPlot_openfile_4->graph(0)-
>addData(key,plot4);
        ui->customPlot_openfile_4->xAxis->setRange(key/2.0,
8, Qt::AlignRight);

    }

}

double bpm=search_open;
ui->label_5->setAlignment(Qt::AlignCenter);
ui->label_5->setText(QString::number(bpm,'f',0));
ui->pushButton_2->setEnabled(true);
ui->pushButton_3->setEnabled(false);
ui->customPlot_openfile->replot();
ui->customPlot_openfile_2->replot();
ui->customPlot_openfile_3->replot();
ui->customPlot_openfile_4->replot();

}

}

void FormOpenFile::receivespathfile(QString x)
{
    sPath = x;
}

void FormOpenFile::on_pushButton_clicked()
{
    this->close();
}

void FormOpenFile::on_pushButton_2_clicked()
{

```

```

browserfolder->show();
readdatasave="";
ui->customPlot_openfile->graph(0)->clearData();
ui->customPlot_openfile_2->graph(0)->clearData();
ui->customPlot_openfile_3->graph(0)->clearData();
ui->customPlot_openfile_4->graph(0)->clearData();
ui->customPlot_openfile->replot();
ui->customPlot_openfile_2->replot();
ui->customPlot_openfile_3->replot();
ui->customPlot_openfile_4->replot();

ui->progressBar->setValue(0);
ui->horizontalScrollBar->setValue(0);
i=0;
plottervertime=0;

listsavedata.clear();
hitung=0;
sampleCounter = 0;
lastBeatTime = 0;
R = 82;
T = 82;
thresh = 103;
amp = 0;
firstBeat = true;
secondBeat = false;
Signal=0;
IBI = 600;
Pulse = false;
BPM=0;
ui->label_5->setText(QString::number(BPM));

ui->pushButton_3->setEnabled(true);

a_open=0;
search_open=0;
}

void FormOpenFile::on_pushButton_3_clicked()
{
    i=0;

    QFile file (sPath);
    if(!file.open(QIODevice::ReadOnly|QIODevice::Text))
    {
        QMessageBox::warning(this,"title","file not open");
    }
    QTextStream datain(&file);
    readdatasave=datain.readAll();
    listsavedata = readdatasave.split("|");
    //qDebug()<<list;
    file.close();
}
}

```

```

void FormOpenFile::horzScrollBarChanged(int value)
{
    if (qAbs(ui->customPlot_openfile->xAxis->range().center()-
value/100.0) > 0.01) // if user is dragging customPlot, we
don't want to re customPlot twice
    {
        ui-> customPlot_openfile->xAxis->setRange(value/10.0, ui-
>customPlot_openfile->xAxis->range().size(), Qt::AlignCenter);
        ui-> customPlot_openfile->replot();
    }
    if (qAbs(ui->customPlot_openfile_2->xAxis->range().center()-
value/100.0) > 0.01) // if user is dragging customPlot, we
don't want to re customPlot twice
    {
        ui-> customPlot_openfile_2->xAxis->setRange(value/10.0,
ui->customPlot_openfile_2->xAxis->range().size(),
Qt::AlignCenter);
        ui-> customPlot_openfile_2->replot();
    }
    if (qAbs(ui->customPlot_openfile_3->xAxis->range().center()-
value/100.0) > 0.01) // if user is dragging customPlot, we
don't want to re customPlot twice
    {
        ui-> customPlot_openfile_3->xAxis->setRange(value/10.0,
ui->customPlot_openfile_3->xAxis->range().size(),
Qt::AlignCenter);
        ui-> customPlot_openfile_3->replot();
    }
    if (qAbs(ui->customPlot_openfile_4->xAxis->range().center()-
value/100.0) > 0.01) // if user is dragging customPlot, we
don't want to re customPlot twice
    {
        ui-> customPlot_openfile_4->xAxis->setRange(value/10.0,
ui->customPlot_openfile_4->xAxis->range().size(),
Qt::AlignCenter);
        ui-> customPlot_openfile_4->replot();
    }
}

void FormOpenFile::on_pushButton_4_clicked()
{
    QString file_Name;
    QString istring;
    for(int i=0;i<1;i++)
    {
        second_number_file++;
        number_file_second =
QString::number(second_number_file);
    }

    for(int i=0;i<4;i++)
    {
        istring = QString::number(i+1);
    }
}

```

```

        if (i<3)
        {

file_Name="/home/pi/Datagambar/lead"+istring+" (" +number_file_sec
ond+").png";
        }
        else
        {

file_Name="/home/pi/Datagambar/Pcg (" +number_file_second+").png";
        }
        QFile file_png(file_Name);

        if (!file_png.open(QIODevice::WriteOnly))
        {
            qDebug() << file_png.errorString();
        }
        else if(istring=="1")
        {
            ui->customPlot_openfile->savePng(file_Name);
        }
        else if(istring=="2")
        {
            ui->customPlot_openfile_2->savePng(file_Name);
        }
        else if(istring=="3")
        {
            ui->customPlot_openfile_3->savePng(file_Name);
        }
        else
        {
            ui->customPlot_openfile_4->savePng(file_Name);
        }
    }
}

void FormOpenFile::on_pushButton_5_clicked()
{
    for(int i =0;i<1;i++)
    {
        nomorgambar_open++;
        QFile
file_ss("/home/pi/ScreenshotDataSave/ScreenshotDataSave"+QString
::number(nomorgambar_open)+".jpeg");

        if (!file_ss.open(QIODevice::WriteOnly))
        {
            qDebug() << file_ss.errorString();
        }
        else
        {
            QPixmap pixmap = QPixmap::grabWidget(this);
            QImage image = pixmap.toImage();

```

```

image.save("/home/pi/ScreenshotDataSave/ScreenshotDataSave"+QString::number(nomorgambar_open)+".jpeg");
    }
}

```

Main CPP

```

#include "mainwindow.h"
#include "ui_mainwindow.h"

int search, average_lead, a=0;
bool pulse_high=false;
bool pulse_low=false;
/*Mendeklarasikan variabel*/
int command = 0;
double inputlead1, inputlead2, inputlead3, inputPCG; //variabel dengan type data double, diletakkan diluar fungsi untuk bisa digunakan diberbagai fungsi
double settingsms = 200.0;
double key1=0;
double key2=0;
int changegraph=0;
double keyswitch=60;
int saving_enable = 0;
int second_number_file_realtime;
int detik_bpm;
int menit_bpm;
int nomorgambar=0;
int R_value=40;
int T_value=40;
int thresh_value=72;
double time_plot;
int detik_realtime;
//Mendeklarasikan variabel

    QString serial, readData;
    QStringList list_data;
    QByteArray byteArray;
    QBuffer buffer;
    QTextStream out(&buffer);
    QTextStream in(&buffer);
    QString number_file_second_realtime;
    QString filename_screenshot;
    QString text, test;

int gel_search_1, gel_search_2, gel_high, gel_reference;
int count, count_result;
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    /*Memanggil fungasi setting grafik*/
    MainWindow::makePlot();
}

```

```

MainWindow::makePlot_2();
MainWindow::makePlot_3();
MainWindow::makePlot_4();

/*Pemanggilan ulang QTimer*/
qtimer=new QTimer(this); //pembaruan class
connect(qtimer, SIGNAL(timeout()), this, SLOT(myTimer()));
//untuk menghubungkan class qtimer dengan slots
qtimer->start(0); // interval 0 atau melakukan refresh
secara cepat

qtimer_savedata = new QTimer(this);

connect(qtimer_savedata, SIGNAL(timeout()), this, SLOT(timersavedata()));
qtimer_savedata->start(1000);

ar=new Arduino(this); //pembaruan class
patientdatasave =new FormPatientDataSave(this);
openfile = new FormOpenFile(this);

/*Menghubungkan Arduino class dengan mainwindow class*/

connect(ar, SIGNAL(alldata(QString)), this, SLOT(receivealldata(QString)));

connect(patientdatasave, SIGNAL(sendnumberpatient(QString)), this, SLOT(receivenumberpatient(QString)));

connect(ar, SIGNAL(data_realtime(QString)), this, SLOT(receive_data_realtime(QString)));

dw = new Dialog_Warning(this);

ui->progressBar->setRange(0,59);
ui->progressBar->setValue(0);

}

MainWindow::~MainWindow()
{
    delete ui;
}

/*Menerima data signal dari Arduino class*/

void MainWindow::receivealldata(QString i)
{
    if(patientnumber=="")
    {
        detik=0;
        menit=0;
    }
}

```



```

}
else if(patientnumber!="" &&menit<1)
{

    ui->progressBar->setValue(detik);
    text =i;

    text = text.replace("\n", "|").trimmed();
    test +=text;
    qDebug()<<"detik"<<detik;

}

else if (patientnumber!=""&&menit==1)
{
    QString coba;
    coba=test;
    qDebug()<<coba;

    QFile file
("/home/pi/DataPasien/"+patientnumber+".txt");
    if(!file.open(QIODevice::WriteOnly|QIODevice::Text))
    {
        QMessageBox::warning(this,"title","file not open");
    }
    QTextStream dataout(&file);
    dataout << coba;
    file.flush();
    file.close();
    patientnumber="";
    text ="";
    test = "";
}

else
{
    patientnumber="";
    detik=0;
    menit=0;
}
}

void MainWindow::receivenumberpatient(QString x)
{
    patientnumber=x;
}

void MainWindow::timersavedata()
{
    detik_realtime++;

    detik++;
    if(detik>59)
    {

```

```

        menit++;
        detik=0;
    }
    //qDebug()<<"detik ="<<detik<<"menit ="<<menit;
    detik_bpm++;
    //qDebug()<<detik_bpm;
    if (detik_bpm>59)
    {
        double bpm=search;
        qDebug()<<"bpm"<<bpm;
        ui->label_7->setText(QString::number(bpm,'f',0));
        search=0;
        detik_bpm=0;
    }
}

void MainWindow::receive_data_realtime(QString s)
{
    if(detik_realtime<1)
    {
        serial+=s;
    }
    else
    {
        list_data = serial.split("\n");
        for(int i=0;i<list_data.count();i++)
        {
            time_plot++;
            readData = list_data.value(i);

            //qDebug()<<serialBuffer.count();
            //qDebug()<<serialBuffer;
            if(readData.contains("A"))
            {
                readData = readData.replace("A","").trimmed();
                double in = readData.toDouble();
                // qDebug()<< in;
                if(in>9)
                {
                    //qDebug()<<"nilai A"<<in;
                    inputlead1=in;
                    //qDebug()<<inputlead1;
                }
            }
            else if (readData.contains("B"))
            {
                readData = readData.replace("B","").trimmed();
                double in = readData.toDouble();
                // qDebug()<< in;

                if(in>9)
                {
                    //qDebug()<<"nilai B"<<in;
                    inputlead2=in;
                }
            }
        }
    }
}

```

```

        // qDebug()<<inputlead2;
        for(int i=0;i<100;i++)
        {
            average_lead +=inputlead2;
            //qDebug()<<L2;
        }
        average_lead = average_lead/100;
        // qDebug()<<average_lead;
        if(average_lead>630&&pulse_high==false)
        {
            pulse_high=true;
            pulse_low=true;
            // qDebug()<<"pulse high";
        }
        if(average_lead<450&&pulse_low==true)
        {
            pulse_low=false;
            pulse_high=false;
            for(int i=0;i<1;i++)
            {
                a++;
                search++;
                // qDebug()<<search;
            }
        }
    }
}
else if (readData.contains("C"))
{
    readData = readData.replace("C","").trimmed();
    double in = readData.toDouble();
    // qDebug()<<in;

    if(in>9)
    {
        //qDebug()<<"nilai C"<<in;
        inputlead3=in;
    }
}
else if (readData.contains("D"))
{
    readData = readData.replace("D","").trimmed();
    double in = readData.toDouble();
    // qDebug()<<in;
    if(in>9)
    {
        //qDebug()<<"nilai D"<<in;
        inputPCG=in;
    }
}
double key = time_plot/160.0; // time elapsed since
start of demo, in seconds
//double input = 1;
static double lastPointKey = 0;

```

```

        if (key-lastPointKey > 0.002) // at most add point
every 2 ms
    {
        // plot new data -> remove old one
        //line

        if(key>keyswitch)
        {
            for(int i=0;i<1;i++)
            {
                changegraph++;
                keyswitch +=60 ;
                key1 +=60;
                key2 +=60;
                if(key2>=180&&changegraph==1)
                {
                    ui->customPlot->graph(1)->clearData();
                    ui->customPlot_2->graph(1)-
>clearData();
                    ui->customPlot_3->graph(1)-
>clearData();
                    ui->customPlot_4->graph(1)-
>clearData();
                }
            }
            if(changegraph>1)
            {
                changegraph=0;
                ui->customPlot->graph(0)->clearData();
                ui->customPlot_2->graph(0)->clearData();
                ui->customPlot_3->graph(0)->clearData();
                ui->customPlot_4->graph(0)->clearData();
            }
        }
        if(changegraph==0)
        {
            ui->customPlot->graph(0)->addData(key-key1,
inputlead1);
            // ui->customPlot->graph(0)-
>rescaleValueAxis();
            ui->customPlot->graph(1)-
>removeDataBefore(key-key2);

            ui->customPlot_2->graph(0)->addData(key-
key1, inputlead2);
            // ui->customPlot_2->graph(0)-
>rescaleValueAxis();
            ui->customPlot_2->graph(1)-
>removeDataBefore(key-key2);

            ui->customPlot_3->graph(0)->addData(key-
key1, inputlead3);
            // ui->customPlot_3->graph(0)-
>rescaleValueAxis();

```

```

        ui->customPlot_3->graph(1) -
>removeDataBefore(key-key2);

        ui->customPlot_4->graph(0)->addData(key-
key1, inputPCG);
        //    ui->customPlot_4->graph(0) -
>rescaleValueAxis();
        ui->customPlot_4->graph(1) -
>removeDataBefore(key-key2);
    }

    if(changegraph==1)
    {

        ui->customPlot->graph(0) -
>removeDataBefore(key-key1);
        ui->customPlot->graph(1)->addData(key-key2,
inputlead1);
        //    ui->customPlot->graph(1) -
>rescaleValueAxis();

        ui->customPlot_2->graph(0) -
>removeDataBefore(key-key1);
        ui->customPlot_2->graph(1)->addData(key-
key2, inputlead2);
        //    ui->customPlot_2->graph(1) -
>rescaleValueAxis();

        ui->customPlot_3->graph(0) -
>removeDataBefore(key-key1);
        ui->customPlot_3->graph(1)->addData(key-
key2, inputlead3);
        //    ui->customPlot_3->graph(1) -
>rescaleValueAxis();

        ui->customPlot_4->graph(0) -
>removeDataBefore(key-key1);
        ui->customPlot_4->graph(1)->addData(key-
key2, inputPCG);
        //    ui->customPlot_4->graph(1) -
>rescaleValueAxis();
    }
    //dot

    // ui->customPlot->graph(1)->addData(key,
inputlead1);
    //lastPointKey = key;

}

}
ui->customPlot->xAxis->setRange(0,0,Qt::AlignRight);
ui->customPlot->replot();

//customplot_2:

```

```

        //ui->customPlot_2->xAxis-
>setRange(key,8,Qt::AlignRight);
        ui->customPlot_2->replot();

        //customplot_3:
        //ui->customPlot_3->xAxis-
>setRange(key,8,Qt::AlignRight);
        ui->customPlot_3->replot();

        //customplot_4:
        //ui->customPlot_4->xAxis-
>setRange(key,8,Qt::AlignRight);
        ui->customPlot_4->replot();

        //
qDebug()<<inputlead1<<inputlead2<<inputlead3<<inputPCG<<endl;
        //qDebug()<<data_read.count();
        detik_realtime=0;
        serial="";

    }

}

/*fungsi untuk mengatur plotter*/
void MainWindow::myTimer()
{

    static QTime time(QTime::currentTime());
    // calculate two new data points:

    // make key axis range scroll with the data (at a constant
range size of 8):
    //customplot_1:

}

/*Setting tampilan grafik*/
void MainWindow::makePlot()
{

    //Untuk memberi warna pada plot
ui->customPlot->addGraph();
ui->customPlot->graph(0)->setPen(QPen(QColor(0,0,0)));
ui->customPlot->addGraph();
ui->customPlot->graph(1)->setPen(QPen(QColor(0,0,0)));

    //Untuk memberi range X dan Y
ui->customPlot->xAxis->setRange(0,60);
ui->customPlot->yAxis->setRange(0,1200);

```

```

    ui->customPlot->xAxis2->setVisible(true);
    ui->customPlot->xAxis2->setTickLabels(false);
    ui->customPlot->yAxis2->setVisible(true);
    ui->customPlot->yAxis2->setTickLabels(false);

    connect(ui->customPlot->xAxis,
    SIGNAL(rangeChanged(QCPRange)), ui->customPlot->xAxis2,
    SLOT(setRange(QCPRange)));
    connect(ui->customPlot->yAxis,
    SIGNAL(rangeChanged(QCPRange)), ui->customPlot->yAxis2,
    SLOT(setRange(QCPRange)));

    ui->customPlot->xAxis2->setBasePen(QPen(Qt::black, 1));
    ui->customPlot->yAxis2->setBasePen(QPen(Qt::black, 1));
    ui->customPlot->xAxis2->setTickPen(Qt::NoPen);
    ui->customPlot->yAxis2->setTickPen(Qt::NoPen);
    ui->customPlot->xAxis2->setSubTickPen(Qt::NoPen);
    ui->customPlot->yAxis2->setSubTickPen(Qt::NoPen);

    ui->customPlot->xAxis->setBasePen(QPen(Qt::black, 1));
    ui->customPlot->yAxis->setBasePen(QPen(Qt::black, 1));
    ui->customPlot->xAxis->setTickPen(Qt::NoPen);
    ui->customPlot->yAxis->setTickPen(Qt::NoPen);
    ui->customPlot->xAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot->yAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot->xAxis->setTickLabels(false);
    ui->customPlot->yAxis->setTickLabels(false);
    ui->customPlot->xAxis->grid()->setPen(QPen(Qt::NoPen));
    ui->customPlot->yAxis->grid()->setPen(QPen(Qt::NoPen));
    ui->customPlot->xAxis->grid()->setSubGridPen(Qt::NoPen);
    ui->customPlot->yAxis->grid()->setSubGridPen(Qt::NoPen);
    ui->customPlot->xAxis->grid()->setSubGridVisible(false);
    ui->customPlot->yAxis->grid()->setSubGridVisible(false);
    ui->customPlot->xAxis->grid()->setZeroLinePen(Qt::NoPen);
    ui->customPlot->yAxis->grid()->setZeroLinePen(Qt::NoPen);

    QPixmap pixmap;
    pixmap.load("/home/pi/ECG_Layout_fix.png");
    ui->customPlot-
>setBackground(pixmap.scaled(599, 115), Qt::KeepAspectRatioByExpanding);

    QLinearGradient plotGradient;
    plotGradient.setStart(0, 0);
    plotGradient.setFinalStop(0, 350);
    plotGradient.setColorAt(0, QColor(255, 255, 255));
    // plotGradient.setColorAt(1, QColor(50, 50, 50));
    ui->customPlot->setBackground(plotGradient);
}
void MainWindow::makePlot_2()
{

    //Untuk memberi warna pada plot
    ui->customPlot_2->addGraph();
    ui->customPlot_2->graph(0)->setPen(QPen(QColor(0, 0, 0)));
}

```

```

ui->customPlot_2->addGraph();
ui->customPlot_2->graph(1)->setPen(QPen(QColor(0,0,0)));

//Untuk memberi range X dan Y
ui->customPlot_2->xAxis->setRange(0,60);
ui->customPlot_2->yAxis->setRange(0,1159);
ui->customPlot_2->xAxis2->setVisible(true);
ui->customPlot_2->xAxis2->setTickLabels(false);
ui->customPlot_2->yAxis2->setVisible(true);
ui->customPlot_2->yAxis2->setTickLabels(false);

connect(ui->customPlot_2->xAxis,
SIGNAL(rangeChanged(QCPRange)),ui->customPlot_2->xAxis2,
SLOT(setRange(QCPRange)));
connect(ui->customPlot_2->yAxis,
SIGNAL(rangeChanged(QCPRange)),ui->customPlot_2->yAxis2,
SLOT(setRange(QCPRange)));

ui->customPlot_2->xAxis2->setBasePen(QPen(Qt::black,1));
ui->customPlot_2->yAxis2->setBasePen(QPen(Qt::black,1));
ui->customPlot_2->xAxis2->setTickPen(Qt::NoPen);
ui->customPlot_2->yAxis2->setTickPen(Qt::NoPen);
ui->customPlot_2->xAxis2->setSubTickPen(Qt::NoPen);
ui->customPlot_2->yAxis2->setSubTickPen(Qt::NoPen);

ui->customPlot_2->xAxis->setBasePen(QPen(Qt::black,1));
ui->customPlot_2->yAxis->setBasePen(QPen(Qt::black,1));
ui->customPlot_2->xAxis->setTickPen(Qt::NoPen);
ui->customPlot_2->yAxis->setTickPen(Qt::NoPen);
ui->customPlot_2->xAxis->setSubTickPen(Qt::NoPen);
ui->customPlot_2->yAxis->setSubTickPen(Qt::NoPen);
ui->customPlot_2->xAxis->setTickLabels(false);
ui->customPlot_2->yAxis->setTickLabels(false);
ui->customPlot_2->xAxis->grid()->setPen(QPen(Qt::NoPen));
ui->customPlot_2->yAxis->grid()->setPen(QPen(Qt::NoPen));
ui->customPlot_2->xAxis->grid()->setSubGridPen(Qt::NoPen);
ui->customPlot_2->yAxis->grid()->setSubGridPen(Qt::NoPen);
ui->customPlot_2->xAxis->grid()->setSubGridVisible(false);
ui->customPlot_2->yAxis->grid()->setSubGridVisible(false);
ui->customPlot_2->xAxis->grid()->setZeroLinePen(Qt::NoPen);
ui->customPlot_2->yAxis->grid()->setZeroLinePen(Qt::NoPen);

QPixmap pixmap;
pixmap.load("/home/pi/ECG_Layout_fix.png");
ui->customPlot_2-
>setBackground(pixmap.scaled(599,115),Qt::KeepAspectRatioByExpanding);

QLinearGradient plotGradient;
plotGradient.setStart(0,0);
plotGradient.setFinalStop(0,350);
plotGradient.setColorAt(0,QColor(255,255,255));
//plotGradient.setColorAt(1,QColor(50,50,50));
ui->customPlot_2->setBackground(plotGradient);
}
void MainWindow::makePlot_3()

```



```

{
    //Untuk memberi warna pada plot
    ui->customPlot_3->addGraph();
    ui->customPlot_3->graph(0)->setPen(QPen(QColor(0,0,0)));
    ui->customPlot_3->addGraph();
    ui->customPlot_3->graph(1)->setPen(QPen(QColor(0,0,0)));

    //Untuk memberi range X dan Y
    ui->customPlot_3->xAxis->setRange(0,60);
    ui->customPlot_3->yAxis->setRange(0,1090);
    ui->customPlot_3->xAxis2->setVisible(true);
    ui->customPlot_3->xAxis2->setTickLabels(false);
    ui->customPlot_3->yAxis2->setVisible(true);
    ui->customPlot_3->yAxis2->setTickLabels(false);

    connect(ui->customPlot_3->xAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_3->xAxis2,
    SLOT(setRange(QCPRange)));
    connect(ui->customPlot_3->yAxis,
    SIGNAL(rangeChanged(QCPRange)),ui->customPlot_3->yAxis2,
    SLOT(setRange(QCPRange)));

    ui->customPlot_3->xAxis2->setBasePen(QPen(Qt::black,1));
    ui->customPlot_3->yAxis2->setBasePen(QPen(Qt::black,1));
    ui->customPlot_3->xAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_3->yAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_3->xAxis2->setSubTickPen(Qt::NoPen);
    ui->customPlot_3->yAxis2->setSubTickPen(Qt::NoPen);

    ui->customPlot_3->xAxis->setBasePen(QPen(Qt::black,1));
    ui->customPlot_3->yAxis->setBasePen(QPen(Qt::black,1));
    ui->customPlot_3->xAxis->setTickPen(Qt::NoPen);
    ui->customPlot_3->yAxis->setTickPen(Qt::NoPen);
    ui->customPlot_3->xAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_3->yAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_3->xAxis->setTickLabels(false);
    ui->customPlot_3->yAxis->setTickLabels(false);
    ui->customPlot_3->xAxis->grid()->setPen(QPen(Qt::NoPen));
    ui->customPlot_3->yAxis->grid()->setPen(QPen(Qt::NoPen));
    ui->customPlot_3->xAxis->grid()->setSubGridPen(Qt::NoPen);
    ui->customPlot_3->yAxis->grid()->setSubGridPen(Qt::NoPen);
    ui->customPlot_3->xAxis->grid()->setSubGridVisible(false);
    ui->customPlot_3->yAxis->grid()->setSubGridVisible(false);
    ui->customPlot_3->xAxis->grid()->setZeroLinePen(Qt::NoPen);
    ui->customPlot_3->yAxis->grid()->setZeroLinePen(Qt::NoPen);

    QPixmap pixmap;
    pixmap.load("/home/pi/ECG_Layout_fix.png");
    ui->customPlot_3-
>setBackground(pixmap.scaled(599,115),Qt::KeepAspectRatioByExpanding);

    QLinearGradient plotGradient;
    plotGradient.setStart(0,0);
    plotGradient.setFinalStop(0,350);
    plotGradient.setColorAt(0,QColor(255,255,255));

```

```

        //plotGradient.setColorAt(1, QColor(50, 50, 50));
        ui->customPlot_3->setBackground(plotGradient);
    }
void MainWindow::makePlot_4()
{
    //Untuk memberi warna pada plot
    ui->customPlot_4->addGraph();
    ui->customPlot_4->graph(0)->setPen(QPen(QColor(88, 255,
85)));
    ui->customPlot_4->addGraph();
    ui->customPlot_4->graph(1)->setPen(QPen(QColor(88, 225,
85)));

    //Untuk memberi range X dan Y
    ui->customPlot_4->xAxis->setRange(0, 60);
    ui->customPlot_4->yAxis->setRange(0, 1200);
    ui->customPlot_4->xAxis2->setVisible(true);
    ui->customPlot_4->xAxis2->setTickLabels(false);
    ui->customPlot_4->yAxis2->setVisible(true);
    ui->customPlot_4->yAxis2->setTickLabels(false);

    connect(ui->customPlot_4->xAxis,
    SIGNAL(rangeChanged(QCPRange)), ui->customPlot_4->xAxis2,
    SLOT(setRange(QCPRange)));
    connect(ui->customPlot_4->yAxis,
    SIGNAL(rangeChanged(QCPRange)), ui->customPlot_4->yAxis2,
    SLOT(setRange(QCPRange)));

    ui->customPlot_4->xAxis2->setBasePen(QPen(Qt::white, 1));
    ui->customPlot_4->yAxis2->setBasePen(QPen(Qt::white, 1));
    ui->customPlot_4->xAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_4->yAxis2->setTickPen(Qt::NoPen);
    ui->customPlot_4->xAxis2->setSubTickPen(Qt::NoPen);
    ui->customPlot_4->yAxis2->setSubTickPen(Qt::NoPen);

    ui->customPlot_4->xAxis->setBasePen(QPen(Qt::white, 1));
    ui->customPlot_4->yAxis->setBasePen(QPen(Qt::white, 1));
    ui->customPlot_4->xAxis->setTickPen(Qt::NoPen);
    ui->customPlot_4->yAxis->setTickPen(Qt::NoPen);
    ui->customPlot_4->xAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_4->yAxis->setSubTickPen(Qt::NoPen);
    ui->customPlot_4->xAxis->setTickLabels(false);
    ui->customPlot_4->yAxis->setTickLabels(false);
    ui->customPlot_4->xAxis->grid()->setPen(QPen(Qt::NoPen));
    ui->customPlot_4->yAxis->grid()->setPen(QPen(Qt::NoPen));
    ui->customPlot_4->xAxis->grid()->setSubGridPen(Qt::NoPen);
    ui->customPlot_4->yAxis->grid()->setSubGridPen(Qt::NoPen);
    ui->customPlot_4->xAxis->grid()->setSubGridVisible(false);
    ui->customPlot_4->yAxis->grid()->setSubGridVisible(false);
    ui->customPlot_4->xAxis->grid()->setZeroLinePen(Qt::NoPen);
    ui->customPlot_4->yAxis->grid()->setZeroLinePen(Qt::NoPen);

    QLinearGradient plotGradient;
    plotGradient.setStart(0, 0);
    plotGradient.setFinalStop(0, 350);
    plotGradient.setColorAt(0, QColor(80, 80, 80));

```

```

        plotGradient.setColorAt(1, QColor(50, 50, 50));
        ui->customPlot_4->setBackground(plotGradient);
    }

void MainWindow::on_pushButton_2_clicked()
{
    patientdatasave->show();
}

void MainWindow::on_pushButton_clicked()
{
    openfile->show();
}

void MainWindow::on_pushButton_3_clicked()
{
    dw->show();
}

void MainWindow::on_pushButton_4_clicked()
{
    QString file_Name;
    QString istring;
    for(int i=0;i<1;i++)
    {
        second_number_file_realtime++;
        number_file_second_realtime =
        QString::number(second_number_file_realtime);

    }

    for(int i=0;i<4;i++)
    {
        istring = QString::number(i+1);
        if(i<3)
        {

file_Name="/home/pi/Datagambar/lead"+istring+"("+number_file_sec
ond_realtime+").png";
        }
        else
        {

file_Name="/home/pi/Datagambar/Pcg("+number_file_second_realtime
+").png";
        }
        QFile file_png(file_Name);

        if (!file_png.open(QIODevice::WriteOnly))
        {
            qDebug() << file_png.errorString();
        }
    }
}

```

```

    }
    else if(istring=="1")
    {
        ui->customPlot->savePng(file_Name);
    }
    else if(istring=="2")
    {
        ui->customPlot_2->savePng(file_Name);
    }
    else if(istring=="3")
    {
        ui->customPlot_3->savePng(file_Name);
    }
    else
    {
        ui->customPlot_4->savePng(file_Name);
    }
}

void MainWindow::on_pushButton_5_clicked()
{
    for(int i =0;i<1;i++)
    {
        nomorgambar++;
        QFile
file_ss("/home/pi/Screenshot/Screenshot"+QString::number(nomorgambar)+".jpeg");

        if (!file_ss.open(QIODevice::WriteOnly))
        {
            qDebug() << file_ss.errorString();
        }
        else
        {
            QPixmap pixmap = QPixmap::grabWidget(this);
            QImage image = pixmap.toImage();

            image.save("/home/pi/Screenshot/Screenshot"+QString::number(nomorgambar)+".jpeg");
        }
    }
}

```

Gambar Pembuatan Alat

Gambar Penyolderan Rangkaian PCG



Gambar Pengeboran Rangkaian PCG



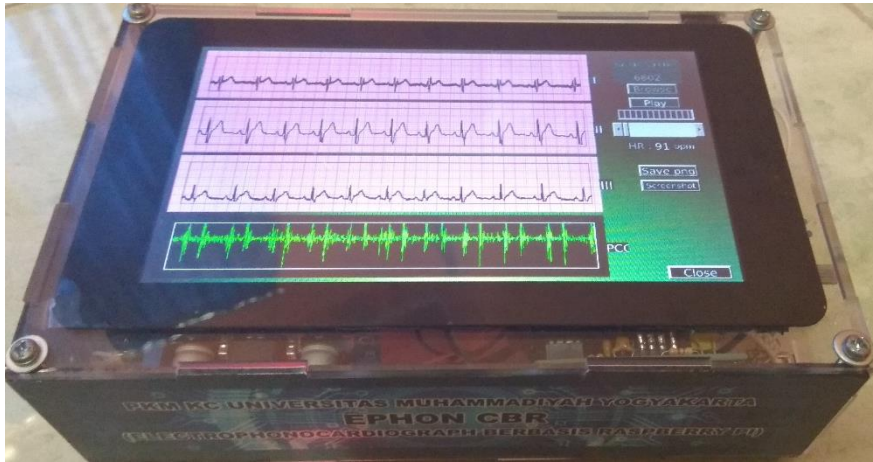
Gambar Penggabungan Rangkaian PCG



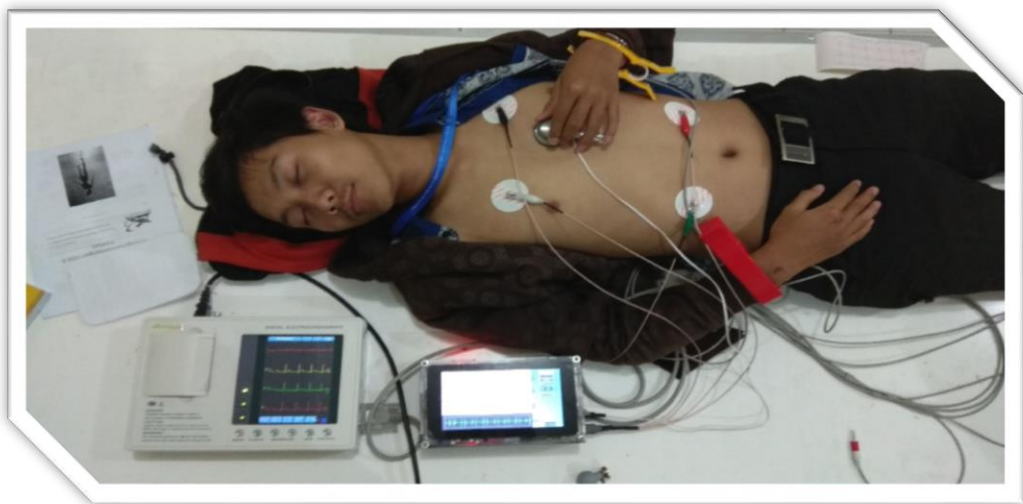
Gambar Penggabungan PCG dengan ECG



Gambar Alat Sudah Jadi



Gambar Pengambilan Data



Gambar Hasil Grafik



Dokumentasi Publikasi



TribunJogja.com Rabu, 11 Juli 2018 Network Login

Home News DIY Solo Jawa Bisnis Super Ball Sport Seleb Hotline Tribun Jogja TV Lainnya

Home » News » Pendidikan

Pendidikan

Gunakan Teknologi ECG dan PCG, Mahasiswa UMY Ciptakan Alat Diagnosa Jantung Portabel

Jumat, 6 Juli 2018 19:35





TRADE BITCOINS SEKETIKA



KEUNTUNGAN LEBIH DENGAN LEVERAGE

MULAI TRADING



UMM

UNIVERSITAS MUHAMMADIYAH MALANG
Indonesia English Arabic

| | | | |
|---|--|---|---|
| Penerimaan Mahasiswa Baru <ul style="list-style-type: none"> Info Pendaftaran Program Internasional | Profil <ul style="list-style-type: none"> Profil Singkat Pimpinan UMM Tentang UMM | Pendidikan <ul style="list-style-type: none"> Studi di UMM Proses Pendidikan Kartu Hasil Studi | Mahasiswa dan Alumni <ul style="list-style-type: none"> Administrasi Mahasiswa Pembinaan Mahasiswa Fasilitas Mahasiswa |
|---|--|---|---|

/ [Berita Muhammadiyah](#)

Membanggakan, Mahasiswa UMY Ini Berhasil Ciptakan Alat Diagnosa Jantung Portabel dengan Teknologi ECG dan PCG

Author : Pimpinan Pusat Muhammadiyah | Sabtu, 07 Juli 2018 09:55 WIB

MUHAMMADIYAH.OR.ID, BANTUL - *Cardiovascular diseases* atau lebih umum dikenal dengan penyakit jantung merupakan gangguan kesehatan yang sangat mungkin dialami oleh setiap orang. Data World Health Organization (WHO) tahun 2012 menunjukkan 17,5 juta orang di dunia meninggal akibat penyakit kardiovaskuler atau 31 persen dari 56,5 juta kematian di seluruh dunia. Pada kasus yang terjadi di Indonesia berdasarkan data yang ditunjukkan oleh Kemenkes (Kementerian Kesehatan) Republik Indonesia tahun 2017, penyakit jantung menjadi penyebab kematian tertinggi pada semua umur untuk penyakit tidak menular. Penyakit jantung ini dapat dihindari dengan menjaga kesehatan dengan baik, salah satunya dengan melakukan pengecekan berkala terhadap aktivitas jantung dengan menggunakan alat diagnosa berupa *electrocardiograph (ECG)* dan *phonocardiograph (PCG)*.

Berita Muhammadiyah

Arsip Berita

- [2018 \[972\]](#)
- [2017 \[2606\]](#)
- [2016 \[1927\]](#)
- [2015 \[422\]](#)
- [2014 \[313\]](#)
- [2013 \[451\]](#)
- [2012 \[573\]](#)

DILENGKAPI ECG DAN PCG Mahasiswa UMY Ciptakan Alat Diagnosis Jantung Portabel



Kelompok PKM-KC UMY dengan karya inovasinya.

YOGYA (KR) - Penyakit jantung atau *Cardiovascular diseases* merupakan salah satu gangguan kesehatan cukup serius. Organisasi Kesehatan Dunia atau *World Health Organization* (WHO) mencatat, angka kasusnya bertambah, seiring gaya hidup tak sehat masyarakat global.

Data di Kementerian Kesehatan (Kemenkes) pada 2017 menunjukkan, penyakit jantung menjadi penyebab kematian tertinggi pada semua umur untuk penyakit tidak menular. Karena itu, deteksi dini, serta peningkatan kualitas layanan kesehatan harus terus dikembangkan.

Apalagi, penyakit jantung ini dapat dihindari dengan menjaga kesehatan dengan baik. Salah satunya dengan melakukan pengecekan berkala terhadap aktivitas jantung dengan menggunakan alat diagnosa berupa *electrocardiograph* (ECG) dan *phonocardiograph* (PCG).

Untuk memenuhi kebutuhan penunjang kesehatan tersebut, Kelompok PKM-KC (Program Kreativitas Mahasiswa - Karya Cipta) Univer-

sitas Muhammadiyah Yogyakarta (UMY) merancang sebuah alat diagnosa jantung portabel yang memadukan kedua teknologi ECG dan PCG.

"ECG bekerja dengan mendiagnosa aktivitas elektrik dari tubuh untuk memeriksa kesehatan fungsi jantung," papar Dede Widianto, salah satu anggota Kelompok PKM-KC UMY di Kampus Terpadu UMY Tamantirto Kasihan, Jumat (6/7).

Sedangkan PCG merupakan stetoskop elektronik yang fungsinya hampir sama dengan stetoskop biasa, yaitu teknik pemeriksaan dengan cara mendengarkan bunyi akibat getaran aktivitas jantung dan gangguan pada aliran darah dalam jantung.

Dede menjelaskan, desain alat yang compact, portable, dan sederhana membuat alat ini mudah digunakan. Alat ini diklaim dapat menjadi solusi bagi petugas medis yang bekerja di daerah tertinggal, perbatasan dan kepulauan untuk membantu pelayanan kesehatan menjadi lebih

KR Agus Sukoto