

## BAB IV PEMBAHASAN DAN ANALISIS

### 4.1 Analisis Perancangan

Perancangan alat terbagi menjadi dua perangkat keras (*Hardware*) dan perangkat lunak (*Software*). Perancangan perangkat keras terdiri dari pengujian setiap individu komponen dan pembuatan alat keseluruhan. Sedangkan untuk perangkat lunak terdiri dari pembuatan *Graphical User Interface* (GUI) dan program pengolahan citra.

#### 4.1.1 Hasil Pengujian Perangkat Keras (*Hardware*)

Pengujian perangkat keras dilakukan terhadap 3 komponen utama diantaranya adalah Kamera NoIR 5 MP, LCD TFT 3.5" *Touch Screen*, dan Inframerah LED. Adapun hasil pengujian yang diperoleh adalah sebagai berikut;

##### 1. Kamera NoIR

Pengujian dilakukan dengan kondisi kamera sudah diaktifkan melalui konfigurasi di raspberry pi 3. Perintah yang digunakan adalah

```
raspistill -h 480 -w 320 -o /home/pi/test_image.png
```

Setelah perintah dijalankan melalui terminal maka jika berhasil menangkap gambar menunjukkan kamera berfungsi dan file hasil tangkapan kamera tersimpan otomatis. Gambar 4.1 merupakan hasil gambar dari tangkapan kamera.

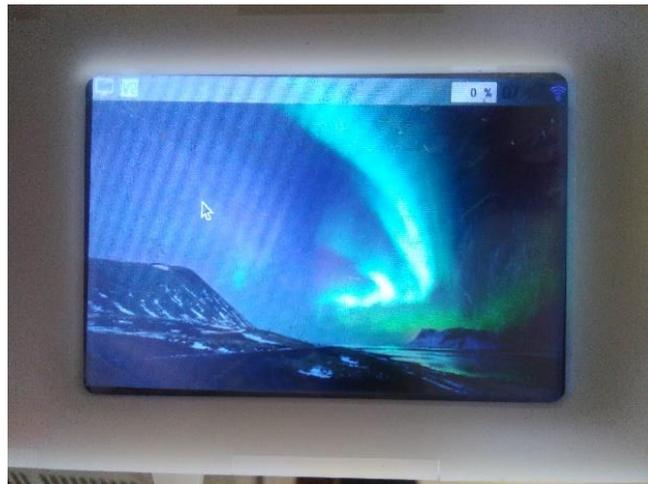


**Gambar.4.1** Gambar hasil tangkapan kamera.

Dari perintah yang dituliskan di terminal, terdapat parameter *-h* yang berarti pengaturan terhadap *height* (tinggi) gambar dan *-w* sebagai pengaturan *width* (lebar) gambar. Kedua parameter tersebut memiliki satuan pixel. Melalui gambar 4.1 maka, dapat disimpulkan bahwa kamera dapat bekerja dengan baik dan bisa digunakan.

## 2. LCD TFT 3.5" Touch Screen

Pengujian terhadap LCD ini dilakukan cukup dengan menuliskan perintah `./LCD35-show` pada terminal dengan direktori dimana file konfigurasi `LCD35-show` berada. Setelah perintah tersebut dijalankan maka otomatis raspberry akan *reboot* otomatis dan kemudian tampilan berpindah ke LCD TFT 3.5" Touch Screen. Gambar 4.2 menunjukkan pada saat tampilan berhasil berpindah di LCD TFT 3.5" Touch Screen.



**Gambar.4.2** Tampilan desktop pada layar LCD TFT 3.5".

## 3. Inframerah LED

Inframerah LED secara otomatis akan menyala pada saat alat dihidupkan, sehingga dapat langsung digunakan. Adapun tanda dari IR LED aktif adalah tampak cahaya cerah berwarna keunguan seperti pada gambar 4.3. Apabila cahaya tersebut tidak tampak ataupun tidak menyala menandakan koneksi daya tidak terpasang atau

kendur. Sinar yang dihasilkan IR LED ini memberikan pengaruh terhadap gambar yang ditangkap oleh kamera.



**Gambar.4.3** Inframerah LED yang telah aktif.

Pengujian IR LED dilakukan juga terhadap tegangan suplainya baik tegangan suplai untuk modul IR LED dan tegangan pada IR LED itu sendiri. Berikut hasil pengukuran yang diperoleh.

**Tabel.4.1** Data pengukuran tegangan modul IR

No	Tegangan Modul
1	3.06 Volt

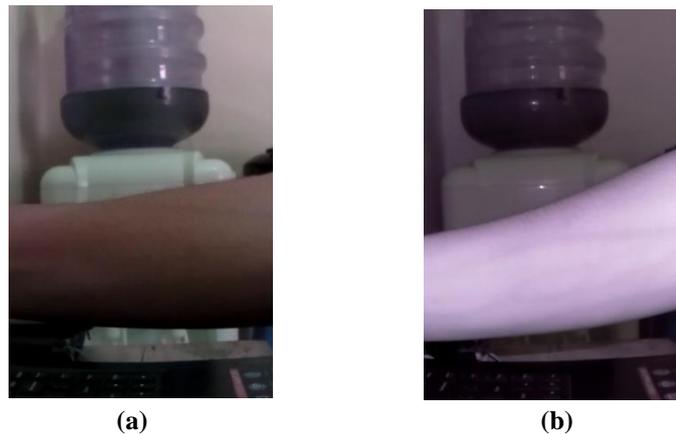
Dari tabel 4.1 tegangan modul adalah sebesar 3.06 volt. Tegangan tersebut tidak mendekati nilai 3.3 volt yaitu tegangan suplai dari Raaspberry. Hal ini disebabkan adanya (*drop voltage*) tegangan jatuh. Terjadinya hal ini disebabkan daya yang berasal dari Raspberry Pi tidak hanya digunakan modul IR LED saja, tetapi digunakan juga oleh modul kamera.

Adapun pengujian tegangan pada IR LED terbagi menjadi 2 kondisi, yaitu saat kondisi LDR tertutup dan terbuka.

**Tabel.4.2** Data pengukuran tegangan di IR LED.

No	Tegangan IR LED	
	LDR tertutup	LDR terbuka
1	1.41 Volt	0.04 Volt
2	Cahaya Terang	Cahaya Redup

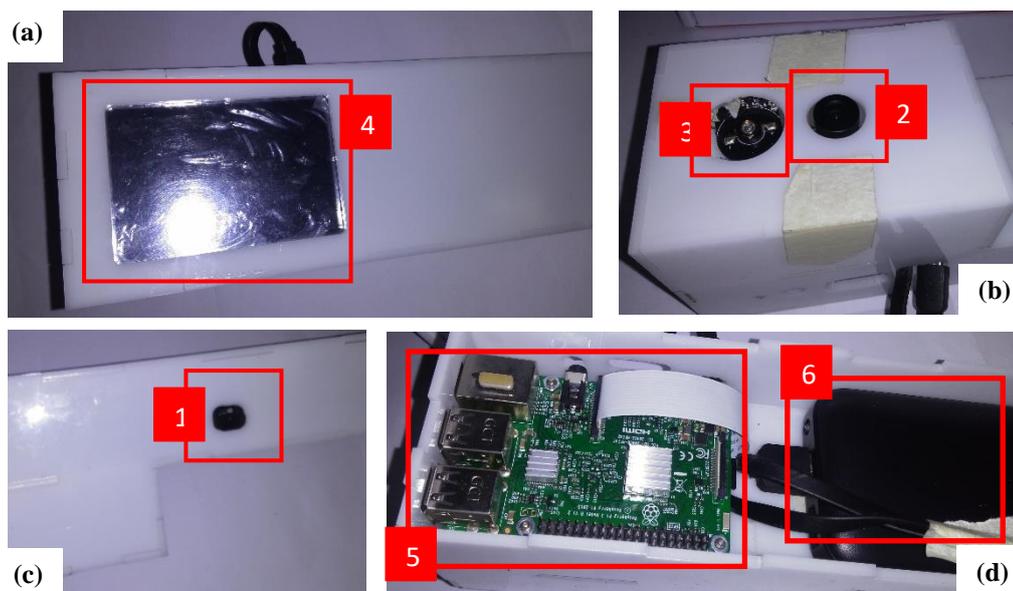
Dari data tabel 4.2, diketahui bahwa dengan kondisi LDR tertutup dapat memberikan cahaya yang terang dan radiasi infrared yang baik. Oleh karena itu, LDR pada modul ditutup menggunakan lakban hitam untuk menghindari cahaya masuk ke LDR. Ditutupnya LDR mampu menjadikan IR LED dapat memancarkan sinarnya secara maksimal dan akan selalu stabil dalam setiap kondisi. Berikut hasil perbedaan tangkapan gambar dengan dan tidaknya sinar IR LED.



**Gambar.4.4** (a). Citra dengan IR LED tidak aktif dan (b) citra dengan IR LED aktif.

#### 4. Hasil Perancangan Perangkat Keras (*Hardware*) Keseluruhan

Adapun hasil pemasangan keseluruhan alat tampak seperti pada gambar 4.5.



**Gambar.4.5** (a) Alat tampak bagian atas, (b) bagian bawah, (c) bagian samping (d) bagian dalam.

Berikut penjelasan singkat dari bagian alat yang tampak pada gambar 4.5.

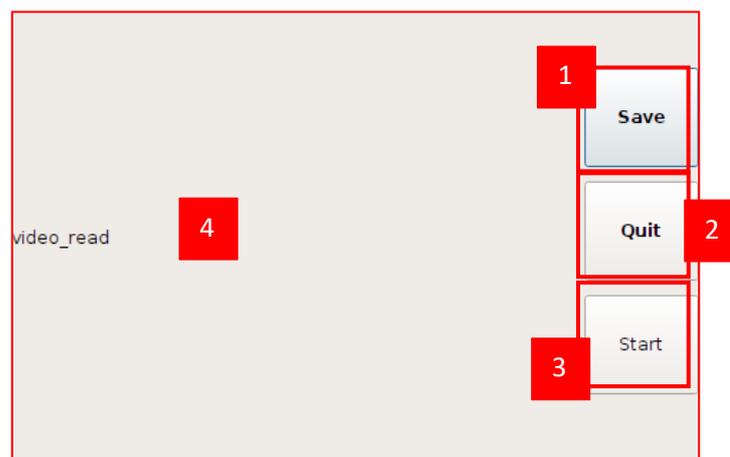
**Tabel.4.3** Penjelasan singkat bagian alat dan fungsinya.

No	Komponen/Bagian	Fungsi
1	Tombol power	Menhidupkan dan mematikan alat
2	Kamera NoIR 5 MP	Sensor yang menangkap citra hingga rentang cahaya inframerah
3	Inframerah LED	Sumber cahaya inframerah dan dipancarkan kearah obyek
4	TFT LCD 3.5"	Menampilkan hasil citra yang ditangkap oleh kamera.
5	Raspberry Pi 3	<i>Singe Board Computer</i> yang digunakan untuk mengolah citra
6	Baterai ( <i>power bank</i> )	Sumber daya untuk seluruh komponen yang digunakan pada alat.

#### 4.1.2 Hasil Perancangan Perangkat Lunak (*Software*)

##### 1. Hasil Desain GUI

Hasil dari perancangan desai GUI alat tampak seperti gambar 4.6 berikut penjelasan setiap bagiannya.



**Gambar.4.6** Desain *Graphical User Interface* (GUI)

Tabel.4.4 Penjelasan singkat bagian GUI dan fungsinya.

No	Komponen	Keterangan
1	<i>Save button</i>	Berfungsi untuk menyimpan citra yang tampil pada display no 4. Data yang tersimpan dalam bentuk format waktu tanggal dan jam.
2	<i>Quit button</i>	Berfungsi untuk keluar dari display dan menonaktifkan alat jika alat sudah tidak digunakan.
3	<i>Start/Stop button</i>	Berfungsi untuk memulai mereka citra sekaligus mengolahnya dan memberhentikan proses pengolahan citra
4	<i>Display</i>	Berfungsi untuk menampilkan hasil citra yang telah diproses.

## 2. Hasil Perancangan Program Perangkat Lunak

Proses pembuatan program sangat didukung dengan adanya diagram alir (*flow chart*) seperti tampak pada gambar 4.7.

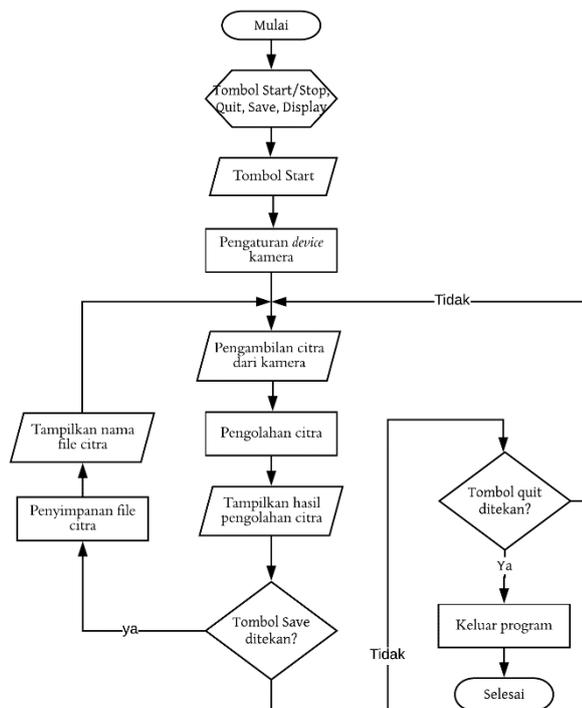
Gambar.4.7 Diagram alir program perangkat lunak (*software*)

Diagram 4.7 ini memberikan gambaran bagaimana program di dalam alat itu berjalan dan berfungsi. Perancangan perangkat lunak (*software*) sendiri, terdiri dari pembuatan GUI (*Graphical User Interface*) dan pengolahan citra yang dibuat secara terpisah. Kemudian diakhir perancangan, keduanya digabungkan sehingga dapat saling berintegrasi.

Berikut ini pemaparan mengenai program GUI dan pengolahan citra yang telah dibuat. Pada gambar 4.8 menunjukkan keseluruhan pustaka yang digunakan. Pada baris pertama dan ke-2 merupakan perintah untuk memanggil pustaka pendukung perangkat keras kamera. Kamera Raspberry Pi tidak dapat berjalan dan berfungsi tanpa adanya pustaka ini.

```

1  from picamera.array import PiRGBArray
2  from picamera import PiCamera
3  import cv2 as cv
4  import sys, os
5  from time import sleep
6  from datetime import datetime
7  from PyQt5.QtCore import QTimer, Qt
8  from PyQt5.QtGui import QImage, QPixmap
9  from PyQt5.QtWidgets import QApplication, QDialog
10 from PyQt5.uic import loadUi

```

**Gambar.4.8** Daftar pustaka yang digunakan di dalam program.

Adapun untuk baris 3 merupakan perintah untuk memanggil pustaka open cv yaitu pustaka untuk membantu dalam pengolahan citra. Baris ke-4 hingga baris ke-6 merupakan perintah untuk memanggil pustaka yang berkaitan dengan sistem, dalam hal ini adalah pustaka pewaktuian, tanggal dan hal yang berkaitan dengan sistem aplikasi. Kemudian untuk baris ke-7 hingga akhir merupakan baris perintah untuk memanggil pustaka pendukung pembuatan GUI.

Konstruksi program ini adalah bentuk dari pemrograman berorientasi obyek atau yang dikenal dengan OOP (*Object Oriented Programming*) sehingga digunakan kelas dan beberapa fungsi. Bentuk kelas yang digunakan hanya satu yaitu *veinviewer*. Kelas ini memiliki banyak fungsi untuk mendukung pada saat pembuatan obyek dalam fungsi utama.

```

class veinviewer(QDialog):

```

**Gambar.4.9** Bagian program yang mendefinisikan kelas *veinviewer*.

Di dalam kelas ini, terdapat satu buah fungsi yang disebut dengan fungsi inisialisasi. Fungsi ini akan dieksekusi pertama kali ketika pembuatan obyek kelas *veinviewer*. Berbeda dengan fungsi lainnya, fungsi ini tidak perlu dipanggil untuk dapat mengeksekusi program yang ada di dalamnya. Oleh sebab itu, pemanggilan file atau modul lain yang akan digunakan perlu didefinisikan lebih awal. Pada perancangan perangkat lunak, desain GUI sendiri dibuat terpisah dengan

```
def __init__(self):
    super(veinviewer, self).__init__()
    loadUi('designUIraspi.ui', self)
```

**Gambar.4.10** Bagian fungsi inisialisasi di dalam kelas *veinviewer*.

menggunakan *software* Qt Designer dalam format ekstensi file *.ui*, maka dari itu file desain tersebut harus diinisialisasi dan dipanggil lebih awal. File desain GUI tersebut bernama *designUIraspi.ui*. Jika diamati tampak pada gambar 4.10 bagian baris terakhir merupakan perintah yang digunakan untuk memanggil sekaligus mengintegrasikan program python dengan desain GUI yang telah dibuat.

Selain file ataupun modul, variabel global yang nantinya akan digunakan di beberapa fungsi didefinisikan dalam fungsi ini. Variabel ini hanya dapat digunakan dalam kelas *veinviewer* dan tidak dapat digunakan dalam kelas lainnya. Berikut beberapa variabel yang didefinisikan dalam fungsi inisialisasi ini.

```
self.cam_H = 480
self.cam_W = 320
self.rgb_result = None
self.start = 0
```

**Gambar.4.11** Variabel global yang didefinisikan dalam bagian fungsi inisialisasi.

Dalam fungsi ini didefinisikan juga eksekusi yang harus dilakukan pada saat tombol di dalam GUI ditekan/disentuh. Perintah dalam tombol ini memanggil fungsi lainnya untuk dieksekusi. Adapun tombolnya sendiri tidak didefinisikan di awal karena sudah termasuk di dalam file desain GUI jadi tidak perlu didefinisikan lagi.

```
self.save_button.clicked.connect(self.save)
self.quit_button.clicked.connect(self.quit)
self.start_button.clicked.connect(self.start_video)
```

**Gambar.4.12** Perintah untuk memfungsikan tombol.

```
def quit(self):
    sys.exit()
```

**Gambar.4.13** Fungsi quit.

Fungsi lain dalam kelas veinviewer ini yaitu fungsi quit. Fungsi ini dipanggil untuk dieksekusi pada saat tombol quit ditekan. Perintah di dalam fungsi ini menggunakan pustaka sys jadi tanpa menggunakan pustaka tersebut fungsi ini tidak akan bisa berjalan. Perintah dalam fungsi ini akan mengakhiri keseluruhan sistem.

Selain tombol quit, terdapat tombol start yang digunakan untuk memulai perekaman video dan menampilkannya secara *real-time*. Tombol start ini memanggil fungsi start\_video untuk dijalankan dan kemudian akan memulai perekaman obyek. Namun sebelum obyek direkam, fungsi start\_video akan memanggil fungsi lain yang digunakan sebagai pengaturan kamera. Fungsi tersebut yaitu fungsi setup\_camera. Fungsi yang dipanggil ini berisikan pengaturan terhadap

```
def start_video(self):
    if self.start == 0 or self.start == 2:
        if self.start == 0:
            self.start_button.setText('Stop')
            self.image = None
            self.camera = PiCamera()
            self.setup_camera()
            self.rawCapture = PiRGBArray(self.camera, size=(self.cam_H, self.cam_W))
            self.start = 1
            self.timer = QTimer(self)
            self.timer.timeout.connect(self.update_frame)
            self.timer.start(0.5)
        if self.start == 2:
            self.start_button.setText('Stop')
            self.timer.start(0.5)
            self.start = 1
    elif self.start == 1:
        self.start_button.setText('Start')
        self.timer.stop()
        self.start = 2
```

**Gambar.4.14** Fungsi start\_video.

beberapa parameter kamera saat perekaman video seperti kontras, resolusi citra, sharpness, kecerahan dan *frame rate* video.

Setelah pengaturan kamera selesai, maka proses selanjutnya program akan mengeksekusi perintah inisialisasi komponen pewaktu dan mengaktifkannya. Komponen pewaktu ini yaitu QTimer. Pustaka ini akan mengulang eksekusi program secara terus menerus sesuai dengan waktu yang diatur dalam satuan mili detik dan digunakan untuk memulai dan menghentikan perekaman obyek. Apabila

timer ini sudah aktif tepatnya setelah tombol start ditekan, maka timer akan memanggil fungsi untuk pengambilan gambar sekaligus pengolah citra yaitu fungsi `update_frame`.

```
def setup_camera(self):
    self.camera.resolution = (self.cam_H, self.cam_W)
    self.camera.framerate = 32
    self.camera.brightness = 40
    self.camera.contrast = 90
    self.camera.rotation = 270
    self.camera.sharpness = 50
    sleep(0.1)
```

**Gambar.4.15** Fungsi `setup_camera`.

Dalam fungsi `update_frame` ini terdapat beberapa perintah yang terbagi menjadi 3 yaitu perintah untuk mengambil citra asli dari kamera seperti pada gambar 4.16, bagian pengolahan citra, dan penampilan hasil citra. Sebelum kamera diperintahkan untuk menangkap obyek, data dalam *buffer* yang akan digunakan untuk menampung data citra dibersihkan dan dihapus terlebih dahulu. Perintah untuk membersihkannya terlihat sebagaimana di baris pertama pada gambar 4.16.

```
self.rawCapture.truncate(0)
self.camera.capture(self.rawCapture, format="bgr", use_video_port=True)
self.image = self.rawCapture.array
```

**Gambar.4.16** Perintah pengambilan citra dengan kamera.

Setelah data *buffer* siap menampung data citra, maka mulanya kamera menangkap citra obyek dalam format BGR. Citra BGR merupakan citra yang sama seperti RGB hanya berbeda urutan kanal warna. Pada baris kedua di gambar 4.16 merupakan perintah untuk menangkap citra obyek. Citra obyek yang ditangkap masih berbentuk *data raw* dalam bentuk array. Data ini selanjutnya diinisialisasi sebagai variabel `image`.

Setelah citra diperoleh dari kamera, maka selanjutnya citra diolah dengan metode pengolahan citra. Metode pengolahan citra ini menggunakan perintah-perintah yang didukung pustaka `open cv`. Setelah proses pengolahan citra selesai, citra tersebut dimasukkan ke dalam fungsi lain yaitu `display_image` untuk ditampilkan pada GUI. Fungsi tersebut dieksekusi dengan memanggilnya

```
self.displayImage(self.rgb_result, 1)
```

**Gambar.4.17** Perintah untuk memanggil fungsi `display_image`.

sebagaimana gambar 4.17. Variabel `rgb_result` dalam perintah tersebut merupakan citra yang telah diproses dan siap untuk ditampilkan.

```
def displayImage(self, img, window = 1):
    qformat = QImage.Format_RGB888
    if len(img.shape)==3:
        if img.shape[2]==4:
            qformat=QImage.Format_RGBA8888
        else:
            qformat=QImage.Format_RGB888

    outImage = QImage(img, img.shape[1], img.shape[0], img.strides[0], qformat)
    outImage = outImage.rgbSwapped()
```

**Gambar.4.18** Perintah untuk mengubah format citra sesuai dengan format display QLabel.

Adapun fungsi dari `display_image` itu sendiri terbagi menjadi 2 bagian yaitu, bagian untuk mendeteksi jenis dan format citra yang akan ditampilkan dan bagian untuk menampilkan citranya. Sebelum citra ditampilkan, citra harus memiliki kesamaan format dengan komponen yang digunakan pada GUI yaitu QLabel. Komponen ini sendiri memiliki 2 format yang berbeda untuk citra RGB yaitu RGB 4 kanal dan 3 kanal.

Mengamati dari gambar 4.18. baris ke-2 hingga baris ke-6 digunakan sebagai perintah untuk mendeteksi jenis citra yang akan ditampilkan. Adapun 2 baris terakhir merupakan perintah untuk penyusunan ulang citra dengan ditambahi format sesuai yang telah dideteksi program sebelumnya. Penyusunan citra ini dibantu oleh modul QImage dari pustaka PyQt5.QtGui. Setelah melalui proses ini barulah citra siap untuk ditampilkan.

```
self.video_read.setPixmap(QPixmap.fromImage(outImage))
self.video_read.setScaledContents(True)
```

**Gambar.4.19** Perintah untuk menampilkan citra dalam fungsi `display_image`.

Pada gambar 4.19 tidak tampak penyebutan komponen QLabel karena dalam program ini komponen tersebut diberi nama `video_read`. Pemberian nama ini diberikan pada saat pembuatan desain GUI diawal. QLabel sendiri merupakan komponen desain GUI yang memiliki fungsi publik. Fungsi tersebut dipanggil untuk pengaturan terhadap properti QLabel. Adapun properti yang digunakan adalah `setPixmap` dan `setScaledContent`. Melalui pengaturan dalam property inilah QLabel dapat digunakan untuk menampilkan citra.

Seperti pada gambar di atas, QLabel memiliki fungsi `setPixmap` yang berfungsi untuk memberikan nilai balik pada komponen QLabel bahwa ada citra yang akan ditampilkan. Adapun baris program dibawahnya berfungsi untuk menampilkan citra pada QLabel dengan memenuhi area QLabel dalam istilah lain *fit* atau sesuai dengan ukuran QLabel. Ukuran dari QLabel sendiri disamakan dengan ukuran LCD yang digunakan yaitu 320x480 piksel.

Sampai pada bagian program ini, keseluruhan program belum bisa berjalan hal ini karena program belum masuk ke dalam fungsi utama. Fungsi utama ini akan dieksekusi secara berulang hingga ada perintah untuk menghentikannya. Fungsi utama dalam struktur penulisan bahasa pemrograman python sering diawali dengan `if __name__ == '__main__':` hal ini tentu berbeda dengan bahasa pemrograman lainnya. Awalan tersebut akan menjadi perintah untuk menjalankan program secara otomatis, jadi tanpa awalan ini maka program tidak akan berjalan dan hanya berfungsi sebagai modul saja.

Dalam fungsi utama tersebut berisikan pembuatan obyek dari kelas `veinviewer` yang diberi nama `window` seperti pada baris awal gambar 4.20. Selain membuat obyek, baris awal ini juga akan mengeksekusi fungsi `init` yang terdapat di dalam kelas `veinviewer`. Adapun fungsi lain belum dieksekusi.

```

window=veinviewer()
window.setWindowFlags(Qt.FramelessWindowHint)
window.setGeometry(0, 0, 480, 320)
window.show()

```

**Gambar.4.20** Perintah dalam fungsi utama.

Kemudian baris ke 2 dari gambar di atas merupakan fungsi untuk menghilangkan toolbar di desain `QDialog` sehingga desain akan tampak *flat*. Kemudian untuk baris selanjutnya berfungsi untuk menentukan titik posisi muncul GUI pada layar. Menganalisis dari program gambar 4.20 di atas, GUI dimunculkan pada posisi `x = 0` dan `y = 0` yaitu dimulai dari pojok kiri atas dan memanjang 480 piksel ke kanan dan 320 piksel ke bawah. Sedangkan untuk baris terakhir digunakan untuk menampilkan `QDialog` pada layar LCD.

Selain hanya dapat menampilkan citra pembuluh darah, program dalam alat ini juga dilengkapi dengan program untuk menyimpan citra yang diperoleh. Oleh

karena itu, dibuatlah tombol *save* sebagai bentuk intruksi untuk mengeksekusi program simpan file. Citra yang disimpan merupakan sebuah file dalam bentuk gambar PNG dengan nama file yang menyesuaikan dengan format tanggal dan waktu pengambilannya. Jadi file yang telah tersimpan tidak akan tergantikan dengan file yang baru. Perintah untuk simpan file ini berada semua di dalam fungsi *save*.

```
def save(self):
    dt = datetime.now()
    dt2 = dt.strftime('%d%m%y-%H%M%S')
    str_save = dt2 + '.png'
```

**Gambar.4.21** Perintah pengambilan data waktu dalam fungsi *save*

Membahas fungsi *save*, sebelum menyimpan file citra harus diambil terlebih dahulu data tanggal dan waktu dengan menambahkan perintah sebagaimana pada baris ke-2 hingga baris ke-4 dari gambar di atas. Perintah pengambilan data waktu ini menggunakan pustaka *datetime*. Kemudian barulah citra disimpan ke dalam bentuk gambar PNG yang didukung dengan pustaka *open cv*. Perintah yang digunakan dalam program adalah tampak sebagaimana pada gambar di bawah ini.

```
cv.imwrite('/home/pi/WISUDA18/93final ' + str_save, self.rgb_result)
```

**Gambar.4.22** Perintah simpan file citra menjadi gambar PNG.

Jika diamati dari program pada gambar 4.22, direktori tempat menyimpan file masih dituliskan dalam program dan tidak bias ditentukan ketika proses penyimpanan. Dalam alat ini sendiri gambar disimpan di direktori folder *WISUDA18*. Setelah gambar disimpan, pengguna akan mendapatkan informasi mengenai nama file sehingga dapat mengidentifikasinya jika suatu waktu dibutuhkan. Informasi tersebut muncul dalam bentuk pesan *Message Dialog* secara *pop-up* pada layar LCD. Pesan itu dimunculkan dengan perintah yang penulisannya sebagaimana gambar di bawah ini.

```
QMessageBox.information(self, 'Information', "You have succsesfully" _
_ "save all your images with filename:\n" + str_save, _
_ QMessageBox.Ok, QMessageBox.Ok)
```

**Gambar.4.23** Perintah untuk menampilkan box pesan (*message box*).

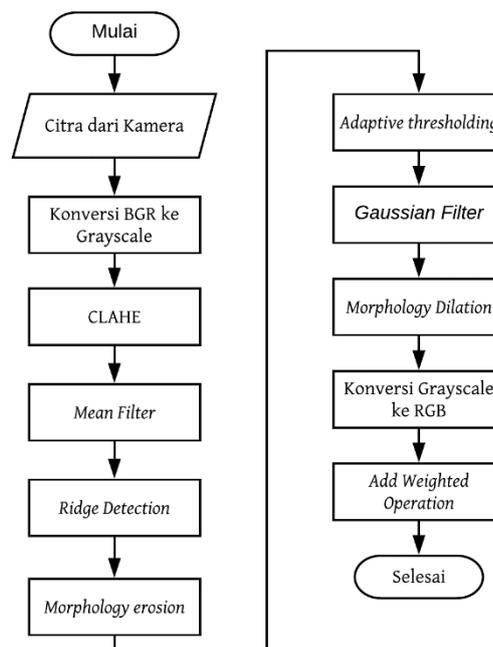
Pada saat alat difungsikan terkadang ditemui *bug* atau *error* terkhusus pada saat tampilan desktop baru berpindah dari HDMI ke layar LCD. Hal ini disebabkan

karena antarmuka kamera raspberry pi belum diaktifkan. Namun, selain masalah tersebut secara keseluruhan program berjalan dengan baik. Oleh karena itu dapat dikatakan melalui program sederhana di atas perangkat lunak alat sudah dapat berfungsi dengan baik.

## 4.2 Analisis Penelitian

### 4.2.1 Peningkatan Kualitas Citra

Dari hasil citra yang diperoleh oleh kamera dalam bentuk format BGR 8 bit, gambar pembuluh sudah terlihat tetapi tampak belum begitu jelas atau kabur. Oleh karena itu, diperlukan metode peningkatan kualitas citra (*Image Enhancement*) sehingga pada akhirnya akan memperjelas tampilan sekaligus mendapatkan pola dari pembuluh darah yang terdeteksi. Proses pengolahan citra yang dilakukan terdiri dari beberapa metode sebagaimana tampak seperti pada gambar diagram alir di bawah ini.

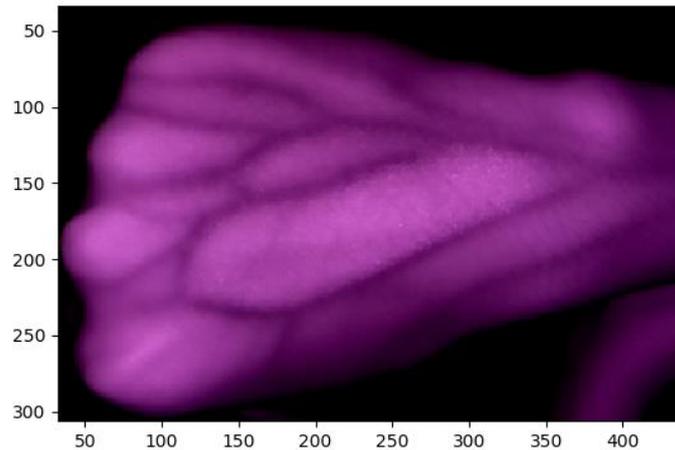


**Gambar.4.24.** Diagram alir saat proses pengolahan citra.

Adapun pembagian proses pengolahan citra yang digunakan terbagi menjadi tiga yaitu, *pre-processing* atau proses sebelum pengolahan citra yang utama, pengolahan citra yang utama dalam hal ini yaitu metode *adaptif thresholding*, dan *post-processing* atau proses yang dilakukan setelah pengolahan citra yang utama. Bagian *pre-processing* terdiri dari konversi citra ke dalam bentuk *grayscale*,

CLAHE, *mean filter*, *ridge detection*, dan morfologi. Sedangkan untuk *post-processing* terdiri dari filter gaussian, morfologi, konversi citra ke dalam bentuk RGB, operasi *addweighted*.

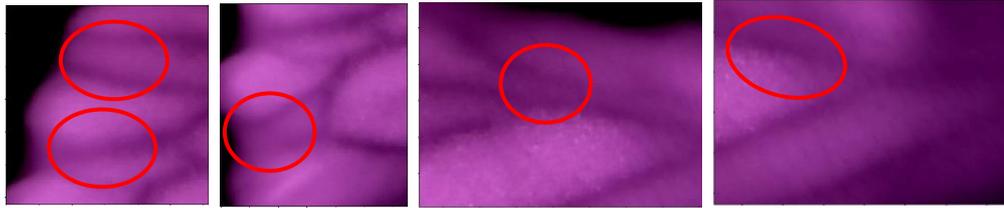
### 1. Citra dari Kamera



**Gambar.4.25.** Citra asli dari tangkapan kamera.

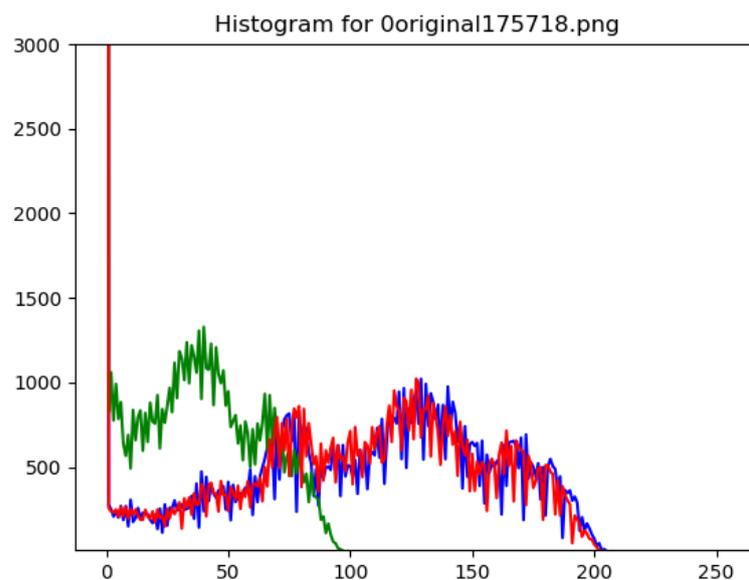
Citra asli hasil dari tangkapan kamera inframerah tampak seperti pada gambar 4.25. Dari gambar tersebut diketahui bahwa resolusi citra adalah 480 x 320 piksel. Ukuran resolusi tersebut sesuai dengan pengaturan resolusi citra di awal program. Resolusi ini dipilih karena tergolong ukuran resolusi yang kecil tetapi citra secara keseluruhan masih tampak jelas. Meskipun resolusi kecil, resolusi ini sangat membantu ketika pengolahan citra dibandingkan dengan ukuran resolusi yang lebih besar. Hal ini dikarenakan resolusi citra yang besar akan lebih memakan waktu dalam pengolahannya.

Apabila mengamati citra yang diperoleh dari kamera tersebut, pembuluh darah yang dideteksi sudah cukup terlihat jelas. Tampak warna obyek pembuluh lebih gelap dibandingkan dengan obyek lain disekitarnya. Kendatipun warna tersebut sudah berbeda tetapi, ada sebagian pembuluh darah yang tampak kabur seperti bagian yang ditunjukkan pada gambar di bawah ini. Dari bagian-bagian tersebut dapat dianalisis dan diketahui bahwa citra ini membutuhkan metode peningkatan kualitas citra sehingga obyek tampak lebih jelas.



**Gambar.4.26.** bagian obyek pembuluh darah pada citra yang warnanya tampak kabur (dilingkari lingkaran berwarna merah).

Citra yang diperoleh dari kamera ini bertipe BGR dengan warna kanal hijau atau G ( $G = Green$ ) tampak lebih gelap dibandingkan 2 kanal lainnya. Warna tersebut cenderung lebih gelap terutama di bagian obyek pembuluh darah dan memiliki nilai piksel yang kecil yaitu di bawah 20. Adapun bagian obyek lainnya yang terang memiliki nilai piksel kanal hijau yang relatif besar tepatnya di atas nilai 20. Analisis ini diperoleh dengan mengamati histogram citra yang diperoleh sebagaimana tampak pada gambar 4.27. Di histogram itu sendiri tampak nilai piksel 0 frekuensi kemunculannya lebih tinggi. Hal ini merepresentasikan bagian warna gelap yang mendominasi di citra lebih tepatnya bagian yang berada di sekitar tangan. Adapun untuk kanal biru (B) dan merah (R) penyebarannya lebih cenderung merata di seluruh nilai piksel meskipun masih ada sebagian nilai piksel yang tidak muncul di kedua kanal tersebut.



**Gambar.4.27.** Histogram citra asli.

## 2. Konversi Citra ke *Grayscale*

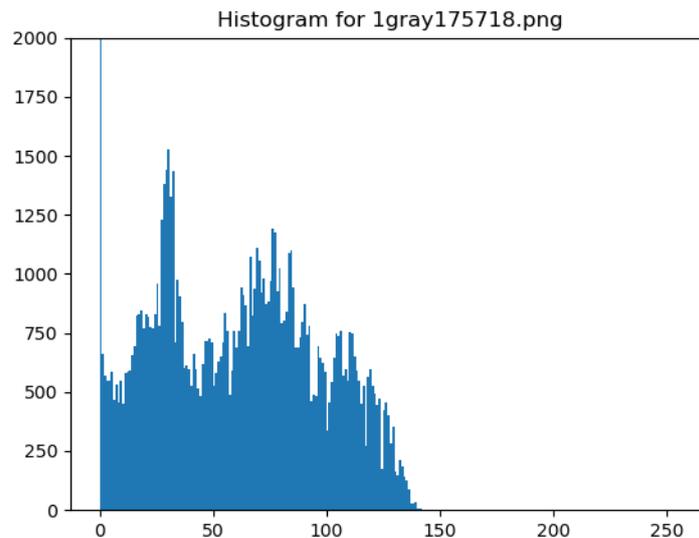
Apabila citra asli hasil tangkapan kamera diolah secara langsung, dapat memakan waktu proses sebagaimana resolusi yang besar. Oleh karena itu, citra tersebut dikonversi terlebih dahulu menjadi citra abu-abu (*grayscale*). Alasan lain mengapa citra ini dikonversi adalah banyaknya metode pengolahan citra selanjutnya yang lebih mendukung citra abu-abu dibandingkan dengan citra RGB atau BGR. Hal ini karena citra abu-abu hanya memiliki satu kanal warna sehingga pengolahannya lebih cepat.

Hasil konversi citra dari BGR ke abu-abu tampak seperti gambar 4.28 di bawah ini. Jika diamati pada citra tersebut, warna keunguan pada citra sebelumnya telah berubah menjadi keabu-abuan dengan intensitas warna yang tidak ada perubahan alias sama. Adapun warna obyek pembuluh darah tampak gelap kehitaman dan obyek disekitarnya terang keputihan.



**Gambar.4.28.** Hasil konversi citra menjadi citra abu-abu.

Bentuk histogram dari gambar abu-abu diatas tampak seperti gambar 4.29, 3 kanal warna pada citra sebelumnya telah menjadi satu hanya satu kanal. Jika diamati dari histogramnya, warna piksel cenderung merata di daerah warna gelap mendekati hitam dengan rentang antara 0 hingga 145. Penyebaran warna piksel di atas sangatlah tidak merata karena masih ada beberapa nilai piksel yang tidak muncul, sehingga dapat disimpulkan bahwa citra memiliki kontras dan kecerahan yang rendah dan perlu ditingkatkan.



**Gambar.4.29.** Histogram citra abu-abu.

### 3. CLAHE

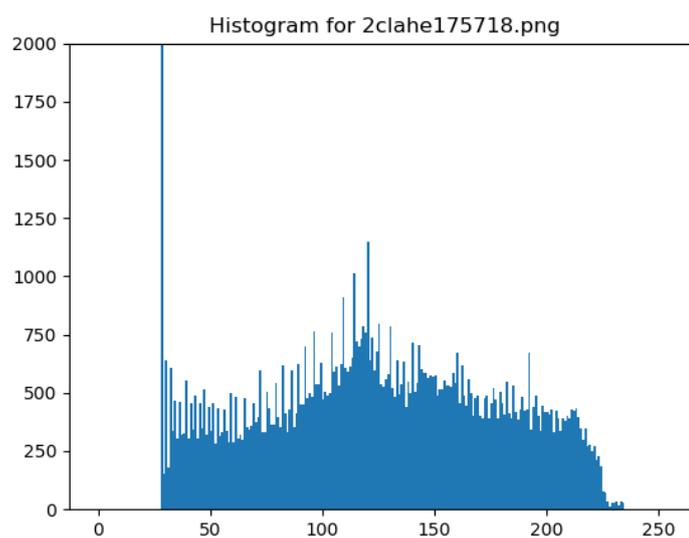
Permasalahan setelah citra dikonversi menjadi citra abu-abu adalah tingkat kecerahan dan kontras citra yang rendah. Oleh karena itu, proses pengolahan citra selanjutnya adalah meningkatkan kedua hal tersebut dengan menggunakan metode pemerataan histogram atau yang dikenal dengan *histogram equalizations*. Adapun metode pemerataan histogram yang digunakan adalah CLAHE (*Contrast Limited Adaptive Histogram Equalizations*). Hasil pengolahan citranya tampak pada gambar 4.30, citra terlihat menjadi lebih terang dan jelas dengan tidak menghilangkan informasi yang ada di dalamnya. Hal inilah yang melandasi mengapa metode CLAH lebih tepat digunakan, karena sebagian lain dari jenis metode ini menaikkan kontras dan kecerahan citra tanpa memperhatikan informasi



**Gambar.4.30.** Hasil citra setelah melewati proses CLAHE.

di dalamnya. Pada akhirnya hasil yang diperoleh dapat menghilangkan informasi di dalam citra dengan tingkat kecerahan citra yang terlalu tinggi. Tentunya informasi yang dimaksud dalam hal ini adalah pola pembuluh darah.

Setelah diproses, nilai piksel citra menyebar pada rentang 28 hingga 235. Penyebaran inilah yang menjadikan citra tampak lebih terang karena adanya nilai piksel yang mendekati warna putih yaitu 255. Nilai piksel yang besar tersebut mewakili daerah citra yang terang keputihan. Adapun bentuk penyebaran nilai pikselnya tampak seperti pada gambar histogram 4.31.



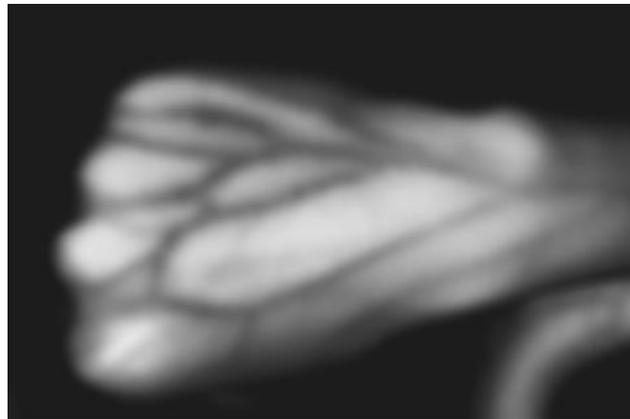
**Gambar.4.31.** Histogram citra CLAHE.

Jika diamati dari grafik histogram yang diperoleh, grafik tampak tidak beraturan dan cenderung memiliki ketinggian atau frekuensi kemunculan nilai piksel yang tidak rata. Bentuk histogram seperti inilah yang menunjukkan adanya *noise* atau gangguan pada citra. Adanya dan bertambahnya *noise* pada citra merupakan kelemahan dari diterapkannya metode CLAHE pada citra. Melalui masalah ini, *noise* pada citra harus dikurangi atau dihilangkan.

#### **4. Mean Filter**

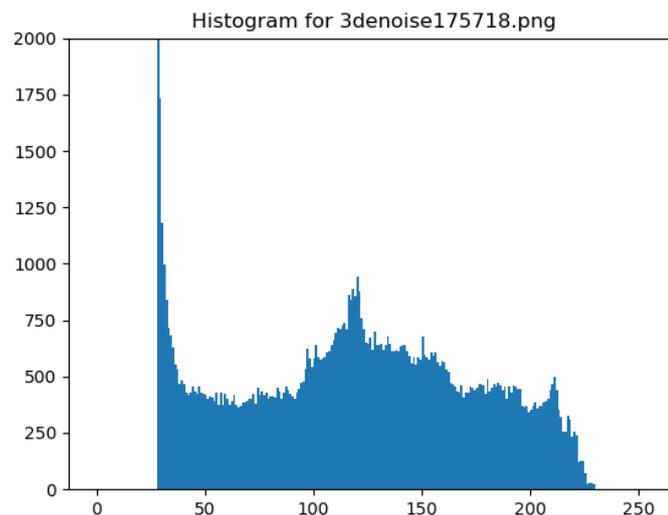
Setelah diketahui bahwa di dalam citra terdapat banyak *noise*, maka pada proses ini citra tersebut difilter (disaring) dengan tujuan mengurangi *noise* tersebut. Ada banyak berbagai jenis filter citra dan yang digunakan dalam metode ini adalah *mean filter* atau filter rata-rata. Filter ini mengubah nilai piksel dengan nilai rata-rata piksel sekitarnya sehingga citra yang dihasilkan tampak kabur atau buram

(*blur*) sebagaimana berikut. Meskipun citra yang diperoleh tampak buram, tetapi informasi pola pembuluh darah masih terlihat dan dapat dikenali.



**Gambar.4.32.** Hasil citra setelah melewati proses *mean filter*.

Dipilihnya filter ini disebabkan oleh pengolahannya yang cepat dan lebih tepat dibandingkan filter lainnya seperti *median filter* ataupun gaussian filter. *Median filter* akan lebih tepat digunakan untuk menghilangkan *noise* berbentuk bintik-bintik kecil seperti bintik garam (*salt-pepper noise*). Adapun filter gaussian memproses dengan waktu yang lambat yaitu sebesar 56 ms dibandingkan dengan filter ini yang hanya membutuhkan waktu 6 ms.



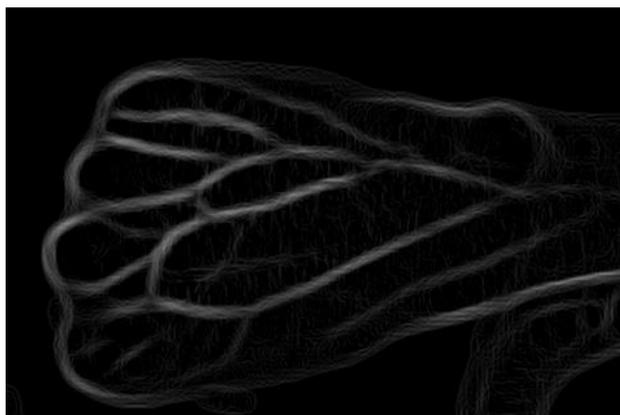
**Gambar.4.33.** Histogram citra setelah difilter.

Mengamati bentuk histogram citranya terlihat bahwa frekuensi kemunculan nilai piksel atau ketinggian grafiknya menjadi lebih merata dibandingkan dengan citra sebelumnya. Namun, nilai penyebaran pikselnya sendiri tidak mengalami

perubahan. Setelah melalui proses ini, selanjutnya citra siap untuk diekstrak pola pembuluh darahnya.

### **5. Ridge Filter Detection**

Setelah citra siap untuk diekstrak pola pembuluh darahnya, citra dimasukkan ke dalam proses filter yang ke dua. Filter ini mendeteksi daerah yang menonjol atau yang dikenal dengan istilah *ridge filter detection*. Dalam proses ini pola pembuluh darah dipisahkan berdasarkan warna. Sebagaimana yang tampak pada gambar 4.32 pada metode sebelumnya, warna pola pembuluh darah lebih gelap dibandingkan dengan warna lainnya. Warna gelap yang berada diantara 2 warna terang ataupun berada diperbatasan antara warna terang dan gelap akan dideteksi dalam proses ini kemudian warnanya akan dibedakan dengan warna disekitarnya. Adapun hasil proses pengolahannya tampak pada gambar di bawah ini.

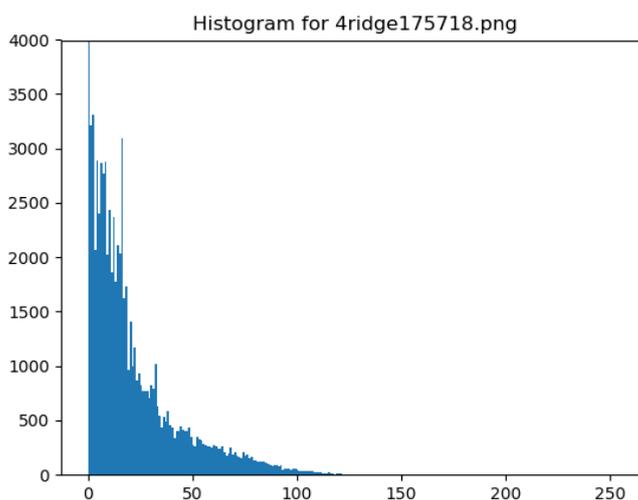


**Gambar.4.34.** Hasil citra setelah proses *ridge filter detection*.

Metode filter ini sangatlah efektif dibandingkan dengan filter pendeteksi garis tepi atau yang dinamai dengan *edge filter detection*. Filter garis tepi memberikan hasil luaran citra berupa 2 garis sebagai tepi dari pada obyek pembuluh darah. Terjadi demikian, karena filter jenis ini hanya mendeteksi tepi atau perbedaan warna antara gelap dan terang. Selain itu, alasan lainnya tidak menggunakan filter tersebut karena seringnya tidak terdeteksinya pola pembuluh darah disebabkan warnanya yang mendekati warna obyek sekitarnya.

Jika mengamati dari citra pada gambar 4.34, pola pembuluh darah tampak berwarna abu-abu terang sedangkan lainnya berwarna hitam. Pada gambar terlihat tidak hanya mendeteksi pola pembuluh darah saja melainkan bentuk punggung

tangan juga terdeteksi. Hal ini terlihat jelas dari bentuk punggung tangan yang juga berwarna abu-abu gelap. Bentuk punggung tangan ini tidak akan dihilangkan karena nantinya akan membantu memperjelas gambaran posisi asli pola pembuluh darah pada punggung tangan tersebut.

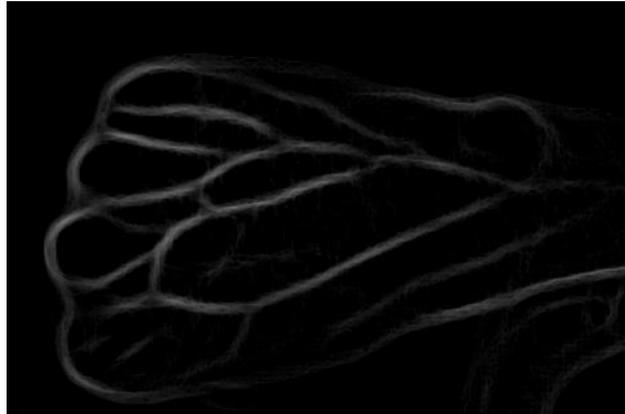


**Gambar.4.35.** Historam citra *ridge filter detection*.

Adapun bentuk grafik histogram dari citra ini tampak didominasi dengan warna gelap hitam. Hal ini terlihat dari histogram yang mana nilai piksel 0 muncul lebih sering begitu pula dengan nilai piksel yang mendekati nilai piksel 0. Oleh sebab citra yang tampak gelap, maka penyebaran nilai pikselnya sempit ataupun cenderung mendekati nilai piksel 0. Adapun rentang nilai pikselnya berada pada rentang 0 hingga 130. Meskipun citra gelap, secara umum pola pembuluh darah sudah tampak jelas.

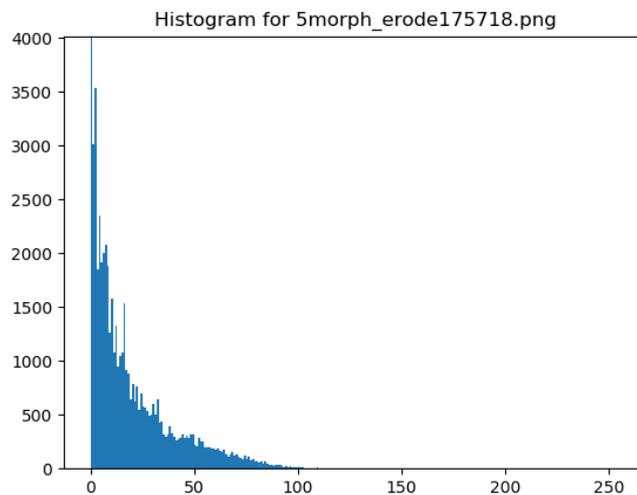
## **6. Erosion**

Menganalisis hasil deteksi pola pembuluh darah, tampak masih terdapat *noise* pada citra dalam bentuk garis-garis kecil dan tipis seperti rambut yang berada disekitar pola pembuluh darah. Hal ini akan menjadikan hasil citra setelah di *thresholding* di dalamnya banyak terdapat *noise*. Oleh karena itu sebelum di *thresholding*, citra sebelumnya dimasukkan ke dalam proses morfologi. Adapun hasil yang diperoleh sebagaimana pada gambar 4.36.



**Gambar.4.36.** Citra setelah melewati proses *erosion*.

Metode morfologi yang digunakan adalah *erosion*. Metode yang merupakan metode *pre-processing* terakhir ini sangat tepat untuk mengurangi atau menghilangkan suatu bentuk obyek tertentu pada sebuah citra yang berukuran kecil dan tipis. Disebabkan *noise* yang terdapat dalam citra berukuran kecil dan tipis, maka hal ini menjadi alasan mengapa metode *erosion* digunakan. Sebagaimana tampak pada gambar 4.36, *noise* dari citra sebelumnya sudah berkurang dan tidak terlihat.



**Gambar.4.37.** Histogram citra setelah proses *erosion*.

Mengamati histogram citranya, terlihat histogram citra yang serupa dengan citra pada proses sebelumnya tetapi terdapat perbedaan yaitu berkurangnya rentang penyebaran piksel dan frekuensi kemunculan nilai piksel. Penyebaran nilai piksel berubah menjadi diantara 0 hingga 115. Penyebab hal ini adalah hilangnya *noise*. *Noise* yang sebelumnya berwarna abu-abu dan terlihat jelas telah hilang, sehingga

frekuensi kemunculan nilai piksel pada warna tersebut menjadi ikut berkurang. Hilangnya nilai piksel tersebut digantikan dengan nilai piksel 0 yaitu warna hitam gelap sehingga tampak pada histogram nilai piksel 0 lebih banyak dan mendominasi.

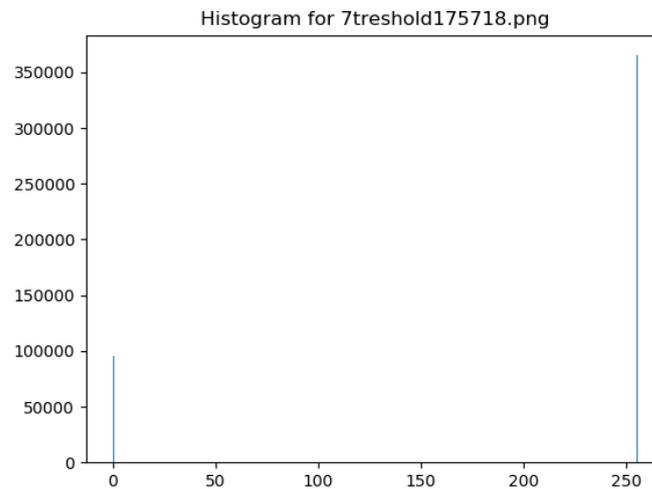
### **7. *Thresholding Adaptive***

Proses ini merupakan metode inti yang digunakan untuk meningkatkan kualitas citra pembuluh darah yang diperoleh. Dalam praktiknya metode ini sangatlah sederhana yaitu mengubah citra menjadi citra biner. Pada saat proses pengolahan citra, metode ini menggunakan parameter `BINARY_INV` yang dimasukkan ke dalam program. Adapun maksud dari parameter tersebut yaitu membalikan warna menjadi warna kebalikannya. Maksudnya adalah citra sebelumnya yang didominasi hitam gelap berubah menjadi didominasi warna putih terang, sehingga hasil dari pola pembuluh darah menjadi tampak sangat jelas. Melalui pengamatan terhadap citra yang diperoleh masih terdapat *noise* yang harus dihilangkan. *Noise* tersebut terlihat jelas bersamaan dengan jelasnya pola pembuluh darah dan tentunya hal ini harus dihilangkan.



**Gambar.4.38.** Citra setelah di-*thresholding*.

Adapun analisis terhadap histogram citra tidak tampak grafik dengan bentuk tertentu melainkan dua garis di kedua ujung nilainya. Hal ini karena citra ini merupakan citra biner. Oleh sebab citra biner itulah, menyebabkan histogram hanya muncul pada nilai piksel 0 dan 255 dengan nilai 255 lebih mendominasi karena citra lebih banyak mengandung warna putih dibandingkan hitam.



**Gambar.4.39.** Histogram citra setelah di-*tresholding*.

### 8. *Gaussian Filter*

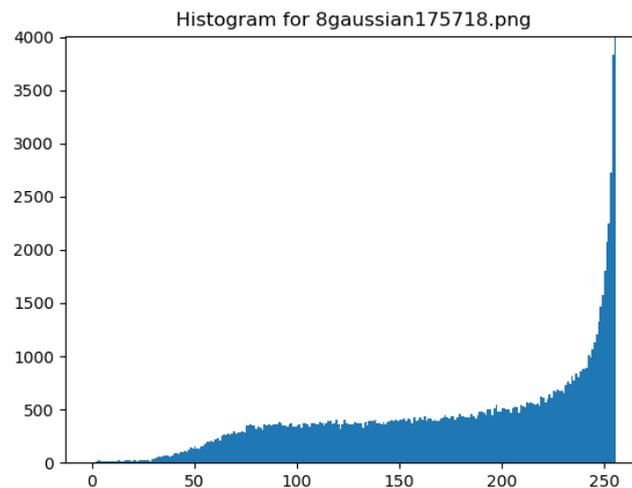
Setelah melalui inti proses yaitu *thresholding*, selanjutnya adalah proses *post-processing* yaitu menjadikan citra yang telah diproses dengan *tresholding* menjadi lebih baik lagi. Oleh sebab adanya *noise* pada citra, maka dibutuhkan filter untuk menghilangkannya. Adapun filter yang digunakan pada proses ini yaitu filter gaussian. Salah satu alasan mengapa filter jenis ini yang digunakan adalah filter ini sangat tepat digunakan untuk merapikan tepi suatu citra sehingga menjadi lebih lembut dan tidak kasar. Selain itu mampu menghilangkan *noise* yang berukuran kecil.



**Gambar.4.40.** Citra setelah difilter.

Mengamati dari gambar di atas, luaran citra yang diperoleh tampak kabur, tetapi informasi pola pembuluh darah tetap terlihat bahkan menjadi lebih lembut dan rapi. Pada gambar di atas, tampak warna hitam menjadi lebih melebur, sehingga

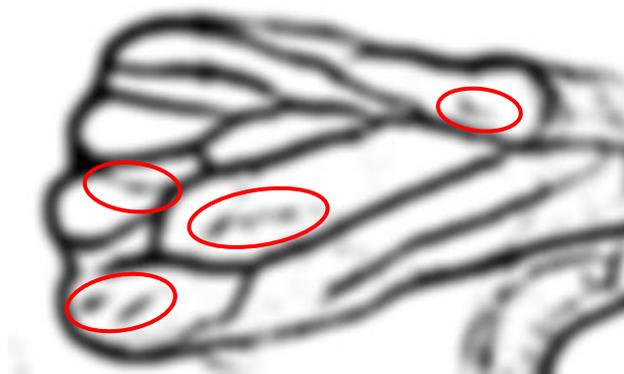
tampak seperti munculnya gradasi warna. Dianalisis dari histogram citra, tampak nilai piksel 0 berkurang sedangkan nilai piksel diantara 0 dan 255 muncul lebih banyak. Hal inilah efek dari munculnya gradasi warna hasil luaran citra dari gaussian filter.



**Gambar.4.41.** Histogram citra setelah melalui filter gaussian.

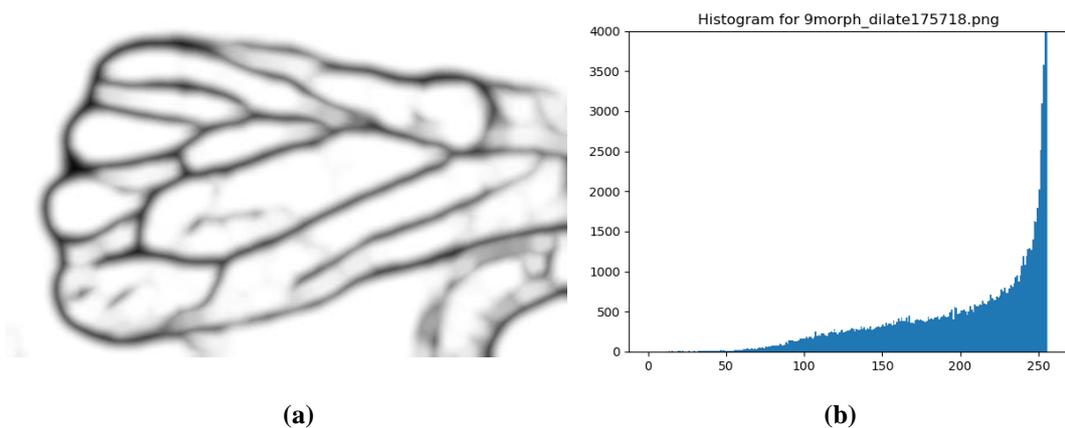
### 9. Dilation

Hasil citra sebelumnya telah mengurangi *noise* yang tampak tetapi masih terdapat *noise* yang obyeknya berukuran cukup besar. Apabila *noise* ini dihilangkan dengan filter gaussian akan mengakibatkan citra sangat kabur yang bisa menghilangkan informasi pola pembuluh darahnya. Bagian *noise* tersebut ditunjukkan oleh linkaran merah pada gambar 4.42. Adapun proses yang digunakan untuk menghilangkan *noise* tersebut yaitu morfologi metode *dilation*.



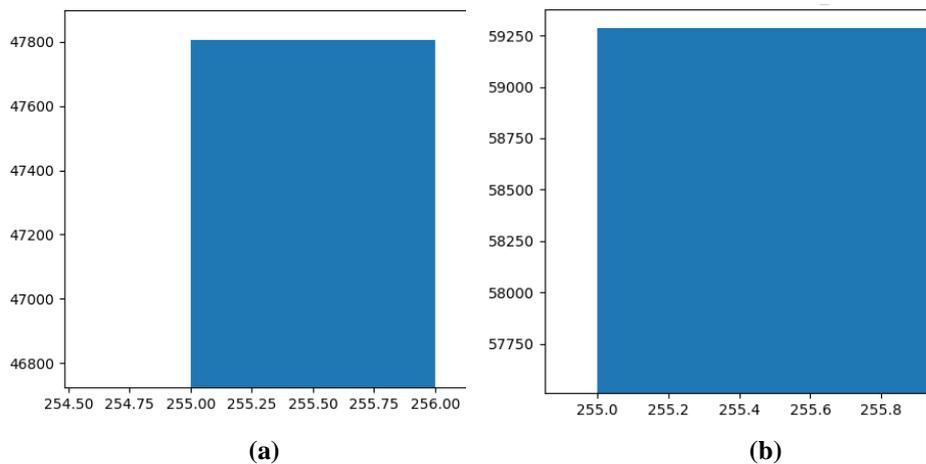
**Gambar.4.42.** *Noise* citra yang ukurannya cukup besar.

Metode ini sama seperti halnya *erosion*. Perbedaannya terletak pada dimana metode tersebut diterapkan. Apabila citra biner ini berwarna putih pola pembuluh daranya, maka metode yang tepat adalah *erosion*. Kemudian jika citra yang diperoleh ini menggunakan *erosion* akan menjadikan *noise* lebih tampak dan mempertebal citra pembuluh daranya. Adapun hasil dari metode ini tampak pada gambar 4.43.



**Gambar.4.43.** Citra hasil *diltion* (a) beserta dengan histogramnya (b).

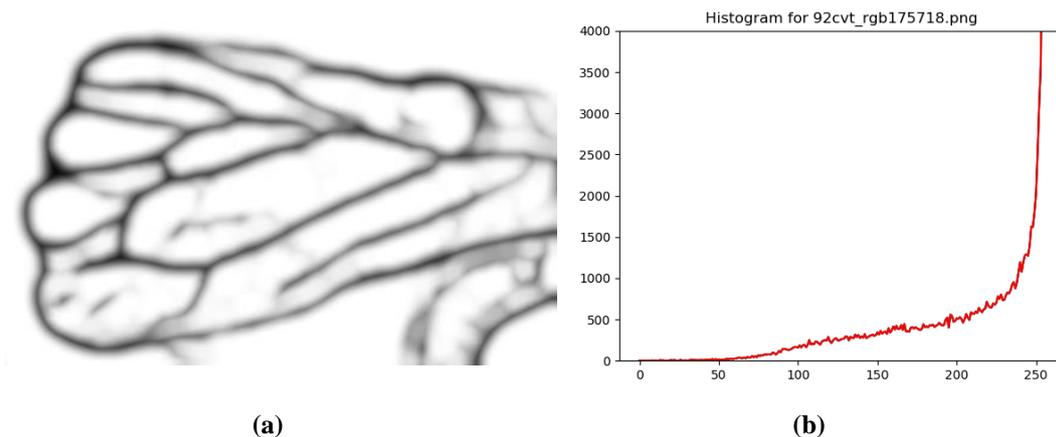
Hasil yang diperoleh tampak ukuran pembuluh darah lebih kecil dibandingkan citra sebelumnya dan hal terpenting *noise* pada citra dapat berkurang. Bagian piksel *noise* yang hilang digantikan dengan nilai piksel berwarna terang, sehingga nilai piksel warna gelap menjadi berkurang. Bersamaan dengan berkurangnya warna gelap tersebut menyebabkan warna terang lebih dominan. Hal ini sesuai dengan histogram citra yang diperoleh. Tampak grafik histogram pada gambar 4.43.b yang bentuknya lebih melengkung dibandingkan pada citra sebelumnya dengan didominasi kemunculan nilai piksel warna terang. Jika menganalisa dan membandingkan dengan histogram sebelumnya, diperoleh nilai piksel histogram citra *dilation* frekuensi kemunculannya 59250 kali sedangkan histogram sebelumnya hanya 47800 kali. Perbandingan ini dapat diamati di gambar 4.44 a dan b.



**Gambar.4.44.** Maksimal nilai piksel 255 pada citra filter gaussian sebelumnya (a) dan *dilation* (b).

## 10. Konversi Citra ke RGB

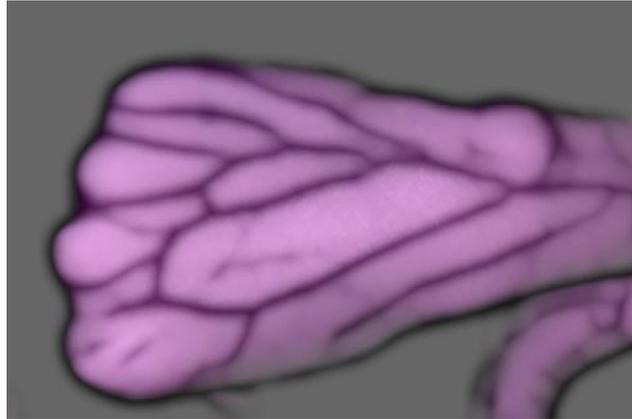
Adapun proses terakhir sebelum citra pola pembuluh darah digabungkan dengan citra aslinya, terlebih dahulu citra abu-abu ini dikonversi menjadi citra RGB. Hal ini disebabkan citra abu-abu tidak bisa digabungkan dengan citra RGB ataupun BGR karena berbeda jumlah kanal warnanya. Meskipun berbeda urutan kanal dengan citra aslinya yaitu BGR, keduanya tetap bisa digabungkan karena memiliki kesamaan 3 kanal warna. Dalam proses ini citra tidak tampak perubahannya begitupula dengan histogramnya.



**Gambar.4.45.** Citra hasil konversi ke citra RGB (a) dan histogramnya (b).

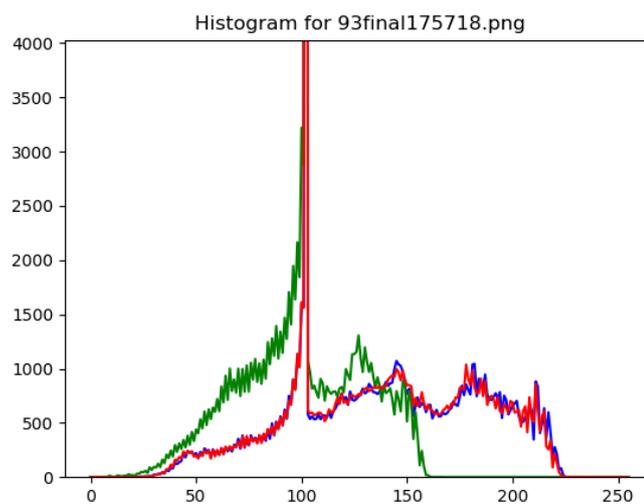
### 11. Operasi *Add Weighted*

Setelah citra asli dan citra pola pembuluh darah, keduanya memiliki kesamaan jumlah kanal, maka keduanya dimasukkan ke dalam operasi pertambahan. Operasi ini sama seperti aritmatika umumnya. Adapun hasil yang diperoleh nantinya adalah penggabungan dari kedua citra seperti pada gambar 4.46.



**Gambar.4.46.** Citra hasil akhir setelah ditingkatkan kualitasnya.

Menganalisis dari citra yang diperoleh tampak warna putih citra digantikan dengan warna citra aslinya yaitu warna ungu cerah. Selain itu warna selain obyek tangan berwarna abu-abu terang yang sebelumnya berwarna putih. Oleh sebab citra yang diperoleh merupakan penggabungan antara dua citra, maka histogram yang diperoleh juga menunjukkan hal yang sama.



**Gambar.4.47.** Histogram citra hasil akhir.

Menganalisis histogram citra, maka terdapat nilai frekuensi kemunculan yang sangat tinggi yaitu pada nilai piksel 102. Terjadi demikian karena nilai tersebut merupakan frekuensi kemunculan nilai 255 dari citra pola pembuluh darah. Grafik citra pada rentang 0 hingga 102 mewakili citra pola pembuluh darah sedangkan citra 102 hingga 255 mewakili citra asli sebelum diproses. Oleh sebab citra pola pembuluh darah berada pada daerah yang mendekati warna gelap (nilai piksel = 0), maka citra ini mempertahankan tingkat kegelapannya sedangkan citra asli menjadi lebih terang. Alasan inilah yang mendasari mengapa warna ungu citra asli menjadi terang dan daerah selain obyek tangan menjadi abu-abu terang.

Nilai piksel 102 yang disebutkan sebelumnya ini merupakan titik temu antara kedua citra dan nilai ini dapat digeser antara 0 hingga 255. Pergeseran ini tergantung dari parameter dimasukkan ke dalam program. Tampak pada gambar 4.48 merupakan program yang diterapkan dengan nilai bobot (*weight*) alpha 0.4 untuk citra pola pembuluh darah. Didalam program diberi nama dengan variable `cvt_rgb`. Sedangkan citra asli diberi nama variabel `image` dengan nilai bobot (*weight*) beta 0.6. Melalui salah satu parameter tersebut dapat diketahui titik temu di antara kedua citra dengan perhitungan sebagai berikut,

$$x = 256 \times 0.4$$

$$x = 102.4$$

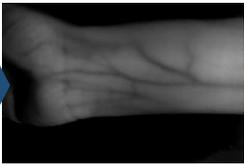
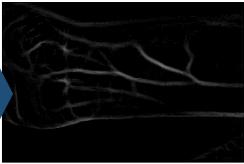
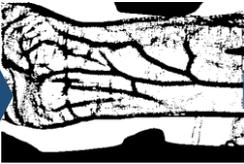
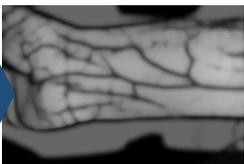
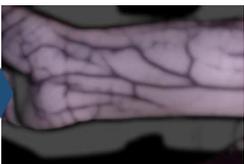
dimana  $x$  merupakan nilai piksel sebagai titik temu kedua citra dan 256 merupakan nilai piksel keseluruhan citra 8 bit.

```
self.rgb_result = cv.addWeighted(self.cvt_rgb, 0.4, self.image, 0.6, 0)
```

**Gambar.4.48.** Program operasi penambahan antara dua citra.

Keseluruhan proses pengolahan citra yang telah dijelaskan di atas, dapat disimpulkan proses yang dilakukan sebagaimana tampak pada tabel 4.5. Secara keseluruhan terdapat 10 tahapan yang dilalui untuk diperoleh citra pembuluh darah yang lebih jelas.

**Tabel.4.5** Kesimpulan dari seluruh metode pengolahan citra yang digunakan.

1. Citra asli	2. <i>Grayscale</i>	3. <i>CLAHE</i>	4. <i>Mean filter</i>
			
5. <i>Ridge Detection</i>	6. <i>Erosion</i>	7. <i>Adaptive Threshold</i>	8. <i>Gaussian filter</i>
			
9. <i>Dilation</i>	10. <i>RGB Image</i>	11. <i>Final Result</i>	
			

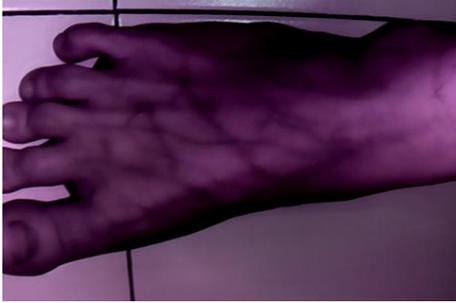
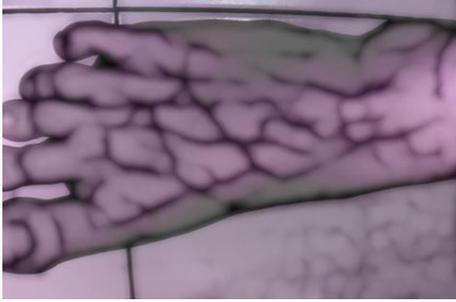
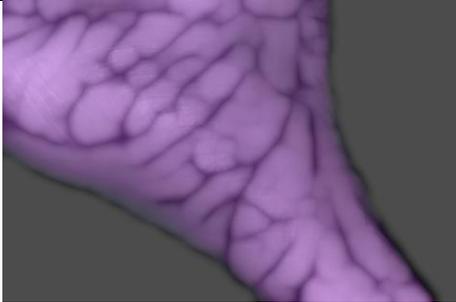
Setelah proses perancangan alat dan penerapan metode *thresholding adaptive* selesai dilakukan, kemudian alat ini diujikan terhadap beberapa obyek yang berbeda. Tujuan dari hal ini adalah untuk diketahuinya keandalan metode yang digunakan dan diterapkan di dalam alat.

Pertama kali alat ini diujikan pada 4 obyek bagian tubuh yang berbeda, yaitu punggung tangan, lengan, bagian pinggir kaki, dan punggung kaki. Adapun hasil pengujian yang diperoleh tampak pada tabel 4.6. Kemudian untuk pengujian kedua, alat ini juga coba diujikan terhadap 3 jenis koresponden yang berbeda sebagai obyeknya. Ketiga koresponden tersebut yaitu, pediatric (anak kecil), orang gemuk, dan orang ramping. Adapun hasil dari pengujian ini tampak pada tabel 4.7.

**Tabel.4.6** Data hasil pengujian terhadap obyek bagian tubuh yang berbeda.

No	Obyek	Citra sebelum diproses	Keterangan
1	Punggung tangan		Setelah diproses obyek pola pembuluh darah menjadi <b>tampak lebih jelas</b>
			
2	Lengan		Setelah diproses obyek pola pembuluh darah menjadi <b>tampak lebih jelas</b>
			

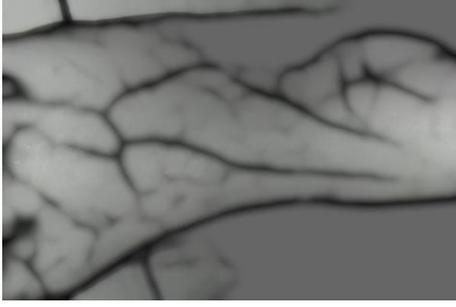
Lanjutan Tabel.4.6 Data hasil pengujian terhadap obyek bagian tubuh

No	Obyek	Citra sebelum diproses	Keterangan
3	Punggung kaki		Setelah diproses obyek pola pembuluh darah menjadi <b>tampak lebih jelas</b>
		Citra setelah diproses	
			
4	Bagian pinggir kaki	Citra sebelum diproses	Setelah diproses obyek pola pembuluh darah menjadi <b>tampak lebih jelas</b>
			
		Citra setelah diproses	
			

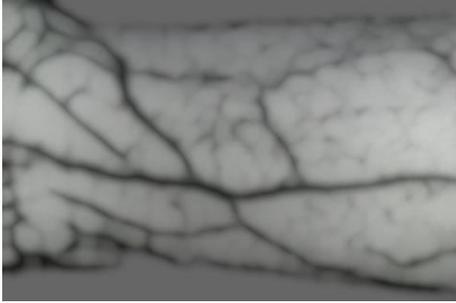
Dari tabel di atas tampak jelas bahwa, citra pola pembuluh darah yang sebelumnya tampak kurang jelas setelah diproses menjadi lebih jelas dan terang.

Melalui data di atas dapat disimpulkan bahwa alat yang dirancang dan metode yang digunakan dapat diterapkan untuk mendeteksi pola pembuluh darah di keempat bagian tubuh. Dipilihnya 4 bagian tubuh di atas disebabkan akses vaskular sering dilakukan melalui keempat bagian tubuh di atas.

**Tabel.4.7** Data Hasil pengujian terhadap 3 koresponden yang berbeda.

No	Koresponden	Citra sebelum diproses	Keterangan
1	Anak kecil		Setelah diproses obyek pola pembuluh darah menjadi <b>tampak lebih jelas</b>
		Citra setelah diproses 	
2	Orang Gemuk	Citra sebelum diproses	Sebelum diproses obyek pola pembuluh darah <b>tampak kurang jelas</b>
			

**Lanjutan Tabel.4.7** Data Hasil pengujian terhadap 3 koresponden yang berbeda.

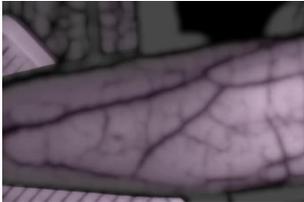
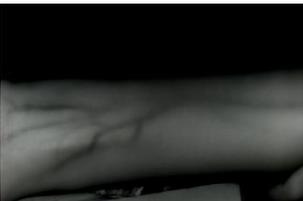
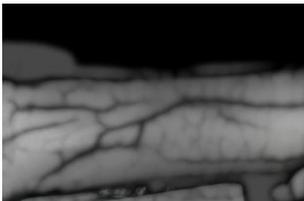
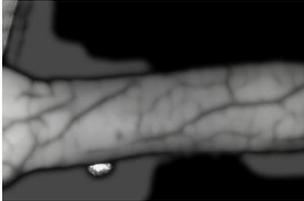
No	Koresponden	Citra setelah diproses	Keterangan
2	Orang Gemuk		Setelah diproses obyek pola pembuluh darah menjadi <b>tampak lebih jelas</b>
3	Orang Ramping	Citra sebelum diproses	Setelah diproses obyek pola pembuluh darah menjadi <b>tampak lebih jelas</b>
		Citra setelah diproses	

Melalui data tabel di atas dapat dianalisis bahwa, dari ketiga responden, hasil citra pola pembuluh darah yang diperoleh tampak kabur dan kurang jelas. Kemudian setelah diproses menjadi lebih terang dan jelas. Maksud dari citra yang jelas adalah terlihatnya pola pembuluh darah dengan warna yang lebih gelap dibandingkan dengan citra yang belum diproses. Sedangkan obyek lain disekitar pembuluh darah setelah diporses menjadi lebih terang. Melalui analisis dan data yang diperoleh dapat disimpulkan bahwa alat ini dapat digunakan dan

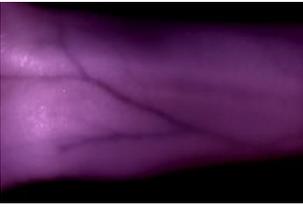
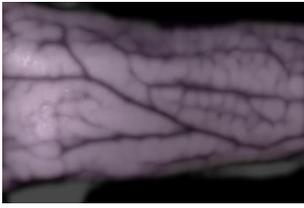
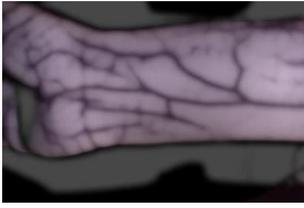
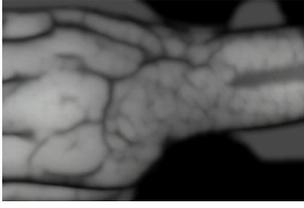
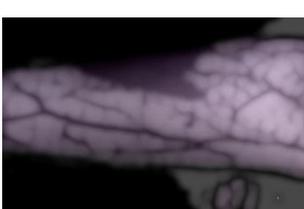
diimplementasikan terhadap 3 koresponden yaitu, anak kecil, orang gemuk, dan orang ramping (orang kurus).

Sebagai data penunjang dari metode pengolahan citra yang digunakan, di bawah ini tabel menunjukkan pengujian terhadap 20 koresponden yang berbeda, lengkap dengan perubahan citranya.

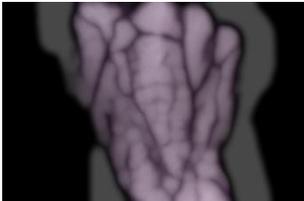
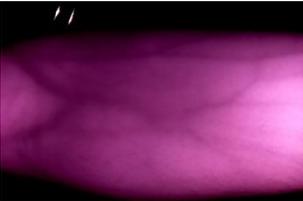
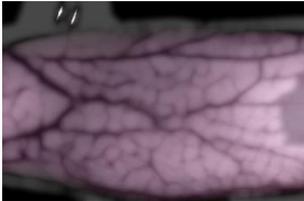
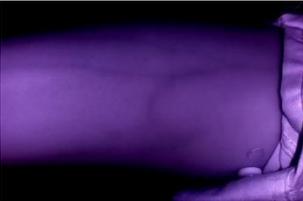
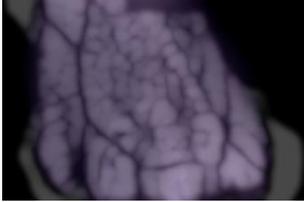
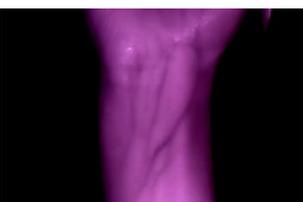
**Tabel.4.7** Hasil pengujian terhadap 20 koresponden.

No	Koresponden	(L/P)	Citra dari kamera	Citra setelah diproses
1	Rina	P		
2	Ulfi	P		
3	Merina	P		
4	Ayu	P		

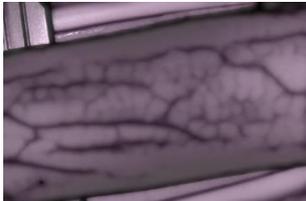
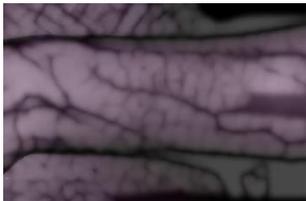
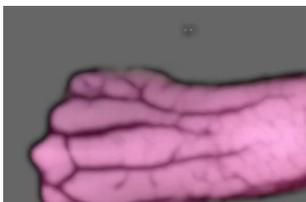
Lanjutan Tabel.4.7 Hasil pengujian terhadap 20 koresponden.

No	Koresponden	(L/P)	Citra dari kamera	Citra setelah diproses
5	Doane	P		
6	Putri	P		
7	Wahyu Sari	P		
8	Hasna	P		
9	Emya	P		
10	Rana	P		

Lanjutan Tabel.4.7 Hasil pengujian terhadap 20 koresponden.

No	Koresponden	(L/P)	Citra dari kamera	Citra setelah diproses
11	Irpan	L		
12	Afif	L		
13	Bagus	L		
14	Irfan Tri	L		
15	Gilang	L		
16	Aditiyo	L		

Lanjutan Tabel.4.7 Hasil pengujian terhadap 20 koresponden.

No	Koresponden	(L/P)	Citra dari kamera	Citra setelah diproses
17	Hasan	L		
18	Lutfi	L		
19	Iskak	L		
20	Safei	L		

Tampak pada tabel data di atas, disandingkan bersamaan didalamnya citra yang diperoleh dari kamera dan hasil citra setelah diolah sehingga dapat diketahui perbedaannya. Pengujian pada tabel ini dibedakan antara koresponden wanita dan pria tetapi tidak diklasifikasikan kategori gemuk atau kurus. Seluruh koresponden dipilih secara acak antara gemuk dan kurus. Jika mengamati dari tabel di bawah, tampak bahwa metode yang digunakan sudah cukup baik jika diimplementasikan untuk memperjelas pola pembuluh darah. Pada data 20 koresponden diatas terdiri dari 10 orang wanita dan 10 orang pria.

#### 4.2.2 Efisiensi Waktu dan *Frame rate*

Selama citra diproses dan diolah untuk ditingkatkan kualitasnya, tidak terlepas kaitannya dengan waktu citra tersebut diproses. Kemudian nantinya waktu ini akan berkaitan juga dengan *frame rate* karena citra yang ditampilkan pada LCD dalam bentuk video *real-time*. Pengujian waktu proses dilakukan dengan menambahkan baris program penghitung waktu sebagaimana pada gambar 4.49.

```
T_0 = time()
#perintah/fungsi pengolahan citra
print("gray :" + str((time()-T_0)*1000))
```

**Gambar.4.49** Baris program penghitung waktu proses pengolahan citra.

Pengujian ini dilakukan sebanyak 10 kali yang kemudian diambil rata-rata dari 10 sampel data tersebut. Adapun data waktu proses pengolahan citra yang diperoleh tampak pada tabel 4.8 di bawah ini.

**Tabel.4.8** Waktu proses pengolahan citra.

No	Metode Pengolahan Citra	Waktu proses (ms)
1	Konversi ke <i>Grayscale</i>	2.534
2	CLAHE	5.910
3	<i>Mean Filer</i>	6.645
4	<i>Ridge Filter Detection</i>	96.281
5	<i>Erosion</i>	7.353
6	<i>Adaptive Thresholding</i>	21.622
7	<i>Gaussian Filter</i>	56.423
8	<i>Dilation</i>	37.637
9	Konversi ke RGB	1.633
10	Operasi <i>Add Citra</i>	21.387
11	<b>Total Keseluruhan</b>	<b>252.156</b>

Dari tabel di atas dapat dianalisis bahwa metode yang banyak membutuhkan waktu banyak dalam memproses citra yaitu *Ridge Filter Detection*. Adapun metode yang waktu prosesnya paling cepat adalah pada saat konversi citra dari bentuk *grayscale* menjadi RGB. Total keseluruhan waktu pengolahan citra adalah 252.156

ms. Perlu diketahui bahwa video tersusun atas beberapa frame, dan frame tersebut merupakan sebuah citra tetap. Oleh sebab itu, melalui penjelasan ini dapat diketahui bahwa untuk memproses citra setiap frame membutuhkan waktu sebagaimana waktu mengolah citra hingga hasil akhir yaitu 252.156 ms.

**Tabel.4.9** Hasil pengujian *frame rate* video.

No	Pengujian	Data <i>Frame rate</i>
1	Frame rate maks	3.6
2	Frame rate min	3.25

Disamping pengujian terhadap waktu proses, diujikan juga *frame rate* video yang kemudian diperoleh data sebagaimana tabel di atas. Hasil data ini diperoleh dengan menambahkan program penghitung *frame rate* menggunakan pustaka FPS.



**Gambar.4.50** Pengujian *frame rate* video dan datanya ditampilkan *real-time* (data frame dilingkari lingkaran merah).

Kemudian data hasil perhitungannya tersebut ditampilkan pada saat alat digunakan. Melalui tabel 4.9 diketahui bahwa hasil citra yang ditampilkan dalam bentuk video memiliki *frame rate* maksimum 3.6 dan minimum 3.25. Jika nilai datanya dibulatkan, maka data *frame rate* ini menunjukkan bahwa selama satu detik alat mampu mengolah citra sebanyak 4 frame.

#### 4.2.3 Jarak Optimum dan Maksimum

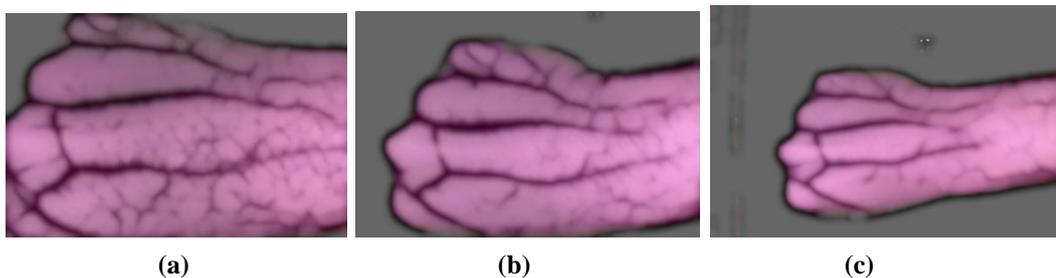
Diperolehnya hasil citra yang baik dipengaruhi juga oleh jarak kamera terhadap obyek yang dideteksi. Penelitian ini juga dilengkapi dengan analisis

terhadap jarak optimum alat dapat menangkap dan memperoleh citra yang baik dan jelas. Setelah melakukan pengujian diperoleh data tabel sebagaimana di bawah ini.

**Tabel.4.10** Hasil pengujian jarak optimum.

No	Jarak	Keterangan
1	5 cm	Kabur, Tidak jelas
2	10 cm	Kabur, Tidak jelas
3	15 cm	Kurang jelas
4	20 cm	Jelas
5	25 cm	Jelas
6	30 cm	Kurang jelas
7	35 cm	Kabur
8	40 cm	Kabur
9	45 cm	Kabur, Tidak jelas
10	50 cm	Kabur, Tidak jelas

Apabila data tabel di atas dianalisis bahwa citra yang diperoleh sangat baik berada pada jarak 20 – 25 cm. Pada jarak tersebut diperoleh citra yang jelas tidak kabur dan sedikit *noise* dibandingkan dengan jarak lainnya. Jarak ini merupakan jarak fokus kamera yang telah diatur pada jarak 20 cm. Jarak pengaturan ini dipertimbangkan atas penyebaran pancaran sinar inframerah karena pada jarak tersebut sinar inframerah lebih banyak menyebar pada obyek.



**Gambar.4.51** Hasil pengujian jarak 15 cm (a), jarak 20 cm (b), dan jarak 30 cm (c) terhadap obyek.

Menganalisis kembali dari gambar dan tabel di atas, jarak 30 cm dan 15 cm citra pola pembuluh darah masih dapat terlihat tetapi kurang jelas. Adapun jarak diluar lingkup jarak tersebut akan diperoleh citra yang tidak jelas dan kabur. Oleh karena itu, dapat disimpulkan bahwa jarak optimum alat adalah 20 – 25 cm dengan

jarak maksimum 30 cm dan jarak minimum 15 cm. Setelah pengujian terhadap alat dilakukan, pada bagian akhir ini dibahas mengenai opini paramedic terhadap alat yang dibuat.

#### 4.2.4 Manfaat Alat

Dalam mengetahui seberapa besar manfaat dari alat yang dibuat, berikut hasil survei terhadap paramedis setelah menggunakan alat ini. Adapun survei yang diberikan kepada paramedis adalah dalam bentuk quisioner bersikan beberapa pertanyaan yang berkaitan dengan alat. Survei diberikan kepada 5 paramedis dengan hasil sebagaimana tabel berikut.

**Tabel.4.10** Hasil survei terhadap manfaat yang diberikan alat kepada paramedis.

No	Pernyataan	Point
1	Desain alat	3.4
2	<i>User Friendly</i> dalam penggunaanya	4.2
3	Kemudahan memahami display/GUI	4
4	Kualitas citra yang ditampilkan LCD	3.2
5	Manfaat dalam dunia medis	4.4
6	Keseluruhan alat	4

Nilai poin di atas merupakan nilai komulatif dari 5 koresponden yang mencoba alat secara langsung dan diberikan pertanyaan koresponden. Nilai poin paling terendah ada pada pernyataan desain dengan poin 3.4 dan kualitas citra yang ditampilkan di LCD dengan poin 3.2. Menganalisis dari pendapat dan masukan dari koresponden mengenai desain alat, sebagian besar menyarankan untuk membuat desain menjadi lebih kecil sehingga mudah digenggam.

Adapun untuk kualitas citra, sebagian besar berpendapat bahwa citra akan kabur bila obyek bergerak sedikit. Hal ini tentunya berkaitan dengan *frame rate* video yang rendah dan menjadi sebab citra akan kabur bila obyek terlalu banyak bergerak. Selain itu, resolusi citra yang kecil membuat citra tampak kurang baik. Penyebab hal ini adalah ukuran citra yang menyesuaikan dengan ukuran LCD yang kecil hanya berukuran 480 x 320 piksel atau 3.5”.

Melalui tabel 4.10 di atas, diketahui juga bahwa poin tertinggi ada pada data penggunaan alat yang sangat mudah dan manfaat alat jika diterapkan dalam dunia medis. Keduanya memiliki nilai poin di atas 4 yaitu, 4.2 untuk parameter *user friendly* dan 4.4 untuk manfaat yang dirasakan paramedis. Disamping itu, kedua parameter ini sangat mendukung alasan betapa besarnya manfaat kepada paramedis apabila alat yang dibuat tersebut digunakan dan diterapkan dalam praktik medis.