3.3 Realisasi

Pada tahap realisasi, setelah proses perancangan telah selesai maka proses selanjutnya merealisasikan rancangan yang sudah dibuat untuk menjadi alat yang utuh. Pada bagian ini akan dijelaskan perihal proses realisasi alat.

3.3.1 Realisasi Perangkat Keras

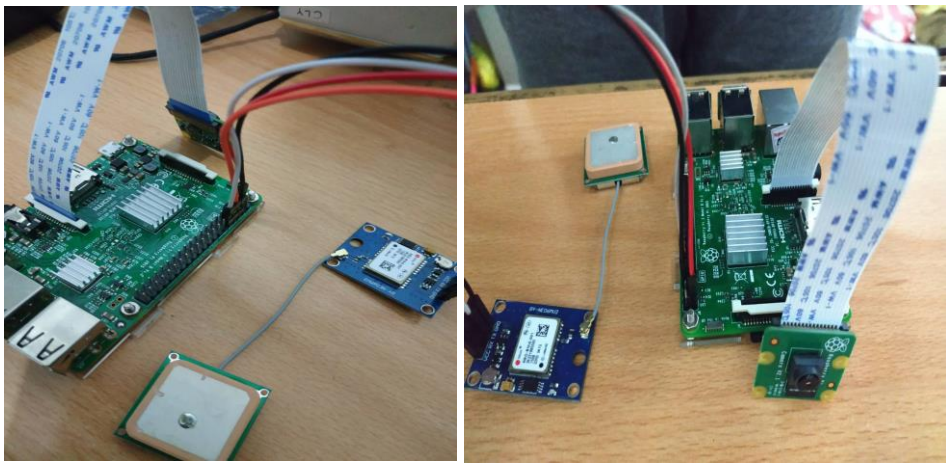
Pada tahap realisasi perangkat keras, ada tiga bagian yang akan direalisasikan yaitu realisasi PCB, realisasi Perakitan dan realisasi Pengkabelan

3.3.1.1 Realisasi PCB

Tahap realisasi PCB, pada alat yang kami buat adalah menggabungkan setiap komponen hardware yaitu Raspberry Pi, Camera Pi, dan modul GPS pada sebuah PCB dot matriks agar menjadi sebuah kesatuan. Akan tetapi sampai SKTA ini, realisasi PCB belum dapat direalisasikan.

3.3.1.2 Realisasi Perakitan

Realisasi perakitan pada alat yang dibuat sudah sesuai rancangan, akan tetapi belum ada *casing* yang mewadahi setiap komponen.



Gambar 3.3.1.2 Realisasi Perakitan

3.3.1.3 Realisasi Pengkabelan

Realisasi pengkabelan hanya pengkabelan sementara, belum dilakukan dengan baik dan fix.

3.3.2 Realisasi Perangkat Lunak

Realiasi perangkat lunak terdiri dari realisasi program pada hardware raspberry pi, user interface WEB dan realisasi database.

3.3.2.1 Realisasi Program

Realisasi Program pada sistem ini terdiri dari dua, yaitu pendeteksian lubang aspal dan pengukuran luas lubang aspal.

1. Pendeteksian Lubang

   Pendeteksian lubang dilakukan oleh raspberry pi dengan menggunakan Bahasa Python. Program ini berisi deteksi lubang dengan menggunakan Open CV library untuk membantu training data serta mengambil keputusan dari citra gambar yang akan diolah dan memberikan informasi atas gambar tersebut.

2. Pengukuran Luas lubang aspal

   Pengukuran luas lubang aspal dilakukan oleh raspberry pi dengan menggunakan metode dan algoritma yang sudah dirancang. Bahasa menggunakan Python untuk merancang programnya.

3.3.2.2 Realisasi Program GPS

```
import serial
from time import sleep
import webbrowser                                                    #import packa
import sys                                                                  #impo

def GPS_Info():
    global NMEA_buff
    global lat_in_degrees
    global long_in_degrees
    nmea_time = []
    nmea_latitude = []
    nmea_longitude = []
    nmea_time = NMEA_buff[0]
    nmea_latitude = NMEA_buff[1]                                                    #
    nmea_longitude = NMEA_buff[3]

    print("NMEA Time: ", nmea_time,'\n')
    print ("NMEA Latitude:", nmea_latitude,"NMEA Longitude:", nmea_longitude,


     lat = float(nmea_latitude)
     longi = float(nmea_longitude)

     lat_in_degrees = convert_to_degrees(lat)
     long_in_degrees = convert_to_degrees(longi)

#convert raw NMEA string into degree decimal format
def convert_to_degrees(raw_value):
     decimal_value = raw_value/100.00
     degrees = int(decimal_value)
     mm_mmmm = (decimal_value - int(decimal_value))/0.6
     position = degrees + mm_mmmm
     position = "%.4f" %(position)
     return position



gpgga_info = "$GPGGA,"
ser = serial.Serial ("/dev/ttyAMA0")
GPGGA_buffer = 0


gpgga_info = "$GPGGA,"
ser = serial.Serial ("/dev/ttyAMA0")
GPGGA_buffer = 0
NMEA_buff = 0
lat_in_degrees = 0
long_in_degrees = 0

try:
    while True:
        received_data = (str)(ser.readline())
        GPGGA_data_available = received_data.find(gpgga_info)
        if (GPGGA_data_available>0):
            GPGGA_buffer = received_data.split("$GPGGA,",1)[1]
            NMEA_buff = (GPGGA_buffer.split(','))                                   #st
            GPS_Info()

            print("lat in degrees:", lat_in_degrees," long in degree: ", long_i
            map_link = 'http://maps.google.com/?q=' + lat_in_degrees + ',' + lo
            print("<<<<<<<<press ctrl+c to plot location on google maps>>>>>>\n
```

```
            print("<<<<<<<<press ctrl+c to plot location on google maps>>>>>>\n
            print("-------------------------+-----------------------------------

except KeyboardInterrupt:
    webbrowser.open(map_link)        #open current position information in goog
    sys.exit(0)
```

3.3.2.2 Realisasi Capture Image

```
from time import sleep
from picamera import PiCamera

camera = PiCamera()
camera.resolution = (1260,720)
camera.start_preview()
# Camera warm-up time
sleep(1)
camera.capture('contoh1.jpg')
```

3.3.2.2 Realisasi Streaming Video

```
import io
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="""\
<html>
<head>
<title>picamera MJPEG streaming demo</title>
</head>
<body>
<h1>PiCamera MJPEG Streaming Demo</h1>
<img src="stream.mjpg" width="640" height="480" />
</body>
</html>
"""

class StreamingOutput(object):
```

```
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
```

```
class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace; bounda
            self.end_headers()
```

```
            self.send_header('Content-Type', 'multipart/x-mixed-replace; bounda
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
                    self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:
            self.send_error(404)
            self.end_headers()
```

```
                    self.client_address, str(e))
            else:
                self.send_error(404)
                self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    camera.start_recording(output, format='mjpeg')
    try:
        address = ('', 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()
```

3.3.2.2 Realisasi Recording Video

```
import picamera

camera = picamera.PiCamera()
camera.resolution = (1280,720)
camera.start_recording('latihan1.h264')
camera.wait_recording(100)
camera.stop_recording()
'
```

3.3.2.2 Realisasi Database

Realisasi Database menggunakan firebase realtime database, karena sistem yang dibuat bersifat *realtime* seperti pengiriman lokasi, gambar yang selalu *update* setiap waktu.

3.3.3 Realisasi Mekanik

Pada tahap ini, alat yang digunakan tidak menggunakan unsur mekanik.

3.3.3.2 Realisasi Kemasan Alat

Realisasi kemasan alat hanya berupa sementara, yaitu menggunakan box handphone yang kemudian ditaruh di bagian spion motor.

Untuk selanjutnya, akan dibuat sebuah casing untuk menyimpan hardware dengan aman.