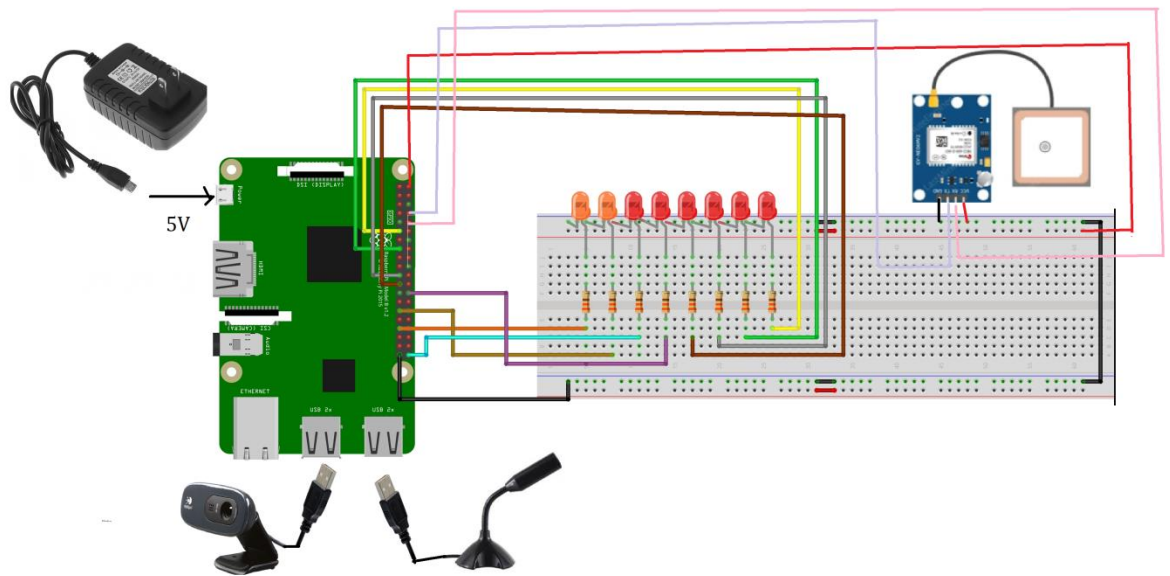


3.1. Realisasi

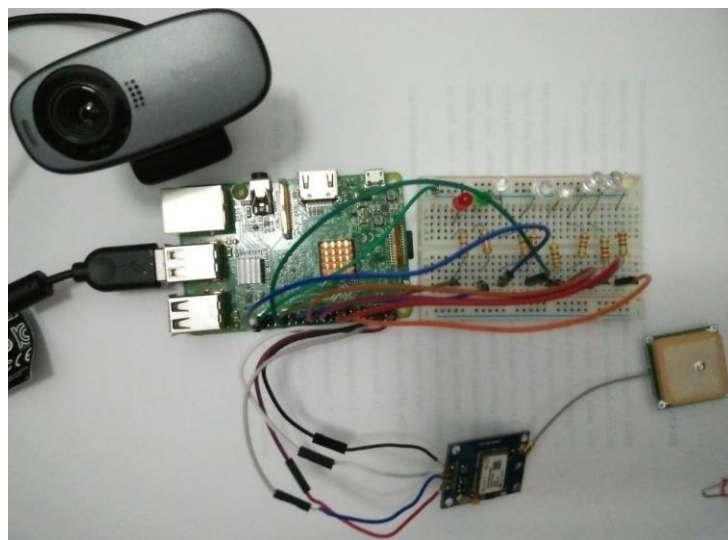
Pada sub bab realisasi ini dijelaskan mengenai proses perwujudan fisik yang sebelumnya telah dirancang.

3.3.1. Realisasi Perangkat Keras

3.3.1.1. Realisasi Pengkabelan



Gambar 3.7 Realisasi Pengkabelan



Gambar 3.8 Realisasi Perangkat Keras

Pada gambar 3.8 terdapat realisasi perakitan berbagai komponen yang digunakan. Rangkaian tersebut terdiri dari raspberry pi 3, webcam, USB microphone, gps module, LED dan resistor yang menjadi inti dari perangkat keras yang mendukung sistem *monitoring* posisi dan keadaan. Raspberrry pi sebagai mikrokontroler ini akan diprogram sedemikian rupa dan selengkapnya akan dipaparkan di realisasi perangkat lunak.

3.3.2. Realisasi Perangkat Lunak

Pada realisasi perangkat lunak terdapat 3 bagian yang direalisasikan, yaitu program sistem *monitoring* keadaan dengan kamera, sistem monitoring *posisi* dengan GPS, dan sistem komunikasi text dan suara. Tetapi untuk saat ini program yang telah direalisasikan barulah proogram pengambilan foto dan video dengan perintah suara dan program pengambilan data dengan GPS .

3.3.2.1. Realisasi Program Pengambilan foto dan video dengan perintah suara

Pada program reakisasi pengambilan foto dan video digunakan pemrograman python dan untuk menampilkan gambar visual digunakan *library openCV*. Dalam program ini Input telah diatur didapatkan dari suara dengan perintah “ambil video” atau “ambil foto”. Input sinyal suara kemudian dibuat menjadi text dengan google *speech* API seperti yang ditunjukkan gambar 3.9 berikut

```
while True :
    input_state = False
    print(input_state)
    if input_state == False:
        print ('Button Pressed')

    AUDIO_FILE = path.join(path.dirname(path.realpath(__file__)), "test.wav")
    call(["arecord" , "-D" , "plughw:1,0" , "-d" , "5" ,"test.wav"])
    r = sr.Recognizer()

    with sr.AudioFile(AUDIO_FILE) as source:
        audio = r.record(source) # read the entire audio file

    try:
        # for testing purposes, we're just using the default API key
        txt = r.recognize_google(audio, language='id-ID')
        print("User Say: " + txt )
```

Gambar 3.9 Program Pengambilan Input Suara

Setelah penyandang disabilitas mengucapkan suatu perintah baik perintah yang telah terdaftar maupun perintah yang tidak terdaftar, perintah tersebut akan tetap ditampilkan di layar. Hanya saja jika perintah yang diucapkan tidak terdaftar maka program akan meminta perintah kembali. Jika kata atau perintah yang diucapkan adalah “ambil video” maka program pengambilan dan penyimpanan video seperti yang ditunjukkan gambar 3.10 akan dijalankan. Dan apabila perintah yang diucapkan adalah “ambil foto” maka program pengambilan dan penyimpanan foto seperti gambar 3.11 akan dijalankan.

```
def videoo():
    while True :
        check, frame = video.read()
        cv2.imshow("Taking Video", frame)
        out.write(frame)
        print('video berhasil diambil')

        if "ambil" in txt:
            if "video" in txt:
                video=cv2.VideoCapture(0)
                fourcc = cv2.VideoWriter_fourcc(*'XVID')
                out = cv2.VideoWriter((str(time.strftime("%Y_%m_%d_%H_"))+'.avi',fourcc, 30.0, (640,480))
                videoo()
                turnoff(kesatu)
                turnoff(kedua)
                turnoff(ketiga)
                turnoff(keempat)
                turnon(kelima)
                turnoff(keenam)
                turnoff(ketujuh)
                turnon(kedelapan)
```

Gambar 3.10 Program pengambilan dan penyimpanan Video

```
def foto():
    check, frame = video.read()
    cv2.imshow("Taking foto", frame)
    cv2.imwrite('foto'+(str(time.strftime("%Y_%m_%d_%H_%M"))+'.jpg', frame)
    cv2.imshow("Hasil Capture", frame)
    print('Foto Berhasil Diambil')
    if "ambil" in txt:
        if "foto" in txt:
            video=cv2.VideoCapture(0)
            foto()
            turnon(kesatu)
            turnon(kedua)
            turnon(ketiga)
            turnon(keempat)
            turnoff(kelima)
            turnoff(keenam)
            turnoff(ketujuh)
            turnon(kedelapan)
            continue
```

Gambar 3.11 Program Pengambilan dan Penyimpanan Fotot

3.3.2.2. Realisasi Program Pengambilan Data GPS

Sebelum memulai fitur monitorin posisi, pastikan terlebih dahulu bahwa perangkat raspberry pi sudah bisa mendapatkan data *longitude* dan *latitude* dari GPS modul. Untuk mengecek apakah GPS sudah dapat menerima data dari satelit, maka dimasukan perintah “cgps -s”. Jika sudah berhasil maka data akan ditampilkan di layar.

Untuk mendapatkan data *longitude* dan *latitude* diambilah data dari NMEA dengan program seperti pada gambar 3.11 berikut.

```
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]           #extract time from GPGGA string
    nmea_latitude = NMEA_buff[1]        #extract latitude from GPGGA string
    nmea_longitude = NMEA_buff[3]       #extract longitude from GPGGA string

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

    lat = float(nmea_latitude)          #convert string into float for calculation
    longi = float(nmea_longitude)        #convertr string into float for calculation

    lat_in_degrees = convert_to_degrees(lat) #get latitude in degree decimal format
    long_in_degrees = convert_to_degrees(longi) #get longitude in degree decimal format
```

Gambar 3.11 Program pengambilan data *longitude* dan *latitude* dari GPS

Data tersebut selanjutnya dikonversi menjadi format derajat agardapat di plotting pada *google maps*. program pengkonversian dan proses plotting dapat dilihat pada gambar 3.12 dan 3.13

```
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
```

Gambar 3.12 Program Konversi Menjadi Format Derajat

```

try:
    while True:
        received_data = (str)(ser.readline())          #read NMEA string received
        GPGGA_data_available = received_data.find(gpgga_info) #check for NMEA GPGGA string
        if (GPGGA_data_available>0):
            GPGGA_buffer = received_data.split("$GPGGA,",1)[1] #store data coming after "$GPGGA," string
            NMEA_buff = (GPGGA_buffer.split(','))              #store comma separated data in buffer
            GPS_Info()                                         #get time, latitude, longitude

            print("lat in degrees:", lat_in_degrees," long in degree: ", long_in_degrees, '\n')
            map_link = 'http://maps.google.com/?q=' + lat_in_degrees + ',' + long_in_degrees #create link to plot location on Google map
            print("<<<<<<<press ctrl+c to plot location on google maps>>>>>>>\n")           #press ctrl+c to plot on map and exit
            print("-----\n")

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

```

Gambar 3.13 Program Ploting pada Google Maps