

## BAB III METODOLOGI PENELITIAN

### 3.1 Alat dan Bahan .

#### 3.1.1 Alat

Sebelum membuat Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry ada beberapa peralatan yang harus disiapkan. Daftar peralatan yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.1.

**Tabel 3.1 Alat Yang Dibutuhkan**

No	Nama Alat	Spesifikasi	Fungsi	Jumlah
1	Komputer/ laptop	Window 7-10 32/64bit	Untuk membuat sebuah aplikasi yang akan dipakai diperangkat keras dan perangkat lunak.	1 unit
2	Multitester	Analog/Digital	Digunakan untuk mengukur tegangan (ACV-DCV), dan kuat arus (mA- $\mu$ A).	1 buah
3	Obeng	Obeng (+) dan (-)	Untuk merangkai alat.	1 buah
4	Solder	-	Untuk menempelkan timah ke komponen.	1 buah
5	Bor pcb	-	Untuk membuat lobang baut atau komponen.	1 buah
6	Tang Potong	-	Untuk memotong kabel dan kaki komponen.	1 buah

#### 3.1.2 Bahan

Sebelum membuat Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry ada beberapa bahan yang harus disiapkan. Daftar komponen yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.2.

**Tabel 3.2. Komponen Yang Dibutuhkan**

No	Nama Bahan	Sepesifikasi	Fungsi	Jumlah
1	Respiberry	Pi3	Sebagai proses perintah yang akan di jalankan.	1 unit
2	<i>Sensor ultrasonik</i>	-	Digunakan sebagai pembaca jarak.	1 unit
3	<i>Motor servo</i>		Digunakan sebagai simulasi pengereman.	1 unit
5	<i>camera</i>	USB	Digunakan sebagai pengambil gambar.	2 buah
6	<i>Memori</i>	SD Card 16GB	Digunakan sebagai penyimpan program respiberry (OS).	1 buah
7	<i>Kabel Power</i>	1	Digunakan sebagai penghantar arus listrik	1 Buah
8	<i>Jumper</i>	-	Digunakan sebagai penghubung/menjumper seluruh komponen.	30 Buah

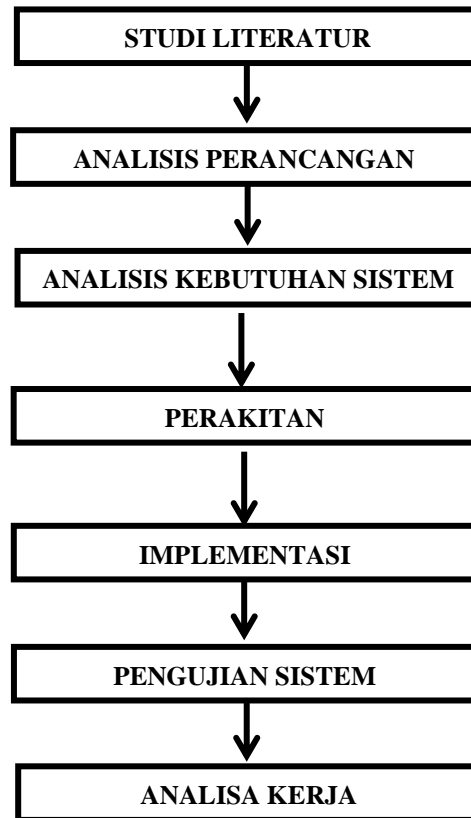
### 3.1.3 Software

Sebelum membuat Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry ada beberapa peralatan yang harus disiapkan. Daftar Software yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.3.

**Tabel 3.3. Daftar Software Yang Digunakan**

No	Nama	Spesifikasi	Fungsi
1	Python	3.7	Membuat program yang akan di download perangkat Arduino
2	<i>Opencv</i>		Digunakan untuk menjalankan program camera.

Bab ini akan menjelaskan langkah-langkah penelitian yang akan dilakukan dalam Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry. Alur penelitian yang digunakan seperti pada gambar 3.1.



**Gambar 3.1. Alur Penelitian**

### **3.2 Studi Literatur**

Pada metode ini penulis mencari bahan penulisan skripsi yang diperoleh dari buku, jurnal dan *website* yang terkait dengan pembuatan Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry.

- **Analisa Perancangan Sistem**

Dalam perancangan sistem Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry meliputi perancangan perangkat keras dan perangkat lunak. Penjelasan dari rancangan sistem berupa diagram blok.

- **Analisa Kebutuhan Sistem**

Analisa kebutuhan meliputi alat dan bahan yang diperlukan dalam Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry merupakan perangkat keras dan software untuk melakukan penelitian.

- **Perakitan**

Perakitan merupakan tahap terakhir dilakukan untuk mengetahui apakah rangkaian keseluruhan yang telah dibuat dapat berkerja dengan baik. Sehingga dapat dilakukan implementasi sistem.

- **Implementasi Perangkat**

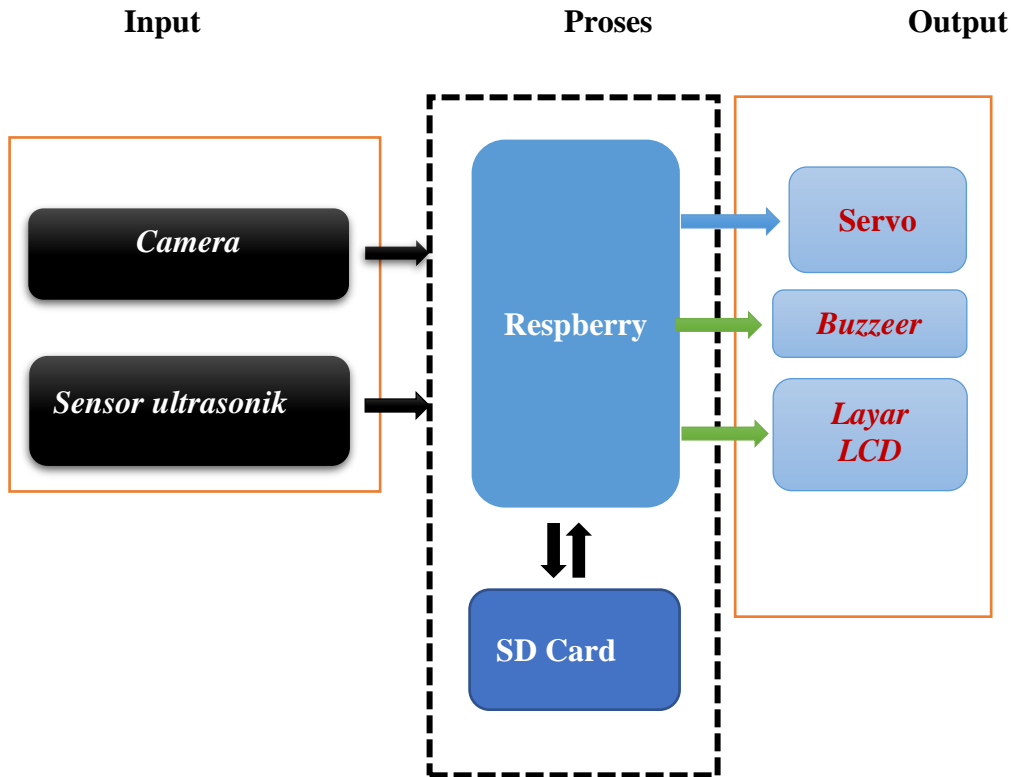
Setelah mengumpulkan alat dan bahan yang diperlukan, langkah selanjutnya adalah melakukan implementasi perangkat. Pada tahapan ini rancangan yang telah dibuat akan diimplementasikan menjadi sistem yang sesungguhnya.

- **Pengujian Sistem**

Uji coba sistem Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry dilakukan untuk memastikan bahwa alat yang dibuat mengetahui kerja bekerja sesuai dengan rancangan, serta untuk memastikan bahwa tidak terjadi kesalahan pada alat.

### **3.3 Analisa Perancangan Sistem**

Perancangan sistem merupakan suatu hal yang dilakukan untuk mempermudah proses pembuatan alat. Konsep Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry digambarkan pada diagram blok dapat dilihat pada gambar 3.2 Blok diagram menjelaskan gambaran umum mengenai cara kerja dari sistem Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry yang akan dibuat.



**Gambar 3.2. Blok Diagram Sistem**

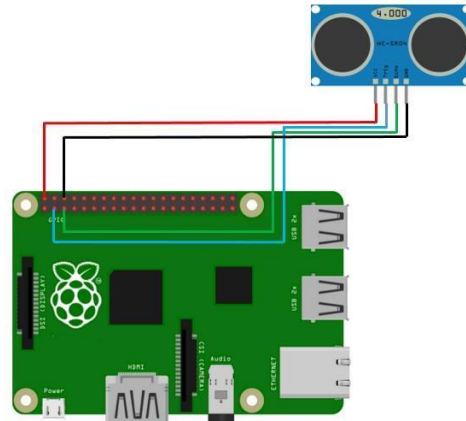
Dari gambar blok diagram sistem dapat diketahui Dari gambar blok diagram di atas dapat diketahui cara kerja alat yaitu Inputan dari sensor ultrasonic dan kamera akan mendeteksi batas parkir dibagian belakang mobil. Sistem kerja alat ini yaitu saat sensor ultrasonic mendeteksi batas parkir dalam radius 50cm motor akan bergerak menekan rem. layar LCD digunakan sebagai tampilan dari camera dan jarak.

### **3.3.1 Perancangan Perangkat Keras**

Perancangan menjadi bagian yang sangat penting dilakukan dalam pembuatan suatu alat karena dengan merancang terlebih dahulu dengan komponen yang tepat akan mengurangi berlebihnya pembelian komponen dan kerja alat sesuai dengan yang diinginkan. Untuk menghindari kerusakan komponen perlu dipahami juga akan karakteristik dari komponen-komponen tersebut.

#### **3.3.1.1 Rangkaian Sensor Ultrasonik**

Rangkaian sensor ultrasonik digunakan sebagai inputan untuk mengukur jarak mundur kendaraan yang akan diproses oleh raspberry gambar rangkaian sensor ultrasonik dapat dilihat seperti pada gambar 3.3



**Gambar 3.3 Rangkaian Sensor Ultrasonik**

Pada rangkaian sensor ultrasonik hanya beberapa kaki yang dihubungkan ke pin GPIO raspberry agar hasil proses pada raspberry dapat memberikan output pengukuran jarak. Penggunaan PIN raspberry dan sensor ultrasonik dapat diketahui yaitu pin GPIO 3 dan GPIO 6 pada raspberry akan dihubungkan ke pin trig dan echo pada pin sensor ultrasonik. Potongan program (*script*) sensor ultrasonik sebagai berikut.

```
def run(self):
    self.runningState = True

    timeoutBeep = time.time()
    while self.runningState:
        GPIO.output(GPIO_TRIGGER, True)

        time.sleep(0.00001)
        GPIO.output(GPIO_TRIGGER, False)

        StartTime = time.time()
        StopTime = time.time()

        while GPIO.input(GPIO_ECHO) == 0:
            StartTime = time.time()

        while GPIO.input(GPIO_ECHO) == 1:
            StopTime = time.time()

        TimeElapsed = StopTime - StartTime

        jarak = (TimeElapsed * 17150)
        jarak = round(jarak, 2)

        if self.runningState:
            self.gui.updateJarak(jarak)

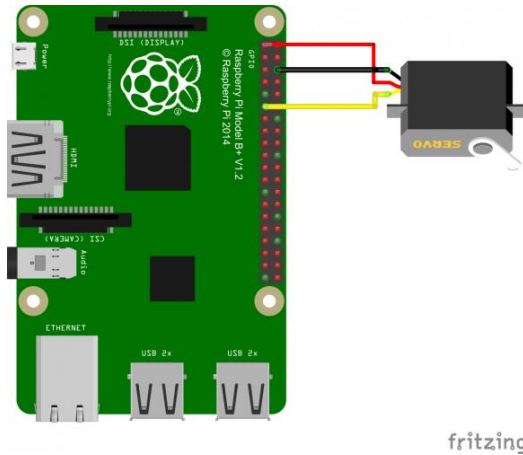
        if timeoutBeep < time.time():
            if jarak < 50:
                # os.system('omxplayer -o local beep.mp3')
                # os.system('mpg321 beep.mp3 &')
                p.ChangeDutyCycle(100)

                timeoutBeep = time.time() + 1
            elif jarak < 100:
                # os.system('omxplayer -o local beep.mp3')
                # os.system('mpg321 beep.mp3 &')
                timeoutBeep = time.time() + 1
```

**Gambar 3.4 Script Program Sensor Ultrasonik**

### 3.3.1.2 Rangkaian Motor Servo

Rangkaian *motor servo* digunakan sebagai *output* pengereman. Sistem kerja dari motor servo akan berputar 90°. Gambar rangkaian motor servo dan tata letak dapat dilihat seperti pada gambar 3.5.



**Gambar 3.5 Rangkaian Motor Servo**

Pada rangkaian *motor servo* hanya beberapa kaki yang dihubungkan ke pin GPIO raspberry agar hasil proses pada raspberry dapat memutar motor servo dari 0° sampai 90°. Penjelasan penggunaan PIN raspberry dan motor servo yaitu pin GPIO17 dihubungkan pada pin out motor servo. Potongan program (*script*) motor servo sebagai berikut.

```
*pinger.py - E:\jarak\pinger.py (2.7.14)*
File Edit Format Run Options Window Help
import RPi.GPIO as GPIO
import time

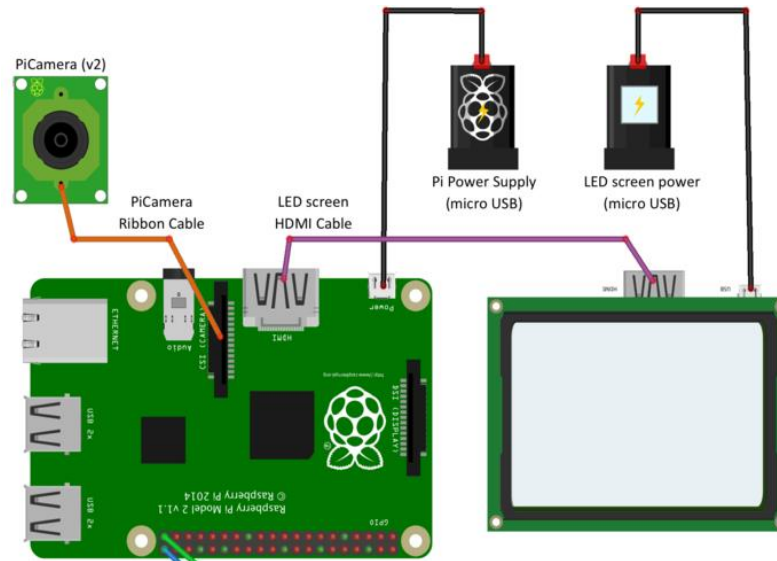
servoPIN = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50) # GPIO 17 for PWM with 50Hz
p.start(2.5) # Initialization
try:
    while True:
        p.ChangeDutyCycle(5)
        time.sleep(0.5)
        p.ChangeDutyCycle(7.5)
        time.sleep(0.5)
except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
```

**Gambar 3.6 Script Program Motor Servo**

### 3.3.1.3 Rangkaian Kamera dan LCD

Rangkaian Kamera digunakan sebagai *input* yang akan diproses oleh raspberry sehingga akan melakukan pengambilan vidio yang akan ditampilkan pada layar *LCD*. Gambar rangkaian Kamera dan tata letak dapat dilihat seperti pada gambar 3.7.



**Gambar 3.7 Rangkaian Kamera dan Layar LCD**

Pada rangkaian Kamera dan layar LCD dapat diketahui jika usb camera akan dimasukan pada slot usb dan layar lcd menggunakan HDMI cable. Potongan program (*script*) camera dan layar LCD sebagai berikut

```
class ThreadKamera(Thread):
    def __init__(self, group=None, target=None, name=None, args=(), kwargs=None, verbose=None):
        super(ThreadKamera, self).__init__(group=group, target=target, name=name)

        self.gui = args[0]
        self.runningState = False

    def run(self):
        self.runningState = True
        self.camera = cv2.VideoCapture(0)
        self.camera.open(0)

        if not (self.camera.isOpened()):
            print("Could not open video device")
        else:
            timeSuaraPeringatan = time.time()
            while self.runningState:
                ret, frame = self.camera.read()
                if ret:
                    cv2image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)

                    if self.runningState:
                        resized = cv2.resize(cv2image, (320, 240), interpolation=cv2.INTER_AREA)
                        img = Image.fromarray(resized)
                        self.gui.updateImage(img)

            self.camera.release()

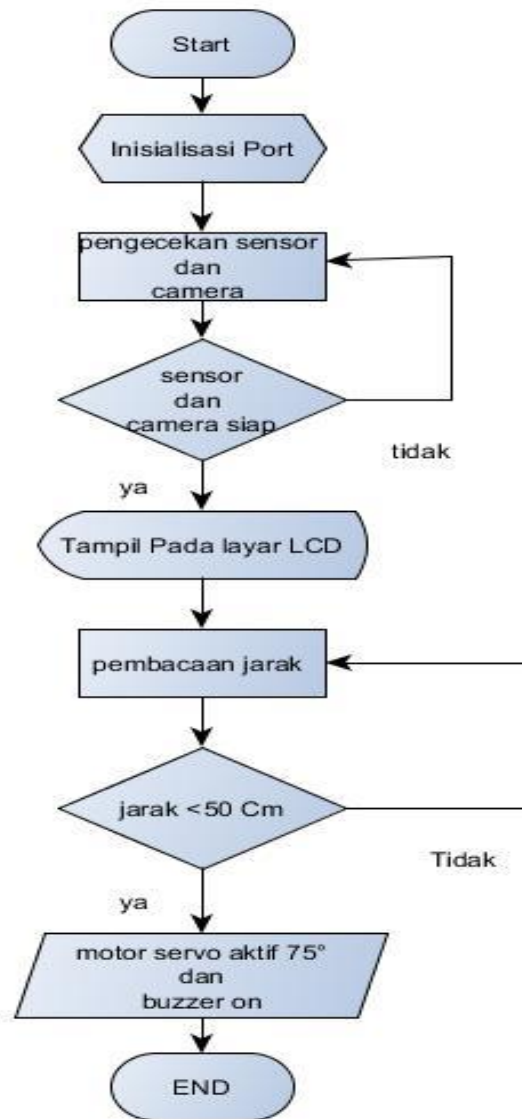
    def stop(self):
        self.runningState = False
```

**Gambar 3.8 Script Program Camera dan Layar LCD**

### 3.3.2 Perancangan Perangkat Lunak



Perancangan perangkat lunak dibuat dari pembuatan *flowchart* untuk pembuatan pada *hardware*. Pada gambar 3.9 akan ditampilkan *flowchart* dari program yang akan dibuat dalam penelitian ini.



**Gambar 3.9 Flowchart Sistem RFID**

Di bawah ini merupakan penjelasan dari *flowchart* program pada gambar 3.9: Inisialisasi proses pembacaan pin pada raspberry jika pengecekan sensor dan camera jika sensor dan camera siap maka akan tampil pada layar LCD kemudian jika jarak sensor <50 cm maka motor servo akan aktif berputar 75° dan buzzer akan aktif. End

### 3.4 Implementasi

Setelah mengumpulkan alat dan bahan, langkah selanjutnya adalah melakukan implementasi rancangan alat yang telah dibuat. Pada tahap ini hasil rancangan yang telah dibuat akan diimplementasikan untuk menjadi sistem yang sesungguhnya. Implementasi pada penelitian ini dibagi menjadi dua bagian, yaitu: Implementasi perangkat keras dan Implementasi perangkat lunak. Implementasi perangkat keras merupakan tahap terakhir dari perancangan sistem yang dilakukan dalam tahap ini seluruh komponen dipasang sesuai dengan sistem yang telah dibuat.

### **3.4.1 Implementasi Perangkat Keras**

Realisasi perangkat keras merupakan tahap terakhir dari perancangan yang telah dilakukan. Dalam tahap ini seluruh komponen dipasang sesuai dengan sistem yang telah dibuat

### **3.4.2 Implementasi Perangkat Lunak.**

#### **3.4.2.1 Persiapan Upload OS Rasbian**

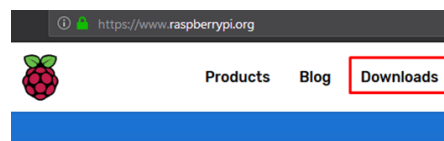
Papan Raspberry Pi Zero W ini agar dapat dioperasikan maka memerlukan sebuah operating system yang tertanam didalamnya. Proses menanamkan OS ini dilakukan dengan mengupload file OS yang digunakan pada sebuah media penyimpanan micro SD dengan kapasitas minimal 8 GB.

#### **1. Upload OS Rasbian**

Proses menanamkan OS Rasbian ini kedalam media perangkat penyimpanan micro sd card dilakukan menggunakan aplikasi etcher dengan langkah-langkah sebagai berikut :

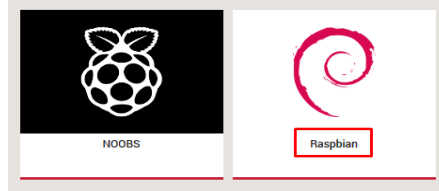
#### **3.4.2.2 Download file OS Rasbian**

1. Masuk ke laman [www.raspberrypi.org](http://www.raspberrypi.org) dan masuk pada menu Downloads dapat dilihat pada gambar 3.10



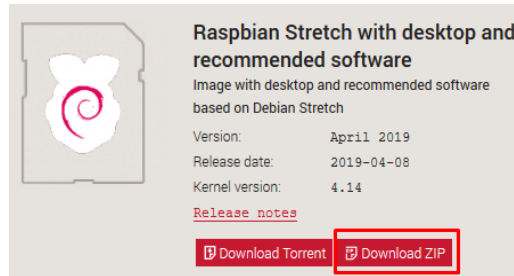
**Gambar 3.10. Halaman Raspberry Pi**

2. Pilih OS Rasbian sebagai pilihan OS yang akan digunakan



**Gambar 3.11. Halaman Pemilihan File OS Raspbian**

3. Tekan tombol Download Zip untuk memulai mendownload OS Raspbian.



**Gambar 3.12 Halaman Download File OS Raspbian**

### 3.5 Langkah – Langkah Pengaktifan Camera Pada Raspiberry Pi

1. Pertama kita cek python3 pada Raspberry pi dengan cara berikut

```
$ python3
```

Selanjutnya kita install pip3, jalankan pada terminal,

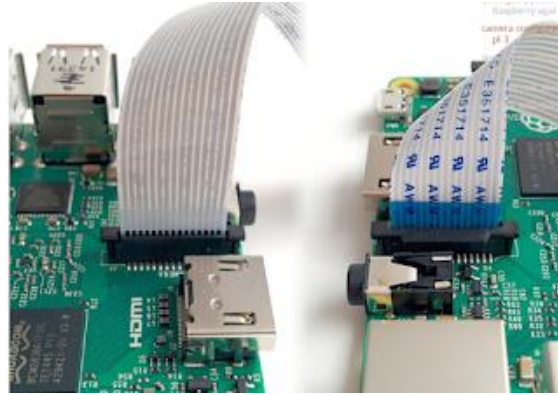
```
$ sudo python3-pip
```

Selanjutnya kita install OpenCV, numpy dengan cara berikut,

```
$ sudo apt-get install python-opencv  
$ sudo apt-get install python-numpy
```

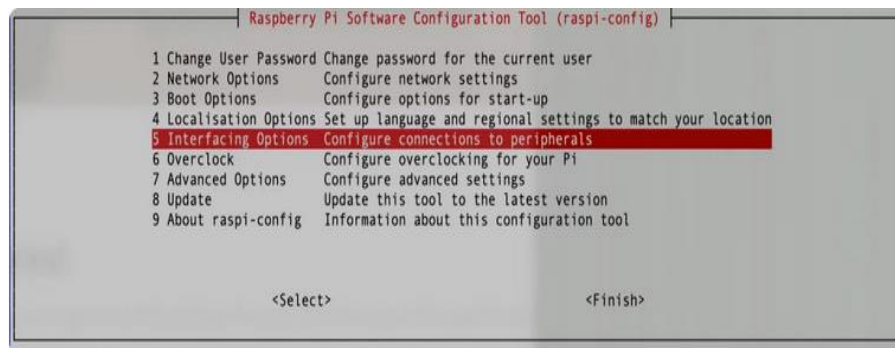
Jika sudah selesai menginstall openCV dan numpy langkah selanjutnya menyalakan camera yaitu dengan menghubungkan modul camera v2 ke port Raspberry PI, dan memastikan perangkat lunak diaktifkan.

2. Cari port kamera dan menghubungkan kamera:

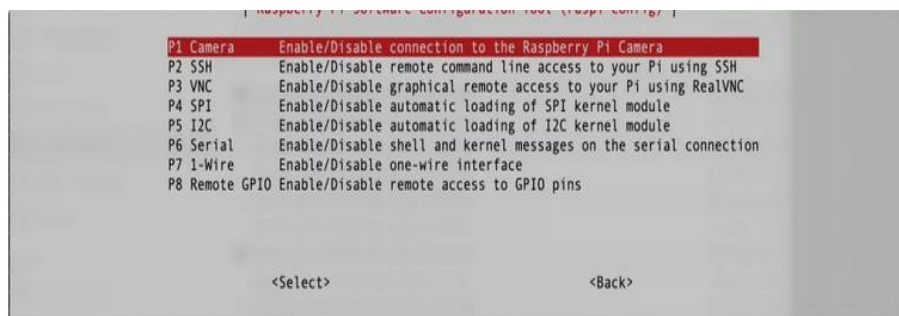


3. langkah selanjutnya harus mengaktifkan kamera terlebih dahulu dari konfigurasi alat  
`sudo raspi-config`

### Kemudian Pilih Interfacing



### Kemudian pilih P1 Camera



4. Selanjutnya mengetes camera dengan membuka file baru dan simpan sebagai **camera.py** di direktori **/home/pi**
5. Masukkan script berikut:

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.start_preview()
```

```
sleep(10)
camera.stop_preview()
```

6. Simpan dengan **Ctrl + S** dan jalankan dengan **F5**

### 3.6 Berikut Ini Penjelasan Dari Scrip Program

1. Program Buzzer yang digunakan sebagai alarm jika jarak sensor ultrasonik sudah mendekati benda dibelakang mobil.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO_BUZZER = 4
GPIO.setup(GPIO_BUZZER, GPIO.OUT)
GPIO.output(GPIO_BUZZER, False)
while True:
    GPIO.output(GPIO_BUZZER, True)
    time.sleep(0.5)
    GPIO.output(GPIO_BUZZER, False)
    time.sleep(0.5)
```

2. Program Sensor Ultrasonik yang digunakan sebagai pendeteksi kendaraan dibelakang mobil.

```
import os
import time
import RPi.GPIO as GPIO

judul = "Judul"
GPIO_TRIGGER = 22
GPIO_ECHO = 23
GPIO_TRIGGER1 = 17
GPIO_ECHO1 = 18

GPIO.setmode(GPIO.BCM)

GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
GPIO.setup(GPIO_TRIGGER1, GPIO.OUT)
GPIO.setup(GPIO_ECHO1, GPIO.IN)

GPIO.output(GPIO_TRIGGER, False)
GPIO.output(GPIO_TRIGGER1, False)

self.runningState = False
```

```

def run(self):
    self.runningState = True

    timeoutBeep = time.time()
    while self.runningState:
        GPIO.output(GPIO_TRIGGER, True)

        time.sleep(0.00001)
        GPIO.output(GPIO_TRIGGER, False)

        StartTime = time.time()
        StopTime = time.time()

        while GPIO.input(GPIO_ECHO) == 0:
            StartTime = time.time()

        while GPIO.input(GPIO_ECHO) == 1:
            StopTime = time.time()

        TimeElapsed = StopTime - StartTime

        jarak = (TimeElapsed * 17150)
        jarak = round(jarak, 2)

        GPIO.output(GPIO_TRIGGER1, True)

        time.sleep(0.00001)
        GPIO.output(GPIO_TRIGGER1, False)

        StartTime = time.time()
        StopTime = time.time()

        while GPIO.input(GPIO_ECHO1) == 0:
            StartTime = time.time()

        while GPIO.input(GPIO_ECHO1) == 1:
            StopTime = time.time()

        TimeElapsed = StopTime - StartTime

        jarak1 = (TimeElapsed * 17150)
        jarak1 = round(jarak1, 2)
        if timeoutBeep < time.time():
            if jarak < 20:
                GPIO.output(GPIO_BUZZER, True)

```

```
if jarak > 20:
    GPIO.output(GPIO_BUZZER, False)
if jarak < 10 or jarak1 < 10:
    # os.system('omxplayer -o local beep.mp3')
    p.ChangeDutyCycle(5)
    time.sleep(2)
    p.ChangeDutyCycle(10)
```

3. Program motor servo yang digunakan sebagai pengereman jika jarak pada sensor sudah mendekati adanya benda atau penghalang dibelakang mobil.

```
servoPIN = 27
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)
p = GPIO.PWM(servoPIN, 50) # GPIO 17 for PWM with 50Hz
p.start(2.5) # Initialization
p.ChangeDutyCycle(5)
time.sleep(2)
p.ChangeDutyCycle(10)
```

### 3.7 Pengujian Sistem

Setelah perancangan *hardware* dan *software* selesai, maka yang dilakukan adalah *running* program, pengujian tiap-tiap rangkaian apakah sudah sesuai dengan yang diinginkan atau belum. Pengujian dilakukan pada bagian-bagian seperti pengujian respon, jangkauan sistem dan rangkaian keseluruhan pada sistem ini..

#### 3.7.1 Pengujian Sistem Sensor Ultrasonik

Pengujian ultrasonik bertujuan untuk mengetahui ketika sensor berkerja mengukur jarak apakah sensor tidak mengalami error, dan akan digunakan meteran sebagai perbandingan agar peneliti dapat mengetahui selisi atau ke erroran sistem sensor.

#### 3.7.2 Pengujian Motor Servo

Pengujian rangkaian motor servo bertujuan untuk mengetahui apakah motor servo dapat bekerja memutar dari sudut 0° sampai 75° dan kebalikannya dari 75° ke 0°. Agar mengetahui apakah rangkaian motor servo telah berkerja sesuai dengan program arduino yang telah dibuat. Dalam melakukan ujicoba motor servo peneliti menggunakan peggaris busur yang digunakan sebagai perbandingan saat motor servo berputar..

### **3.5.3 Pengujian Kamera**

Pengujian kamera untuk memastikan apakah program yang dibuat sudah dapat memutar video yang akan ditampilkan pada layar LCD.

### **3.5.4 Pengujian Sistem Keseluruhan**

Pengujian sistem secara keseluruhan bertujuan untuk memastikan semua komponen dapat berjalan dengan sempurna. Mulai dari sensor ultrasonik, motor servo, camera blok dan program yang mengatur jalannya sistem keseluruhan.

### **3.8 Analisis Kerja**

Untuk analisa kerja, dilakukan bersama pada saat melakukan uji coba alat yang bertujuan untuk mengetahui kerja alat tersebut. Selain itu yang akan dianalisa adalah jarak, respon dalam untuk inputan pada sistem Sistem Pendeteksi Dan Penampil Jarak Otomatis Menggunakan Sensor Parkir Mundur Mobil Berbasis Raspberry. Berdasarkan hasil pengujian sistem yang telah di dapat akan dianalisis untuk memastikan bahwa sistem yang telah dibuat sesuai dengan harapan.