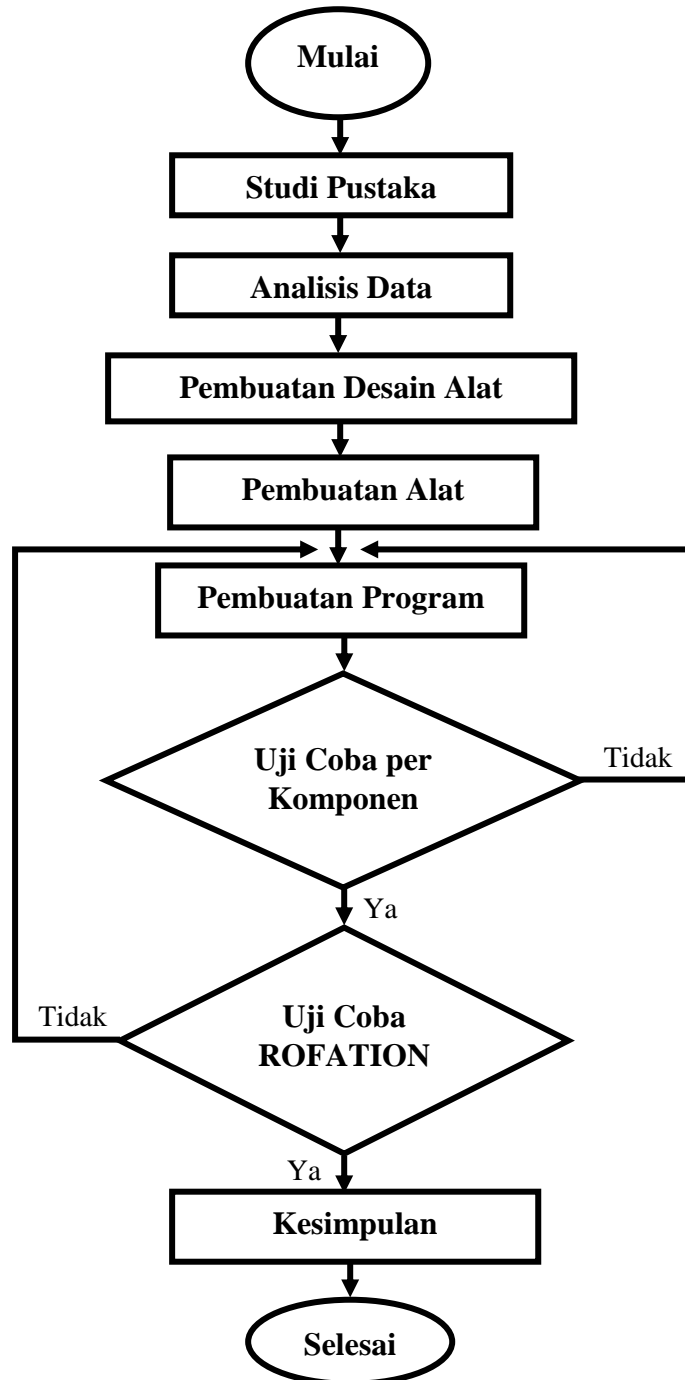


BAB III METODE PENELITIAN

3.1 Diagram Alir Penelitian

Pada penelitian ini terdapat diagram alir untuk pembuatan ROFATION terlihat seperti gambar dibawah ini:



Gambar 3. 1 Diagram Alir Penelitian

3.1.1 Penjelasan Diagram Alir Penelitian

a. Mulai

Pada proses ini penulis memulai untuk mengerjakan penelitian mengenai ROFATION.

b. Studi Pustaka

Pada tahap ini akan dilakukan pengumpulan data-data dan informasi dari buku maupun dari internet yang berkaitan dengan bahan dan berbagai komponen yang digunakan dalam ROFATION.

c. Analisis Data

Data-data dan informasi yang diperoleh dari pengumpulan informasi kemudian dilakukan proses analisis data yang berfungsi untuk menentukan bentuk, bahan, dan komponen yang akan digunakan pada ROFATION.

d. Pembuatan Desain Alat

Dalam tahap ini dilakukan pembuatan desain alat yang bertujuan untuk langkah selanjutnya yaitu pembuatan alat sehingga dapat mengetahui gambar sesungguhnya dari alat yang akan dibuat dan mengetahui komponen-komponen yang dapat saling terhubung.

e. Pembuatan Alat

Setelah dilakukan pembuatan desain alat, bahan – bahan dan komponen – komponen yang telah terkumpul akan dilakukan proses perancangan alat sesuai dengan desain yang telah dibuat.

f. Pembuatan Program

Pada pembuatan program ini, dimana alat yang sudah jadi akan dimasukkan sebuah program untuk menjalankan kerja dari alat ROFATION.

g. Pengujian per Bagian

Pada tahan ini, setelah dilakukan pembuatan program, selanjutnya komponen – komponen pada alat akan diuji coba satu per satu untuk memastikan komponen dapat bekerja dengan baik jika terjadi kesalahan dapat diperbaiki sebelum menguji alat secara keseluruhan.

h. Pengujian Keseluruhan

Pengujian keseluruhan dilakukan setelah semua bagian komponen bekerja dengan baik, selanjutnya rangkaian digabungkan baik alat maupun program jadi satu kesatuan dan dapat dilakukan uji coba, namun apabila masih terjadi kesalahan atau kerusakan sistem maka perlu dilakukan perbaikan alat maupun programnya sehingga ROFATION dapat bekerja dengan baik.

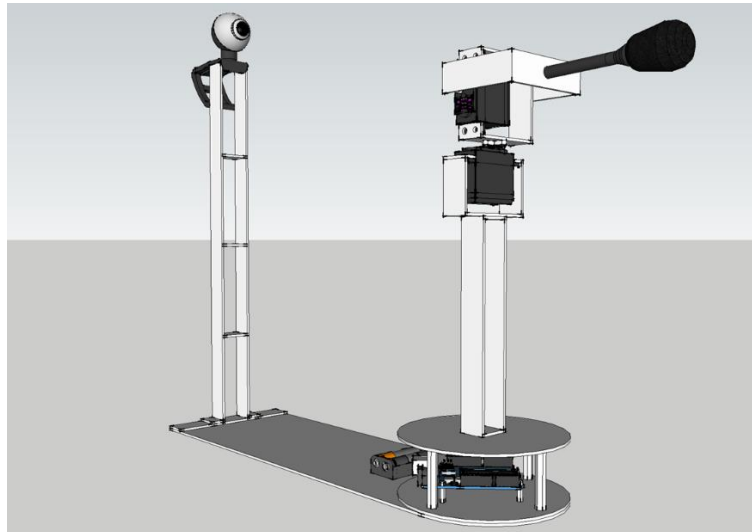
i. Selesai.

Seluruh kegiatan penelitian telah selesai dilakukan baik pembuatan alat, percobaan, dan analisis data yang dilakukan.

3.2 Rancangan Alat (*Hardware & Software*)

Pada tahap perancangan alat ROFATION memiliki 2 bagian dasar, yaitu bagian perangkat keras (*hardware*) dan bagian perangkat lunak (*software*). Pada bagian perangkat keras (*hardware*) terdapat dua buah rancangan yaitu rancangan motor *servo 2-axis* yang terhubung dengan mikrofon dan terdapat rancangan kamera terdapat di belakang motor *servo 2-axis*. Sensor yang digunakan pada ROFATION yaitu kamera *Logitech C270* dimana kamera ini dapat mendeteksi wajah lalu mengirimkan koordinat x dan y .

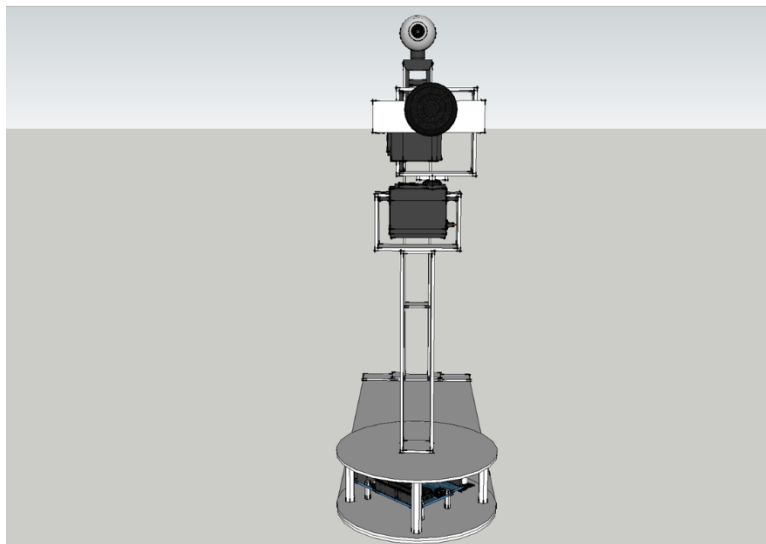
3.2.1 Perancangan Perangkat Keras (*Hardware*)



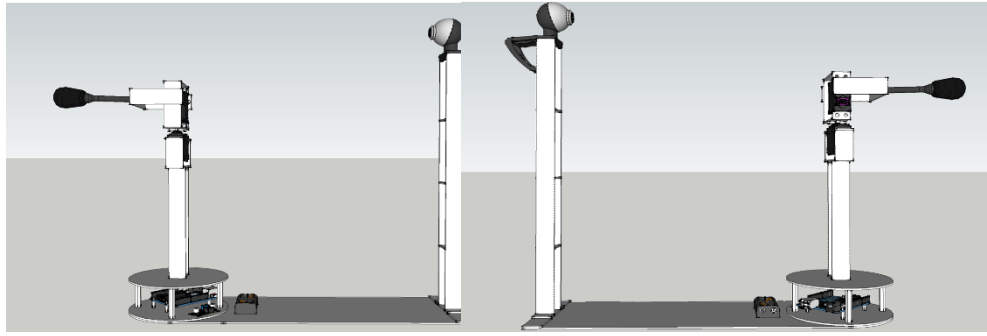
Gambar 3. 2 Desain alat ROFATION

3.2.1.1 Perancangan Desain

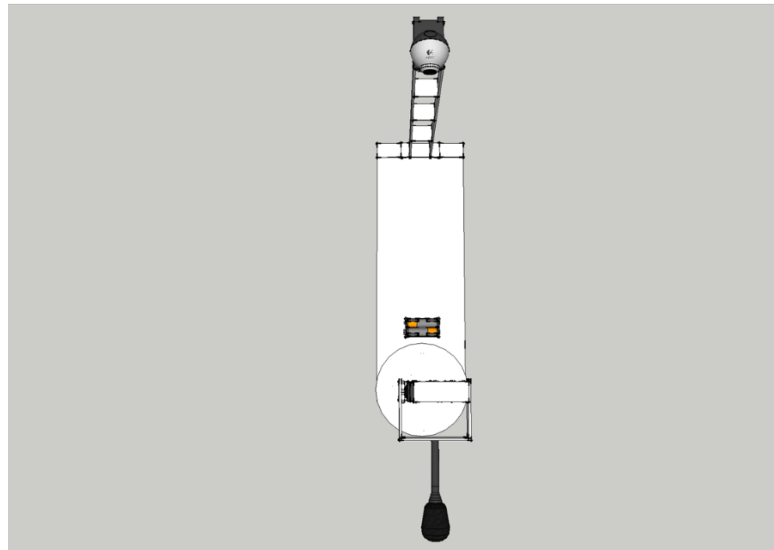
Pada perancangan ROFATION ini, dilakukan skenario perancangan menggunakan sebuah *software* yaitu *SketcUp 2016* dimana *software* ini dapat membuat desain 3D bentuk dari ROFATION sehingga dapat memudahkan dalam pembuatan alatnya. ROFATION dibuat menggunakan 2 buah motor *servo* yang bergerak *2-axis* yaitu x dan y . Servo bergerak berdasarkan objek wajah yang terdeteksi yang dikirim ke *Python* dan data yang terbaca di *Pyhton* menjadi input ke *Arduinio IDE* sehingga dapat menggerakkan *servo*. Pergerakan servo secara horizontal (*tilt up* dan *tilt down*) dan vertikal (*pan right* dan *pan left*) sehingga wajah yang terdeteksi pada titik x dan y di kamera akan diikuti oleh motor *servo* dan mikrofon akan menyesuaikan posisinya dengan mulut wajah yang terdeteksi. Berikut adalah gambar skenario perancangan ROFATION yang mampu mendeteksi objek wajah dengan gerak *servo 2-axis*:



Gambar 3. 3 Tampak depan ROFATION



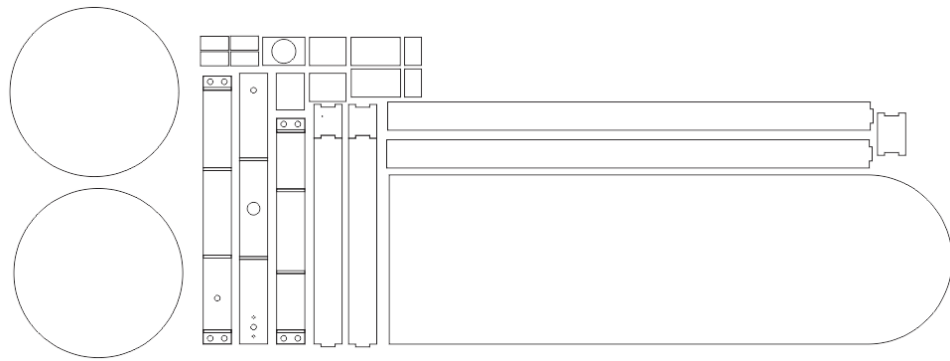
Gambar 3. 4 Tampak samping kiri dan kanan ROFATION



Gambar 3. 5 Tampak atas ROFATION

3.2.1.2 Perancangan Mekanik

Pada perancangan mekanik bahan yang digunakan pada pembuatan ROFATION ini adalah akrilik dengan ketebalan 2mm. Penggunaan bahan akrilik dipilih karena bahan tersebut cukup mudah dan kuat dalam pembuatannya serta akrilik cukup ringan sebagai beban motor *servo* yang bergerak secara horizontal maupun vertikal. Perancangan mekanik ROFATION dimulai dengan membuat desain pada *software SketchUp 2016* dimana dapat membuat desain 3D. Pertama dilakukan membuat desain letak mikrofon dan terdapat 2 buah motor *servo pan* dan *tilt* atau motor *servo 2-axis* yang dapat menggerakkan mikrofon baik secara horizontal maupun vertikal. Setelah desain letak mikrofon selanjutnya desain letak kamera dimana letak kamera berada dibelakang mikrofon sehingga dapat mendeteksi wajah dengan tepat tidak terlalu jauh maupun dekat.



Gambar 3. 6 Desain 2D akrilik ROFATION

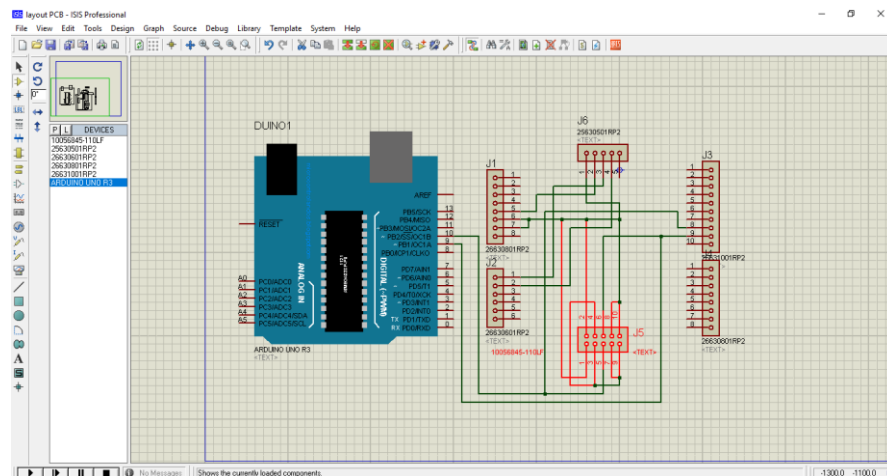
Setelah desain 3D selesai, tahap selanjutnya adalah membuat desain 2D dengan *software Corel Draw X* dimana dapat memudahkan dalam pencetakan akrilik. Akrilik yang dicetak sesuai desain menggunakan *printer laser* untuk mendapatkan hasil yang presisi. Setelah bahan sudah terpotong, didapatkan komponen dudukan motor *servo x* dan *y*, dudukan kamera, dudukan mikrofon, lengan dan dudukan mikrokontroler Arduino Uno R3. Setiap bagian dihubungkan menggunakan baut M3 1CM, dan untuk dudukan Arduino Uno R3 menggunakan *spacer* kuningan 10MM+6MM serta *spacer* perak 20MM+6MM. Selanjutnya motor *servo* diatur pada posisi sentral atau sudut 90^0 sehingga mikrofon tepat pada posisi horizontal menghadap kedepan.



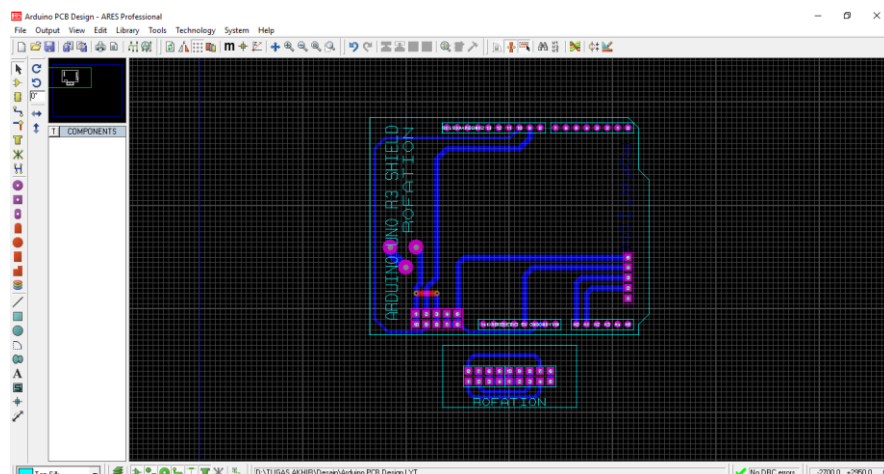
Gambar 3. 7 Realisasi dari desain mekanik

3.2.1.3 Perancangan Elektrik

Perancangan elektrik pada ROFATION sistem rangkaian alat dibuat skematiknya terlebih dahulu yaitu *shield board*. Pembuatan *shield board* dimulai dengan merancang skematik dan rangkaian pada *software Proteus 7 Professional*, dimana skematik tersebut akan dibuat *board* jalurnya pada *ARES 7 Professional* sehingga dapat menghubungkan setiap komponen yang saling terintegrasi. Selanjutnya, desain dari *ARES 7 Professional* akan di *import* ke PDF untuk melakukan proses selanjutnya yaitu mencetak *layout PCB* pada kertas *art paper* untuk dijadikan *shield board*. Skematik dan *layout PCB* yang dibuat pada *software Proteus 7 Professional* ditampilkan pada gambar berikut.



Gambar 3. 8 Skematik PCB shield board



Gambar 3. 9 Layout PCB shield board

Setelah *layout PCB shield board* yang di cetak pada kertas *art paper* kemudian diletakkan pada papan *PCB* yang sudah dihaluskan permukaannya

menggunakan amplas 120 untuk mempermudah tinta pada *art paper* menempel pada papan *PCB*. Papan *PCB* yang sudah ditempelkan dengan kertas *art paper* selanjutnya dipanaskan menggunakan mesin laminating maupun setrika agar skematik pada kertas *art paper* dapat tertempel pada papan *PCB*. Setelah proses memindahkan gambar skematik dari *art paper* ke papan *PCB*, maka papan *PCB* direndam dengan air biasa sampai desain terlihat dan kertas *art paper* terkelupas.



Gambar 3. 10 Proses penempelan rangkaian ke papan *PCB*

Proses selanjutnya papan *PCB* direndam pada larutan $FeCl_3$ yang telah dicampur dengan air hangat dengan menggerakkan wadah pelarutan sehingga dapat mempercepat proses pelarutan tembaga pada papan *PCB*. Selanjutnya setelah terlihat rangkaian skematik pada papan *PCB*, maka dapat dibersihkan dengan air mengalir sehingga diharapkan tidak ada larutan $FeCl_3$ yang tertinggal pada papan *PCB* dan tinta yang masih tertempel pada jalur dapat dihilangkan menggunakan tiner sehingga permukaan papan *PCB* terlihat jalur tembaga.



Gambar 3. 11 Proses melarutkan PCB

Tahap selanjutnya papan *PCB* yang sudah memiliki jalur terdapat lubang kaki komponen yang harus dilubangi menggunakan bor. Setelah papan *PCB* dilubangi, maka komponen – komponen dipasang sesuai letaknya kemudian disolder. *Shield board* yang sudah jadi selanjutnya dapat dihubungkan ke Arduino Uno R3 dan *Module regulator XL4005 step-down*. *Module regulator XL4005 step-down* berfungsi untuk memberikan khusus tegangan ke motor *servo* yang didapat melalui baterai *Li-ion 3.7 V* yang dirangkai seri. *Motor servo* bekerja dengan baik pada tegangan 4.7 V sehingga pada *output regulator* dapat diatur sehingga memiliki *output* sebesar 4.7 V.



Gambar 3. 12 Proses pengeboran PCB dan penyolderan kaki komponen

3.2.1.4 Perancangan Mikrokontroler Arduino Uno R3

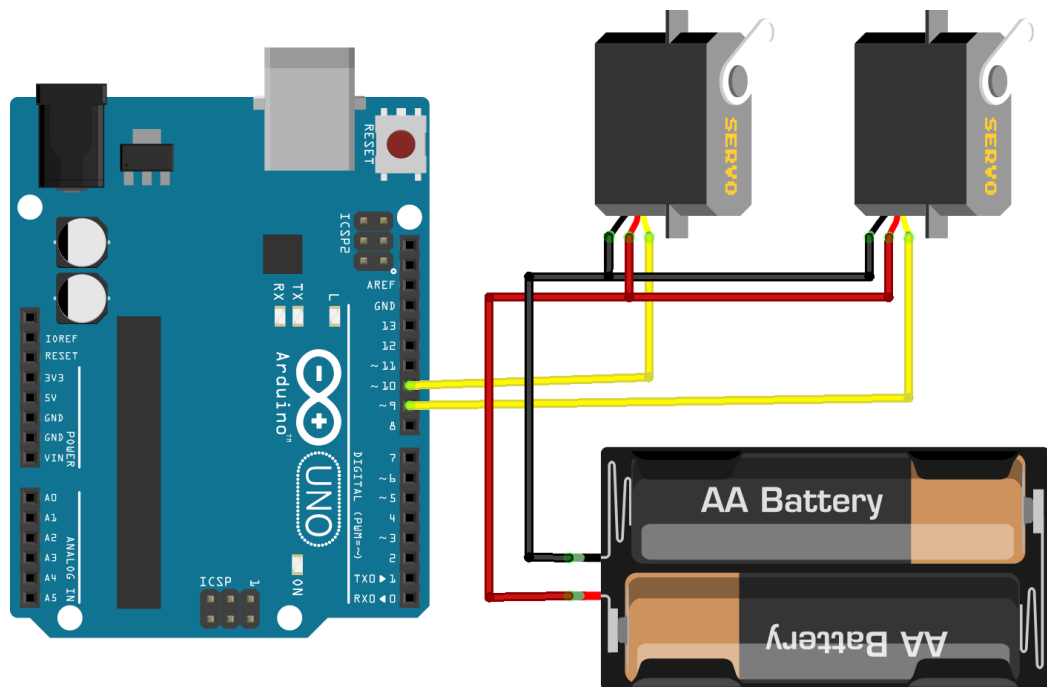
Dalam perancangan kontroler menggunakan sebuah mikrokontroler Arduino Uno R3 dimana *board* mikrokontroler berbasis ATmega328p. Pada Arduino Uno R3 memiliki digital *input/ output* sebanyak 14 pin dan dapat beroperasi pada tegangan 5 V. Setiap pin dapat menyediakan atau menerima arus maksimal 40 mA dan memiliki resistor *pull-up internal* (terputus secara *default*) 20-50 kOhms. Ardunio Uno R3 memiliki *output PWM* 8 bit dengan fungsi `analogWrite()`. Adapun pin yang digunakan dalam pembuatan ROFATION sebagai berikut:

Tabel 3. 1 Pin Arduino yang terhubung

Pin Mikrokontroler	Pin Arduino	Tipe pin	Fungsi
PD0 (RXD)	0	I/O	Tidak dipakai
PD1 (TXD)	1	I/O	Tidak dipakai
PD2 (INT0)	2	I/O	Tidak dipakai
PD3 (INT1)	3	I/O	Tidak dipakai
PD4	4	I/O	Tidak dipakai
PD5	5	I/O	Tidak dipakai
PD6	6	I/O	Tidak dipakai
PD7	7	I/O	Tidak dipakai
PB0	8	I/O	Tidak dipakai
PB1	9	I/O	Data Servo 1
PB2 (SS')	10	I/O	Data Servo 2
PB3 (MOSI)	11	I/O	Tidak dipakai
PB4 (MISO)	12	I/O	Tidak dipakai
PB5 (SCK)	13	I/O	Tidak dipakai
PC0	A0	I/O	Tidak dipakai

Tabel 3. 2 Pin Arduino yang terhubung (Lanjutan)

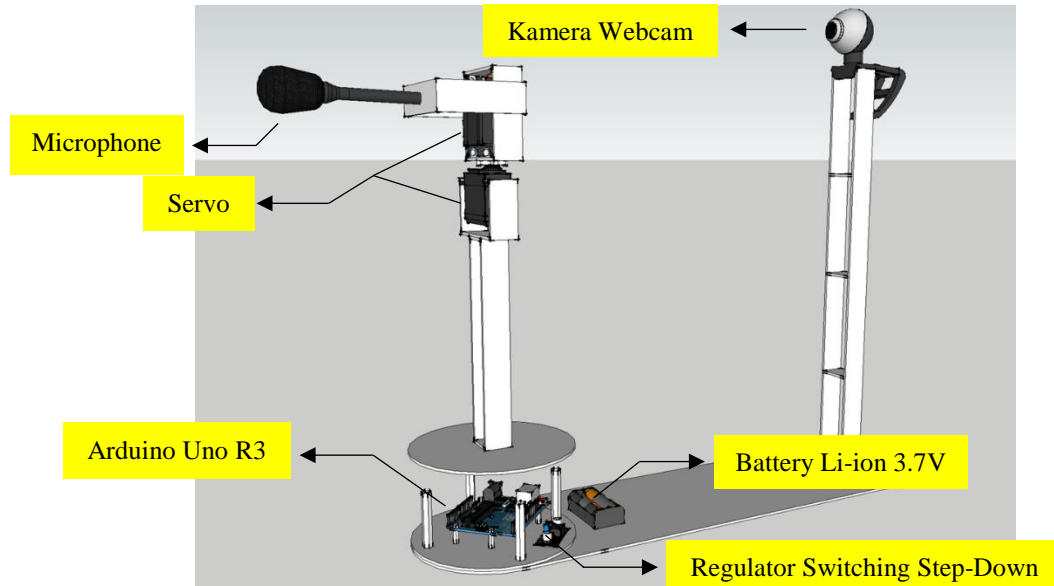
Pin Mikrokontroler	Pin Arduino	Tipe pin	Fungsi
PC1	A1	I/O	Tidak dipakai
PC2	A2	I/O	Tidak dipakai
PC3	A3	I/O	Tidak dipakai
PC4 (SDA)	A4	I/O	Tidak dipakai
PC5 (SCL)	A5	I/O	Tidak dipakai
GND	GND	Ground	Ground
5V	5V	5V	5V Input



Gambar 3. 13 Skematik penggunaan pin arduino ke servo

3.3.1 Bahan Yang Digunakan ROFATION

Bahan-bahan yang diperlukan pada tahap pembuatan ROFATION adalah sebagai berikut:



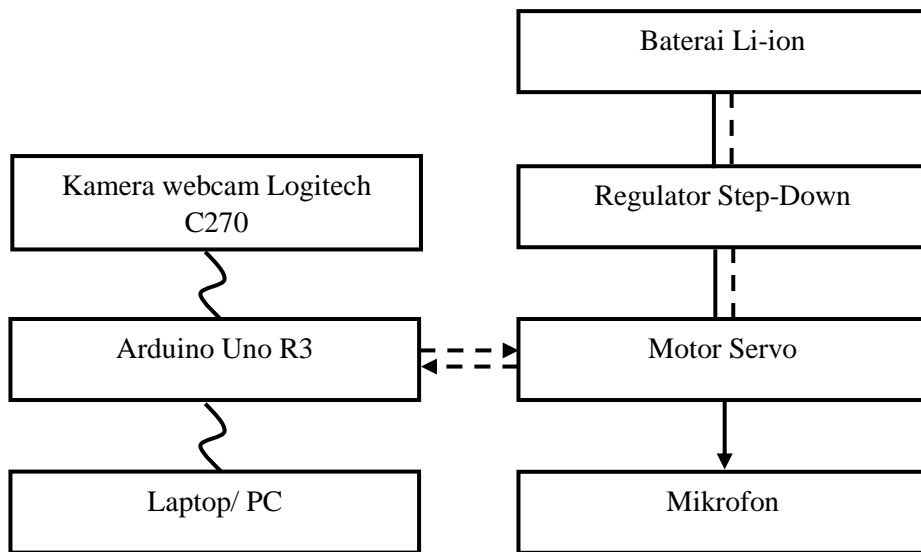
Gambar 3. 14 Komponen-komponen ROFATION

1. Komponen elektronika pada ROFATION:
 - a. Kamera webcam Logitech C270
 - b. Arduino Uno R3
 - c. Servo Tower Pro MG 996R
 - d. Module Regulator XL4005 Step-Down
 - e. Baterai Li-ion 3.7V
 - f. Panasonic mikrofon kondensator
 - g. Kabel 10P 30cm
 - h. Kabel jumper male-male
 - i. Kabel jumper
2. Komponen mekanik pada ROFATION:
 - a. Akrilik 2mm
 - b. Spacer kuningan 10MM+6MM
 - c. Spacer perak 20MM+6MM
 - d. Baut M3 1CM
 - e. Mur M3

- f. Sekrup 1CM
- 3. *Software* Pendukung
 - a. *SketchUp + Layout 2016*
 - b. *Proteus 7 Professional*
 - c. *Ardunio IDE*
 - d. *Python 2.7*
 - e. *Open CV*


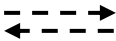


3.3 Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak (*software*) dilakukan berguna untuk memberikan perintah terhadap mikrokontroler supaya dapat bekerja sesuai program yang diberikan. Terdapat dua buah program yang ada pada pembuatan ROFATION sehingga perlu dilakukan penghubungan antara dua program tersebut. Diagram alir (*flowchart*) sistem yang akan dirancang adalah prinsip kerja sistem secara keseluruhan. Berikut adalah diagram blok seluruh sistem pada ROFATION.



Gambar 3. 15 Diagram blok sistem kerja ROFATION

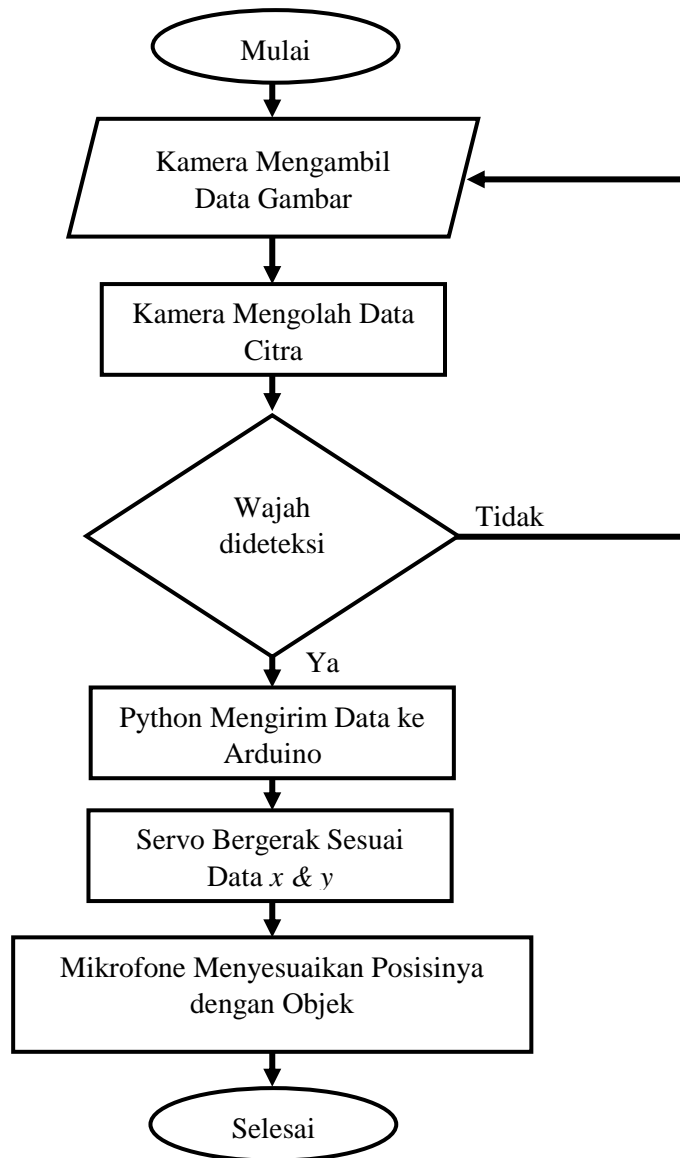
Keterangan:

-  = Kabel *USB*
-  = Data pin arduino
-  = Arus listrik *DC*
-  = Terhubung Langsung

Prinsip kerja diagram blok sistem adalah sebagai berikut:

1. Baterai *Li-ion* 2 sel berfungsi sebagai pencatu daya 2 motor *servo*.
2. *Module regulator XL4005 Step-Down* berfungsi sebagai penurun tegangan yang dibutuhkan oleh motor *servo* yaitu 4.7 V.
3. Kamera webcam *Logitech C270* bekerja sebagai pengelola gambar atau mendeteksi wajah kemudian akan memberikan *input* data pada mikrokontroler Arduino Uno R3.
4. Arduino Uno R3 berfungsi sebagai pengolah data yang diterima oleh kamera webcam *Logitech C270* yang kemudian data akan dikirim melalui komunikasi serial antara *Python* dengan Arduino.
5. *Python* akan menerima data koordinat *x* dan *y* yang telah terdeteksi oleh kamera kemudian data tersebut akan menjadi *input* yang dikirim ke Arduino untuk menggerakkan motor *servo 2-axis*.
6. Kamera tidak bergerak namun mikrofon yang bergerak sehingga mikrofon dapat sesuai posisinya dengan mulut wajah yang terdeteksi oleh kamera.

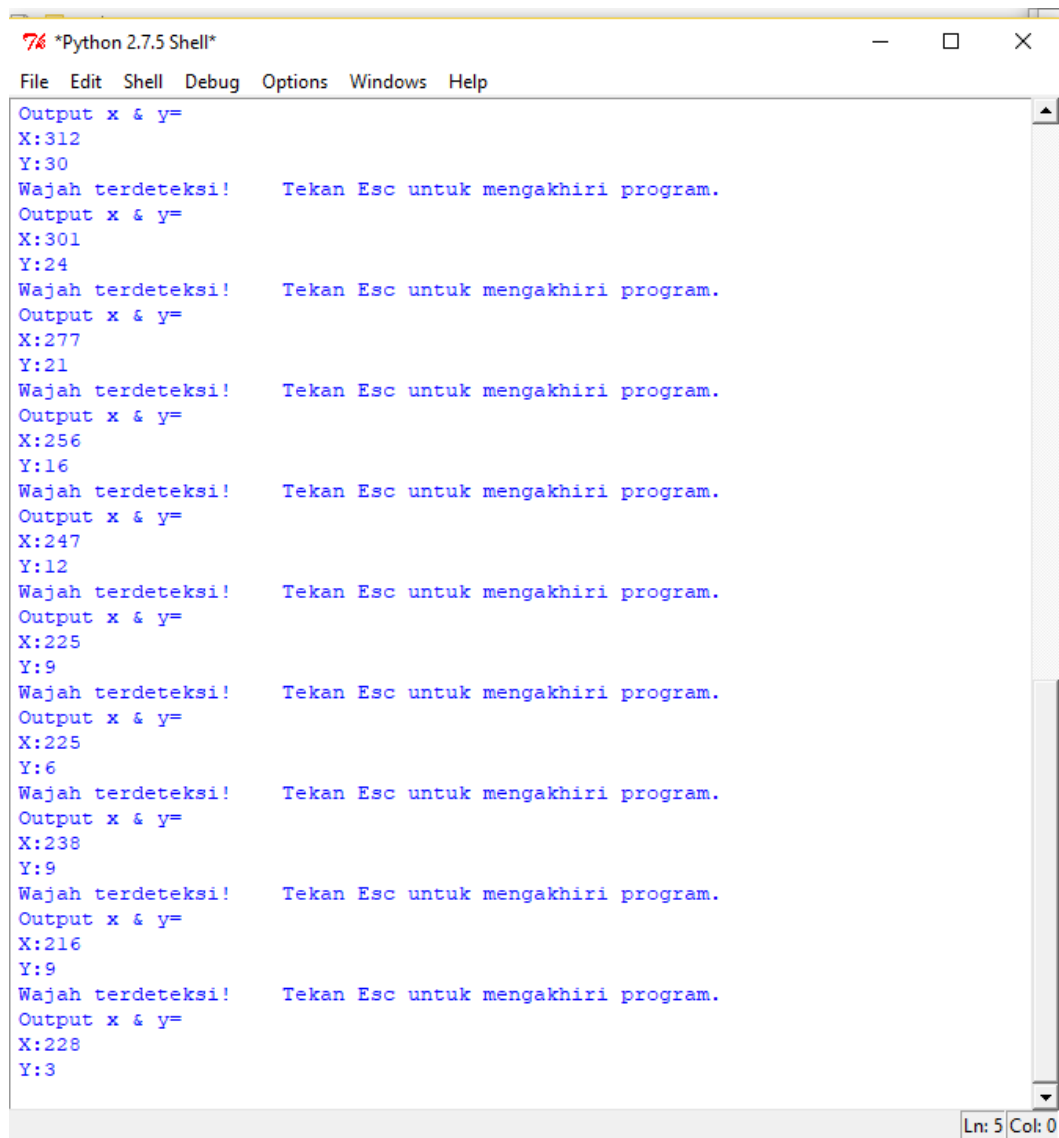
Dari diagram blok diatas memamparkan bagaimana setiap komponen terhubung satu dengan yang lain dan bagaimana kerja dari setiap masing – masing komponen pada ROFATION. Dimana Data *x* dan *y* yang diterima oleh *Python* dari pendeteksian wajah oleh kamera selanjutnya akan dikirimkan ke mikrokontroler Arduino Uno R3. Data keluaran yang berupa nilai *ADC (Analog to Digital Converter)* dan *PWM (Pulse Width Modulation)* akan mengatur putaran motor *servo 2-axis* sesuai dengan nilai *input* koordinat *x* dan *y* sehingga pergerakan motor *servo* dapat mengikuti pergerakan wajah/ objek yang tertangkap pada layar kamera dan dapat menyesuaikan posisi microphone dengan mulut pada wajah/ objek. Adapun *flowchart* sistem kerja dari ROFATION dari pendeteksian wajah hingga *output* motor *servo* yang bergerak secara vertikal dan horisontal sehingga dapat menyesuaikan posisi mikrofon dengan mulut ditampilkan pada gambar 3.16.



Gambar 3. 16 Flowchart sistem kerja ROFATION

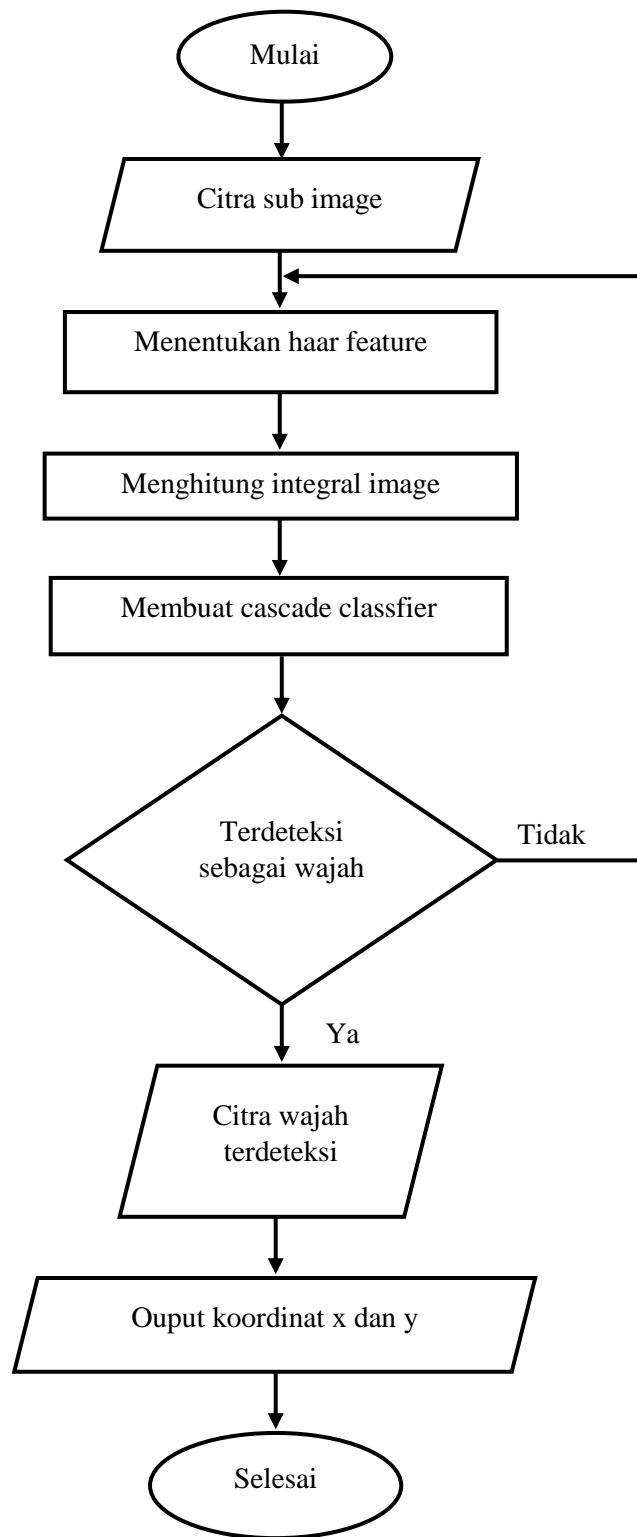
Kamera *Logitech C270* akan mengambil data citra sesuai *library* yang dimasukkan pada program *python*, dan *library* yang dimasukkan untuk mendeteksi wajah. Wajah yang terdeteksi akan dibaca oleh *python* yaitu sumbu *x* dan *y* dimana data tersebut akan dikirim ke Arduino melalui komunikasi serial antara *python* dengan Arduino. Apabila objek wajah tidak terdeteksi oleh kamera maka Arduino akan diam dan posisi servo berada pada posisi terakhir wajah yang terdeteksi. Jika wajah terdeteksi maka Arduino yang sudah mendapat data dari *python* akan diolah untuk menggerakkan servo *2-axis* dengan data *x* menggerakkan motor *servo* secara horizontal dan data *y* menggerakkan motr *servo* secara vertikal.

Terdapat proses pada pendeteksian wajah sehingga mendapatkan hasil koordinat x dan y untuk menggerakkan motor servo. Untuk melakukan pendeteksian wajah dilakukan dengan menggunakan *haar cascade classifier* dimana dapat mendeteksi bagian tubuh termasuk wajah. *Haar cascade classifier* yang terdapat di dalam *OpenCV* berguna untuk ROFATION dimana dapat melakukan pendeteksian wajah sehingga *output* motor *servo* dapat bergerak sesuai objek/ wajah yang terdeteksi. Gambar 3.13 merupakan proses pendeteksi wajah menggunakan *haar cascade classifier*.



```
*Python 2.7.5 Shell*
File Edit Shell Debug Options Windows Help
Output x & y=
X:312
Y:30
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:301
Y:24
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:277
Y:21
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:256
Y:16
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:247
Y:12
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:225
Y:9
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:225
Y:6
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:238
Y:9
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:216
Y:9
Wajah terdeteksi! Tekan Esc untuk mengakhiri program.
Output x & y=
X:228
Y:3
Ln: 5 Col: 0
```

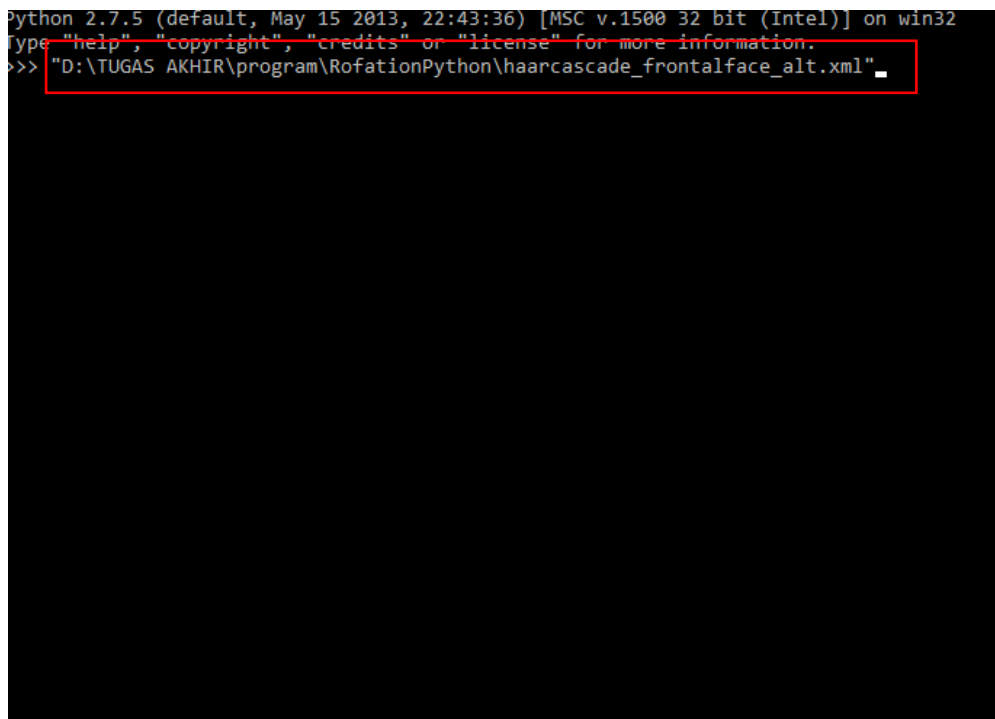
Gambar 3. 17 Output wajah terdeteksi dengan koordinat x dan y



Gambar 3. 18 Flowchart proses deteksi wajah

3.3.1 Perancangan Perangkat Lunak Komputer

Perancangan perangkat lunak pada komputer meliputi bagaimana proses instalasi dan fungsi dari *Python*. *Python* merupakan salah satu bahasa pemrograman tingkat tinggi yang bersifat *interpreter*, *interactive*, *object-oriented*, dan dapat beroperasi hampir di semua *platform*. Salah satu *library* yang mendukung *Python* pada penelitian ini adalah *OpenCV*, *Spicy*, *PySerial*, *Numpy*, dan *Matplotlib*. Untuk mendapatkan pendeteksian wajah maka perlu ditambahkan *source code* dari *library OpenCV* dimana *library* yang digunakan adalah untuk pendeteksian seluruh wajah. Saat proses pendeteksi wajah ketika ada bagian wajah yang tertutup seperti mata maka kamera tidak dapat mendeteksi wajah.

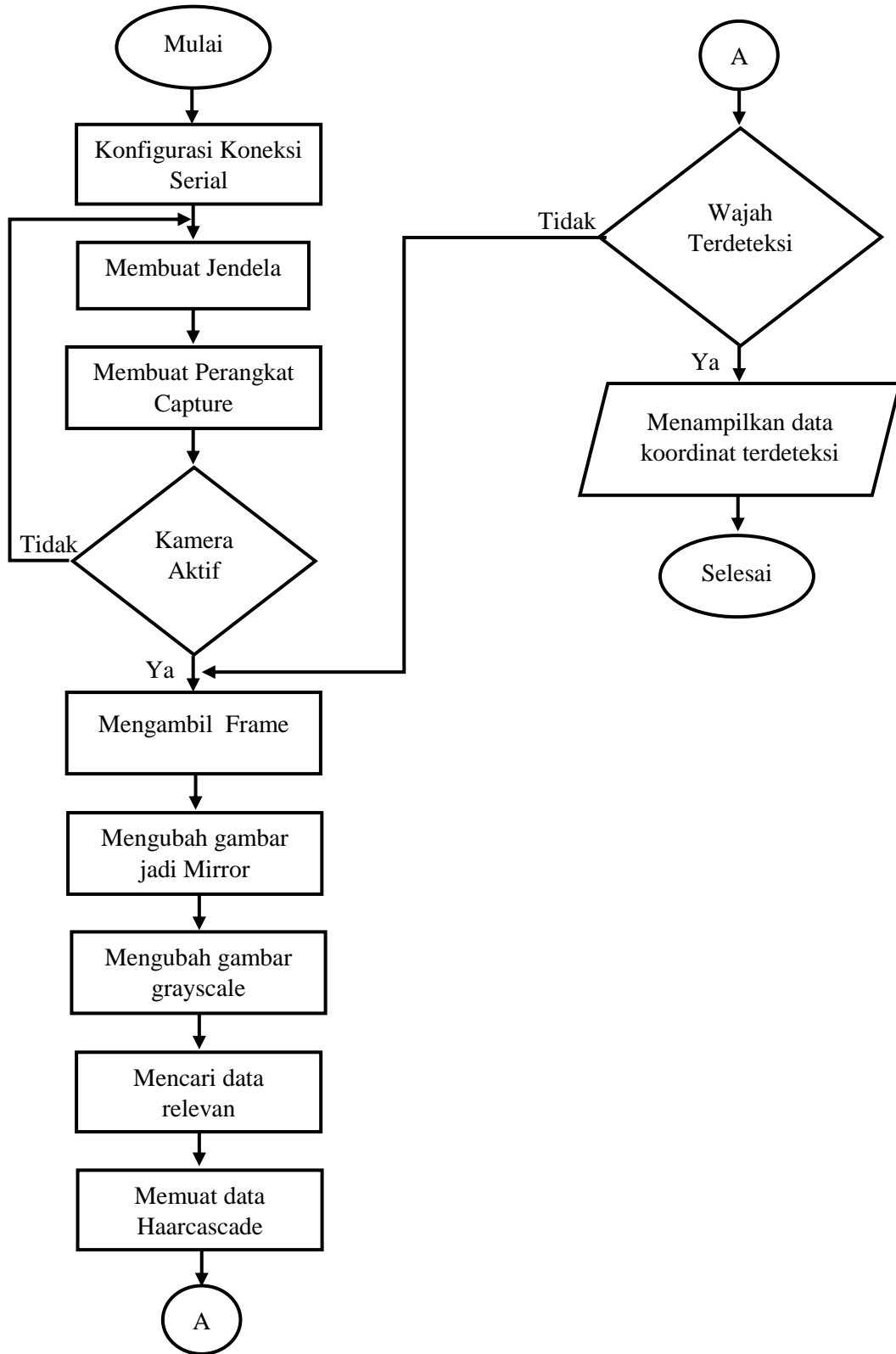


```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> "D:\TUGAS AKHIR\program\RofationPython\haarcascade_frontalface_alt.xml" _
```

Gambar 3. 19 Source Code OpenCV Pendeteksi Wajah

Dari *source code OpenCV* tersebut disalin ke *Python Command Line* untuk mendapatkan file *XML* yang akan dipanggil pada *IDLE Python GUI*. File *XML* yang telah dimasukkan ke *IDLE Python GUI* berfungsi untuk mencari data kaskade yang relevan sehingga kamera dapat melakukan pendeteksian dan mengikuti wajah. Berikut adalah bagaimana proses pendeteksiannya sampai mendapatkan koordinat wajah pada *flowchart* dan penjabaran dari programnya:

A. Flowchart program Python



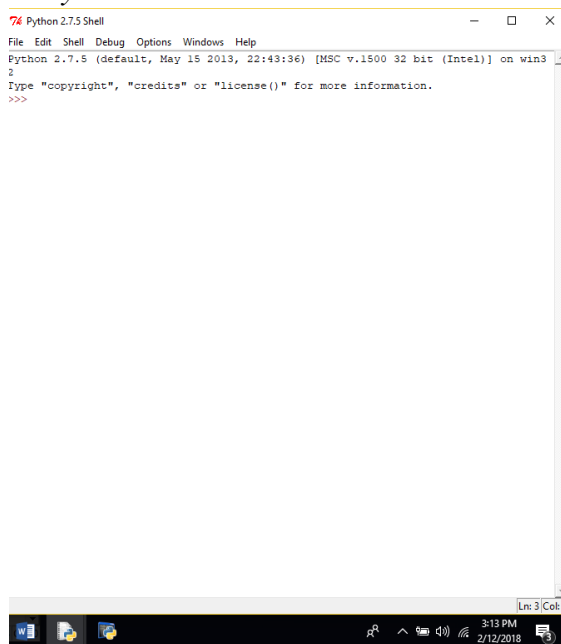
Gambar 3. 20 Flowchart proses kerja program Python

B. Untuk menjalankan *Python* klik ikon seperti gambar di bawah



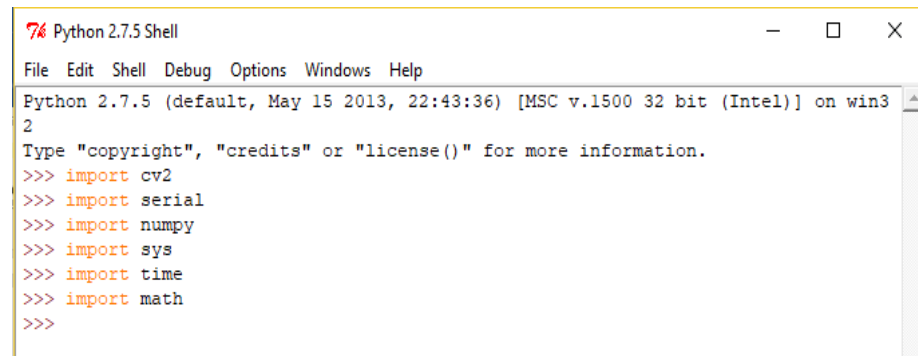
Selain *IDLE (Python GUI)* terdapat dua *software* pendukung *Python* yaitu *Module Docs* dan *Python Command Line*. *Module Docs* berisi *library* atau dokumen *Python* yang sudah permanen terinstal, adapun *library* yang diinstal manual tersimpan di *Module Docs* seperti *PIL*, *Numpy*, *Scipy*, dan *Serial*. Sedangkan *Python Command Line* sama seperti *command promp* pada *windows* yaitu untuk mengeksekusi file dengan menggunakan *di-keyboard*. *Python Command Line* dapat menterjemahkan data *source code* dari *OpenCV* supaya dapat dimasukkan ke program *Python* yang akan dibuat.

C. Tampilan *IDLE Python GUI*



Python tidak memiliki semua *library* yang dibutuhkan, sehingga *library* yang perlu untuk digunakan dapat diinstal manual. Jika *library* yang dibutuhkan sudah terinstal, maka untuk melakukan pengecekan dapat

dilakukan dengan menggunakan dengan perintah “*Import*” yang diikuti dengan nama library yang diinstal gambar dibawah ini.



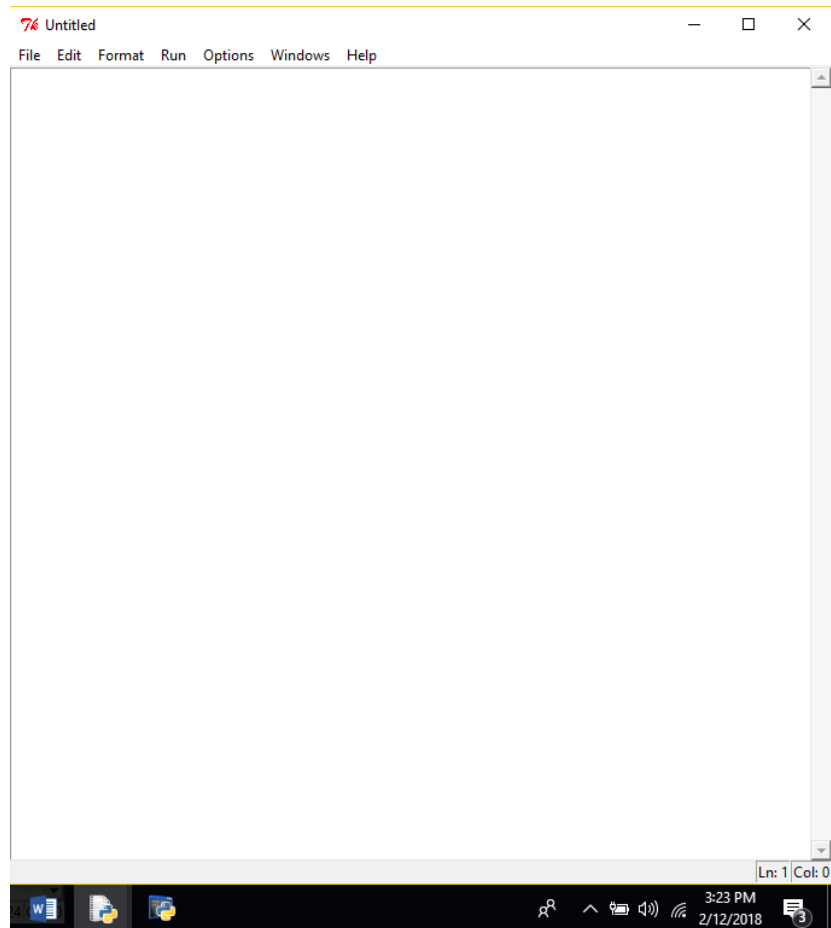
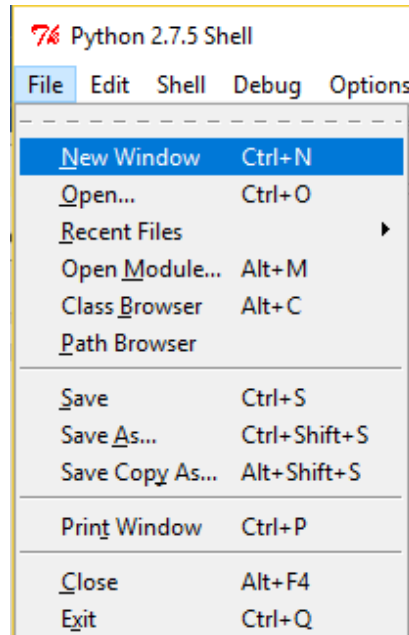
```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win3
2
Type "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> import serial
>>> import numpy
>>> import sys
>>> import time
>>> import math
>>>
```

Berikut adalah fungsi dari *library* yang digunakan pada alat ROFATION:

- Import cv2
Merupakan perintah untuk menjalankan fungsi dari *source code OpenCV*
- Import Serial
Merupakan perintah untuk menjalankan fungsi dari *PySerial* yang berfungsi menghubungkan antara *Python* dengan *Arduino*
- Import Numpy
Merupakan perintah untuk menjalankan modul yang berfungsi dalam komputasi ilmiah.
- Import sys
Modul sys digunakan untuk mengakses konfigurasi interpreter pada saat *runtime* dan berinteraksi dengan *environment* sistem operasi.
- Import time
Modul time merupakan modul untuk memberikan *delay* (waktu penundaan).
- Import math
Modul math berisi fungsi-fungsi matematika yang dapat menjalankan perintah-perintah menghitung dalam program.

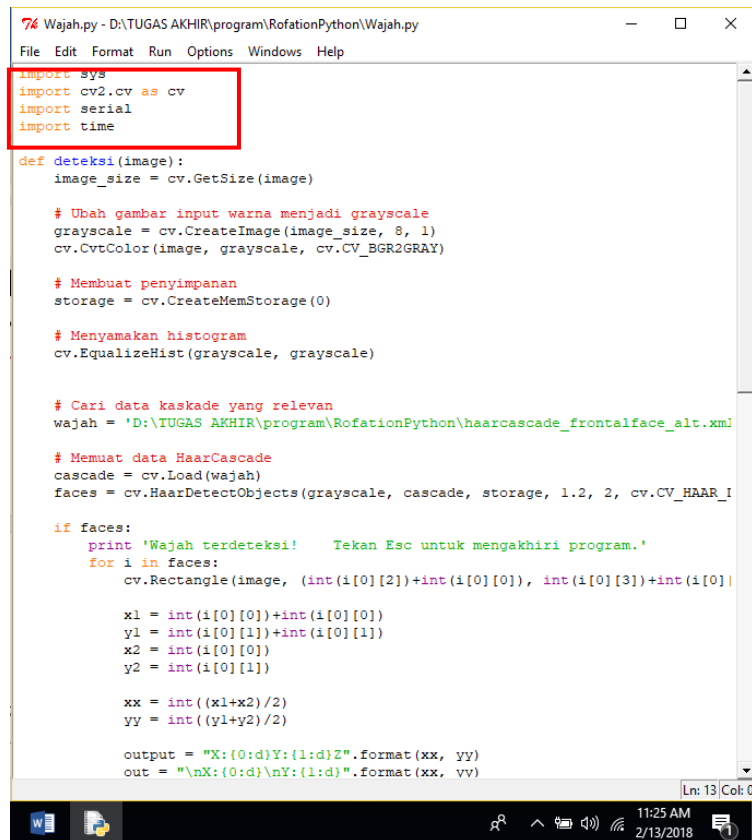
D. Membuat Jendela Pemrograman

Untuk membuat skrip program tidak dapat dilakukan pada *Python Shell*, jadi untuk membuat program dapat dilakukan dengan cara mengklik *File* selanjutnya memilih *New Window* dan tertampil seperti gambar dibawah ini.



Selanjutnya adalah tahap pembuatan program dari ROFATION, dan untuk memulai program yang utama adalah *import* modul – modul yang diperlukan dalam program. Pada penelitian ini modul – modul yang

digunakan adalah *import cv2*, *serial*, *sys*, dan *time* seperti pada gambar dibawah ini.



```
Wajah.py - D:\TUGAS AKHIR\program\RofationPython\Wajah.py
File Edit Format Run Options Windows Help
import sys
import cv2.cv as cv
import serial
import time

def deteksi(image):
    image_size = cv.GetSize(image)

    # Ubah gambar input warna menjadi grayscale
    grayscale = cv.CreateImage(image_size, 8, 1)
    cv.CvtColor(image, grayscale, cv.CV_BGR2GRAY)

    # Membuat penyimpanan
    storage = cv.CreateMemStorage(0)

    # Menyamakan histogram
    cv.EqualizeHist(grayscale, grayscale)

    # Cari data kaskade yang relevan
    wajah = 'D:\TUGAS AKHIR\program\RofationPython\haarcascade_frontalface_alt.xml'

    # Memuat data HaarCascade
    cascade = cv.Load(wajah)
    faces = cv.HaarDetectObjects(grayscale, cascade, storage, 1.2, 2, cv.CV_HAAR_I

if faces:
    print 'Wajah terdeteksi! Tekan Esc untuk mengakhiri program.'
    for i in faces:
        cv.Rectangle(image, (int(i[0][2])+int(i[0][0]), int(i[0][3])+int(i[0]

        x1 = int(i[0][0])+int(i[0][0])
        y1 = int(i[0][1])+int(i[0][1])
        x2 = int(i[0][0])
        y2 = int(i[0][1])

        xx = int((x1+x2)/2)
        yy = int((y1+y2)/2)

        output = "X:{0:d}Y:{1:d}Z".format(xx, yy)
        out = "\nX:{0:d}\nY:{1:d}".format(xx, yy)
Ln: 13 Col: 0
```

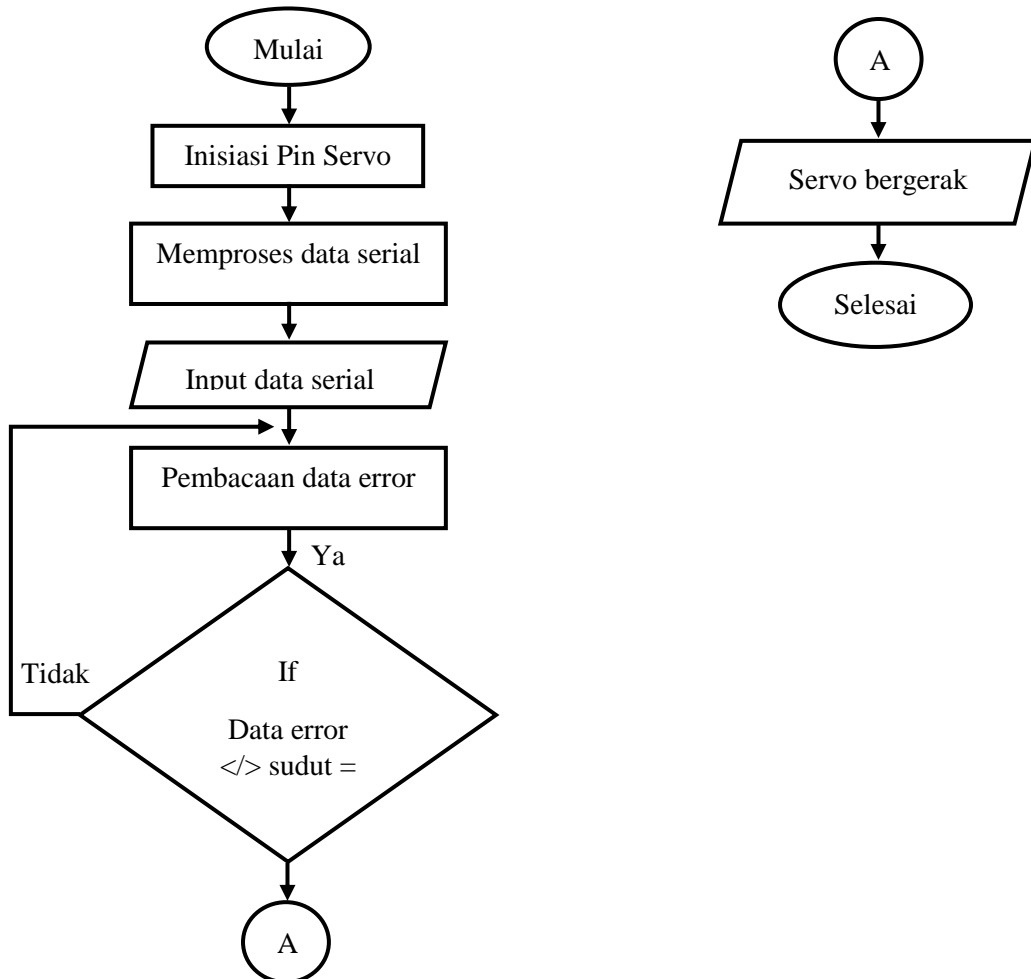
Untuk menjalankan program, terlebih dahulu Arduino di-*compile* dan mengupload program. Apabila program sudah berhasil di-*upload* ke Arduino selanjutnya mengklik *Run* pada menu *bar Python* dan pilih *Run Module* atau bisa dengan menekan tombol F5 pada keyboard maka program akan berjalan. Setelah *Run* dimulai maka kamera akan aktif dan memunculkan penangkapan gambar dengan resolusi 720x480.

Setelah kamera aktif kamera akan melakukan pendeteksian wajah ketika ada objek didepan kamera dan indikatornya terdapat kotak merah pada wajah dan mengikuti wajah selama masih dalam batas penangkapan kamera. Pada kotak berwarna merah akan mengirimkan data koordinat wajah yaitu *x* dan *y* dimana data tersebut akan dikirim ke Arduino untuk menggerakkan 2 buah motor *servo* sehingga kemana arah wajah yang terdeteksi dapat diikuti oleh motor *servo* dan mikrofon posisinya dapat sesuai dengan mulut.

3.3.2 Perancangan Perangkat Lunak Mikrokontroler

Perancangan perangkat lunak pada komputer meliputi bagaimana proses instalasi dan fungsi dari Arduino *IDE*. Arduino *IDE* merupakan *software* untuk membuat *sketch* program yang akan di masukkan ke dalam mikrokontroler Arduino. *Sketch* program yang telah dibuat *output* yang dihasilkan akan digunakan untuk menggerakkan motor *servo* yang terdapat pada ROFATION. Motor *servo* akan dapat bergerak apabila data yang diterima oleh Arduino *IDE* dihubungkan secara serial dengan *output* koordinat dari *Python*. Adapun cara menjalankan Arduino *IDE* untuk menggerakkan dan menghubungkan antara Arduino *IDE* dengan *Python* sebagai berikut.

A. Flowchart program Arduino

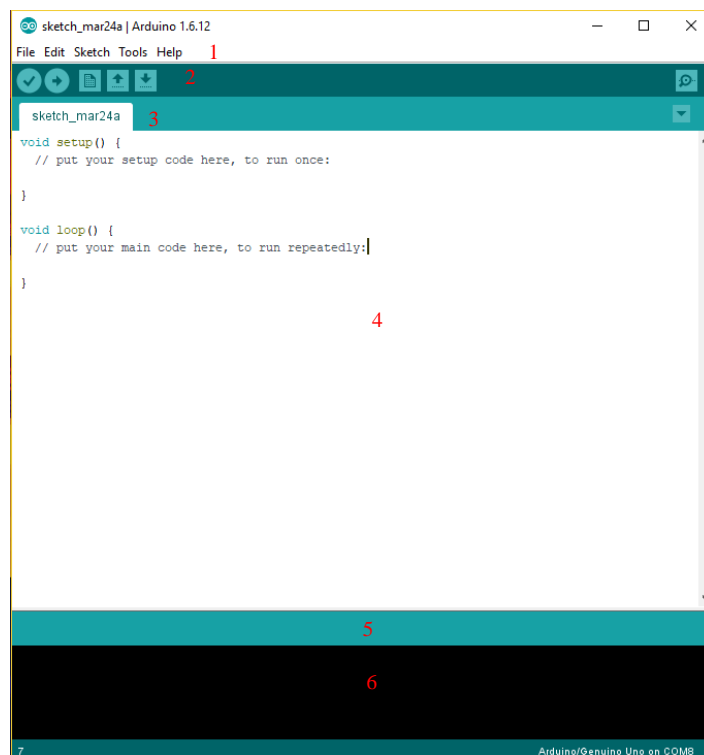


Gambar 3. 21 Flowchart proses kerja program Arduino

B. Pada desktop klik ikon seperti gambar dibawah ini untuk membuka Arduino *IDE* dan menjalankannya.



C. Setelah mengklik ikon diatas maka akan tertampil sketch untuk membuat program Arduino *IDE* seperti gambar di bawah.



Adapun bagian – bagian yang ada pada Arduino *IDE* dan fungsinya:

1. Menu Arduino *IDE*

Fungsi dari menu diatas adalah untuk membantu melakukan *programming* dimana terdapat *File*, *Edit*, *Sketch*, *Tools*, dan *Help*. Dari masing – masing menu diatas terdapat sub menu yang memiliki fungsinya sendiri seperti *FILE* terdapat sub menu *New*, *Open*, *Save*, *Print*, *Sketch Book*, *Example*, *Save* dan banyak untuk melakukan interaksi.

2. *Shortcut*

Fungsi dari bagian *shortcut* untuk melakukan proses pemrograman seperti *verify* untuk memverifikasi *sketch* program yang telah dibuat, *upload* untuk memasukkan program ke dalam mikrokontroler Arduino, *New* untuk membuat *sketch* program baru, *Open* untuk membuka program yang telah dibuat, dan *Save* untuk menyimpan program. *Shortcut* paling kanan yaitu *Serial Monitor* yang berfungsi untuk memonitor data *output* dari *sketch* program yang dibuat.

3. Nama *Sketch*

Fungsi dari bagian ini untuk mengetahui nama *sketch* program yang telah dibuat apakah *sketch* program yang telah dibuat telah tersimpan dengan nama yang diinginkan.

4. *Sketch Editor*

Fungsi dari bagian ini adalah tempat menulis dan mengedit program Arduino dalam bahasa C.

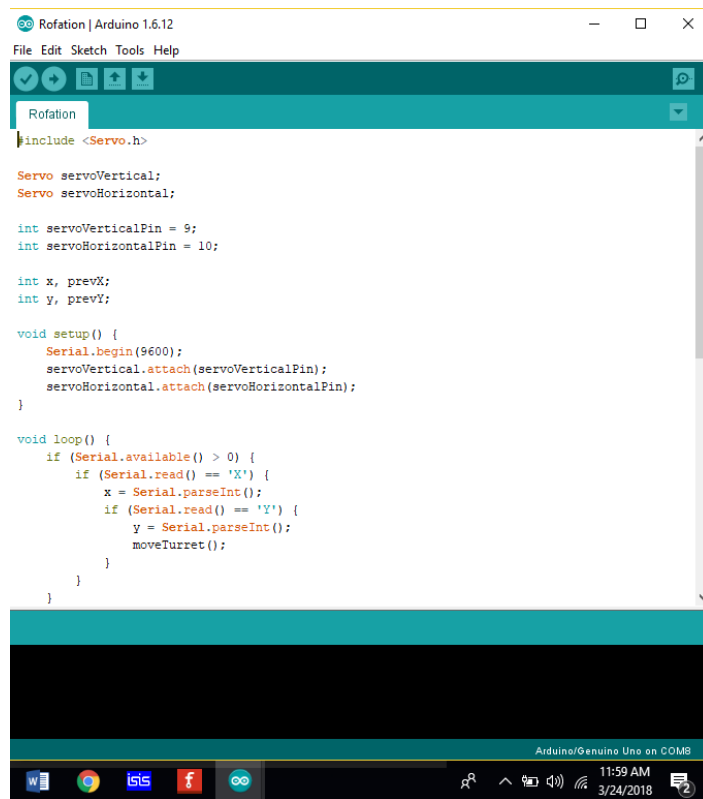
5. Jendela Informasi

Fungsi dari bagian ini dimana ketika melakukan *verify* atau *upload* program dapat melihat informasi yang terjadi apakah program sudah terverifikasi atau sudah masuk ke *IC* mikrokontroler.

6. Jendela *Debug*

Fungsi dari bagian ini adalah untuk melihat hasil dari *verify* program apakah terjadi kesalahan atau *error* dari program, sehingga dapat memudahkan mencari *bug* kesalahan yang ada pada *sketch* program.

D. Untuk menjalankan ROFATION sesuai fungsinya maka pada *sketch* programnya seperti gambar dibawah ini.



```
Arduino IDE - Rofation | Arduino 1.6.12
File Edit Sketch Tools Help
Rofation
#include <Servo.h>

Servo servoVertical;
Servo servoHorizontal;

int servoVerticalPin = 9;
int servoHorizontalPin = 10;

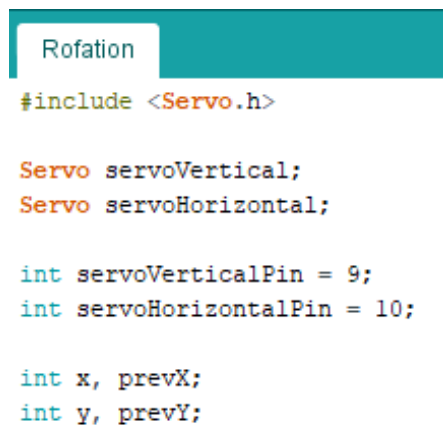
int x, prevX;
int y, prevY;

void setup() {
  Serial.begin(9600);
  servoVertical.attach(servoVerticalPin);
  servoHorizontal.attach(servoHorizontalPin);
}

void loop() {
  if (Serial.available() > 0) {
    if (Serial.read() == 'X') {
      x = Serial.parseInt();
      if (Serial.read() == 'Y') {
        y = Serial.parseInt();
        moveTurret();
      }
    }
  }
}
```

Terdapat tiga bagian pada *sketch* program yaitu deklarasi awal, bagian *setup*, dan bagian *loop*/ program utama. :

1. Deklarasi awal



```
Rofation
#include <Servo.h>

Servo servoVertical;
Servo servoHorizontal;

int servoVerticalPin = 9;
int servoHorizontalPin = 10;

int x, prevX;
int y, prevY;
```

Sketch program diatas merupakan bagian deklarasi awal program dimana fungsinya untuk mendeskripsikan variable – variable yang akan digunakan pada program utama termasuk motor *servo* yang digunakan.

2. Bagian *setup*

```
void setup() {  
  Serial.begin(9600);  
  servoVertical.attach(servoVerticalPin);  
  servoHorizontal.attach(servoHorizontalPin);  
}
```

Sketch program diatas merupakan bagian *setup* program dimana fungsinya untuk mengatur *baudrate* serial *port* yang akan dihubungkan dengan *Python* dan untuk menginisiasi variabel pin yang digunakan pada Arduino.

3. Bagian *loop*/ program utama

```
Rofation  
void loop() {  
  if (Serial.available() > 0) {  
    if (Serial.read() == 'X') {  
      x = Serial.parseInt();  
      if (Serial.read() == 'Y') {  
        y = Serial.parseInt();  
        moveTurret();  
      }  
    }  
  }  
}  
  
void moveTurret() {  
  float kp = 0.15; //konstanta proporsional  
  int errorX = 90-kp*(240 - x);  
  int errorY = 90+kp*(360 - y);  
  
  if (errorX < 45) {  
    errorX = 45;  
  }  
  else if (errorX > 135){  
    errorX = 135;  
  }  
  if (errorY < 90) {  
    errorY = 90;  
  }  
  else if(errorY > 145) {
```

Bagian terakhir yaitu bagian *loop* serta program utama untuk menjalankan ROFATION. Bagian *loop* berfungsi untuk menjalankan program secara berulang – ulang dimana program utama untuk menggerakkan motor *servo* menggunakan *PID* dengan parameter yang digunakan adalah proporsional atau konstantan proporsional (kP).

Sistem dengan *PID* membutuhkan *input* dan *feedback* dari *output* untuk memprediksi sudut motor *servo* agar sudut tidak bergerak lebih dari sudut *error*-nya.

Pengontrolan dengan menggunakan *PID* dilakukan secara *realtime* dimana target *output* dengan *output* yang diharapkan dikomparasi berdasarkan rumus *transfer function* sistem tersebut. Jika ada perubahan pada alat – alatnya, maka rumus transfer functionnya juga akan berubah. Jadi *PID* bukan hanya rumus perhitungan *PID*-nya, tapi konstanta yang diberikan lebih penting untuk mendapatkan *output* yang diharapkan dan konstanta akan didapat apabila rumus *transfer function*-nya benar.

3.4 Tahap Pengujian

Tahap pengujian dilakukan pada penelitian ini bertujuan untuk mengetahui apakah komponen – komponen yang ada pada ROFATION dapat bekerja dengan baik atau tidak serta pengujian seluruh alat. Adapaun tahap pengujian yang dilakukan pada penelitian ini sebagai berikut:

3.4.1 Pengujian komponen

Pengujian komponen dilakukan secara satu persatu komponen, dimana untuk mengetahui apakah komponen yang ada pada alat mampu bekerja baik sesuai yang diinginkan pada penelitian ini.

A. Pengujian Baterai

Tujuan dilakukan pengujian baterai disini berguna untuk mengetahui baterai tahan digunakan hingga berapa lama dari baru digunakan sampai penggunaan terakhir.

B. Pengujian *Module Regulator XL4005*

Tujuan dilakukan pengujian pada *module regulator XL4005*, untuk mengetahui keluaran *output* tegangan sehingga *ouput* dapat diatur agar sesuai kebutuhan alat bekerja dengan baik.

C. Pengujian Mikrokontroler Arduino UNO R3

Tujuan dilakukan pengujian terhadap mikrokontroler untuk mengetahui apakah *output* yang dihasilkan sesuai dengan programnya dimasukkan pada *sketch* program Arduino *IDE*.

D. Pengujian Motor *Servo*

Pengujian motor *servo* dilakukan untuk mengetahui apakah motor *servo* dapat bekerja sesuai dengan sudut *input* yang diberikan sama dengan sudut yang terukur pada motor *servo*.

E. Pengujian Mikrofon

Pengujian mikrofon dilakukan untuk mengetahui apakah intensitas suara yang dihasilkan pada aplikasi pengukur decibel (dB) mendapatkan hasilnya normal dengan beberapa kondisi dan jarak yang diberikan.

3.4.2 Pengujian ROFATION

Pengujian ROFATION dilakukan dengan melakukan pengujian seluruh komponen yang digabungkan dan diberikan masukan program menggunakan *Python* yang dihubungkan secara serial dengan *Arduino IDE*, dimana untuk mengetahui apakah alat dapat bekerja dengan baik sesuai fungsinya. Adapaun pengujian dilakukan dengan menguji alat pada cahaya yang berbeda yaitu di dalam dan di luar ruangan untuk mengetahui kamera dapat bekerja baik pada cahaya yang berbeda. Pada mikrofonnya dilakukan pengujian dengan memberikan jarak dan kondisi mikrofon bergerak mengikuti wajah atau diam yang bertujuan untuk mengetahui pola radiasi yang dihasilkan dari mikrofon.