

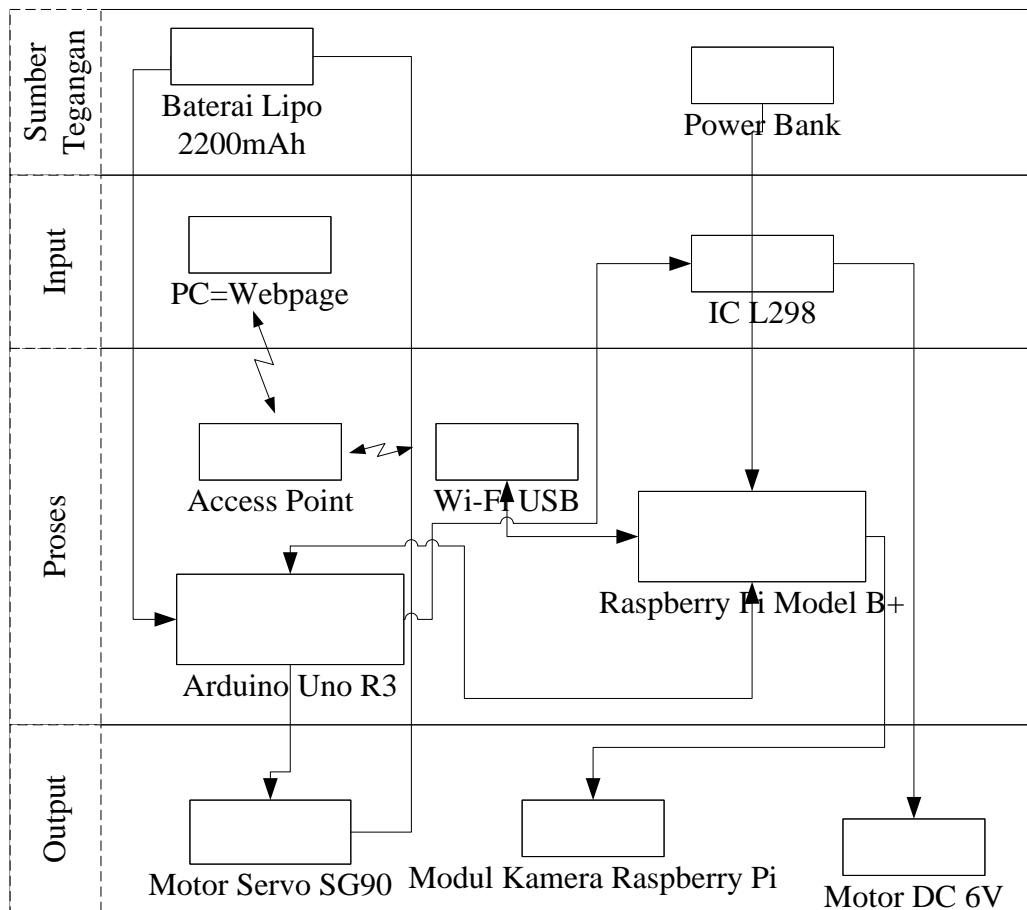
## BAB III

### PERANCANGAN SISTEM DAN PEMBUATAN ALAT

Perancangan sistem dan pembuatan alat berada pada Bab III ini. Hal-hal yang dibahas pada bab ini diperlukan untuk mempermudah dalam proses pengerjaan tugas akhir, karena berisi urutan proses pengerjaan alat sehingga proses pengerjaan lebih teratur.

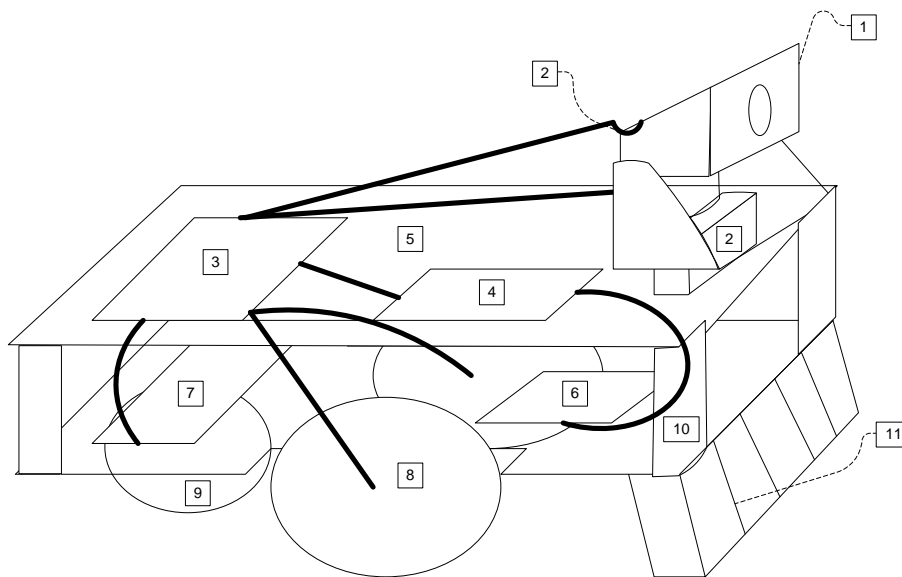
#### 3.1. PERANGKAT KERAS

Rancangan sistem secara keseluruhan pada Telerobot Beroda Pembersih Lantai dapat dilihat pada blok diagram gambar 3.1.



Gambar 3.1. Diagram Blok Perancangan Sistem

Diagram blok perancangan sistem pada gambar 3.1 terdapat empat buah pembagian diantaranya yaitu Sumber Tegangan, *Input*, Proses dan *Output*. Pertama pada bagian input terdapat PC yang terdapat halaman *web* untuk mengendalikan telerobot. Input dilakukan manual oleh pengguna dengan menekan tombol-tombol perintah di halaman *web*. Kemudian dalam bagian proses, data yang dikirimkan dari halaman *web* tersebut ditangkap oleh *access point* yang kemudian diteruskan ke WiFi USB pada *Raspberry Pi*. Pada *Raspberry Pi* ini data di olah dan diteruskan ke *Arduino Uno R3* serta memberikan *output* pada modul kamera *Raspberry Pi*. Sehingga pada bagian *output* terdapat motor DC dan motor servo SG90. Pada motor DC, sebelum motor DC dapat berfungsi maka terdapat masukan dari IC L298 yang digunakan sebagai driver motor DC. Bagian terakhir yaitu sumber tegangan. Sumber tegangan yang digunakan untuk menjalankan telerobot ini adalah Baterai Lipo dan *Power Bank*. Dimana *Power Bank* digunakan untuk mencatu *Raspberry Pi*, sedangkan baterai lipo digunakan untuk mencatu *Arduino Uno R3* dan motor servo.



Gambar 3.2. Desain Rancang Bangun Telerobot Beroda Pembersih Lantai  
(Tampak Depan)

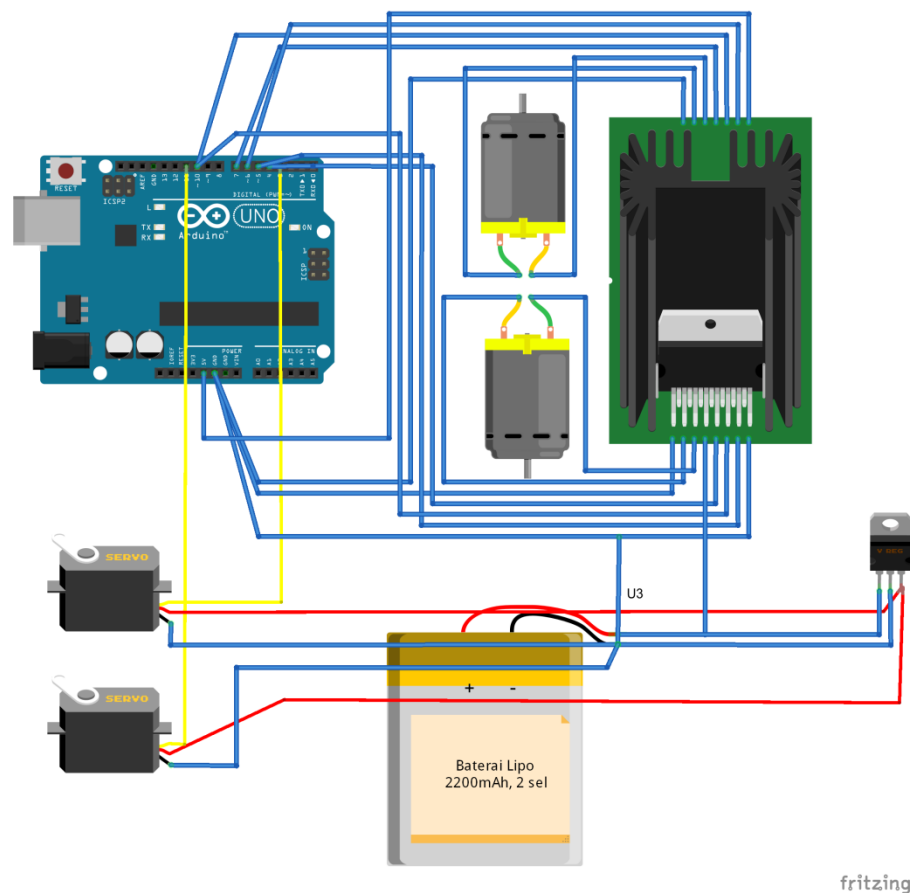
Keterangan desain rancang bangun telerobot pembersih lantai ini pada gambar 3.2. sebagai berikut:

1. Modul Kamera Raspberry Pi
2. Motor Servo SG90
3. Arduino Uno R3 dan Driver Motor
4. Raspberry Pi Model B+

5. Akrilik
6. Power Bank
7. Baterai Lipo
8. Motor DC dan Roda
9. Roda Statis
10. Penyangga dengan menggunakan *spacer*.
11. Ilustrasi Sapu

Desain roda menggunakan 3 buah roda, dengan sebuah roda statis. 2 buah roda terhubung dengan motor DC, dengan masukan logika berasal dari driver motor DC. Penggunaan driver motor DC ini yang dapat menyebabkan pergerakan motor DC untuk maju, mundur, bergerak ke kanan dan ke kiri. Sehingga cukup menggunakan 2 buah motor DC yang masing-masing terhubung ke roda dan sebuah roda statis.<sup>[20]</sup>

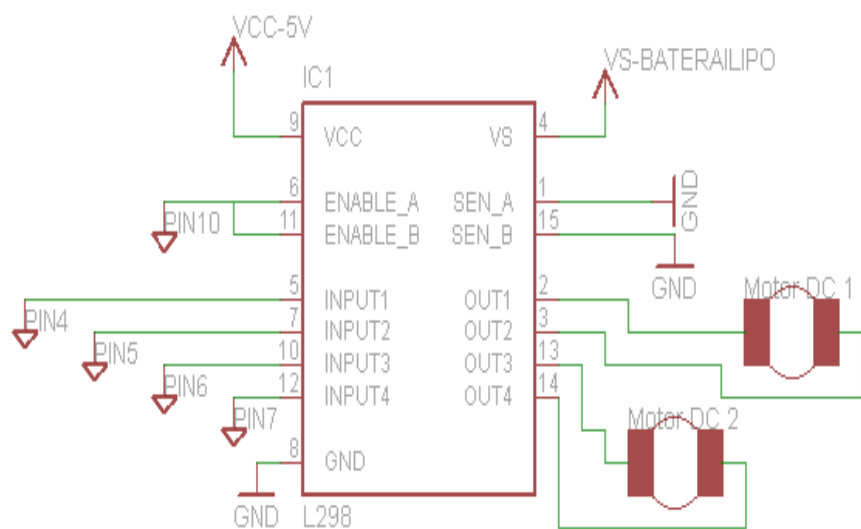
### 3.1.1. Perangkat Keras *Arduino Uno R3*



Gambar 3.3. Rangkaian keseluruhan Arduino Uno R3 (1)

Mikropengendali yang digunakan menggunakan *Arduino Uno R3*. *Arduino Uno R3* ini terhubung dengan pengendali berupa motor DC dan motor Servo, yang masing-masing terdiri dari dua buah. Motor DC yang terhubung pada *Arduino* tidak langsung terhubung, melainkan melalui *driver motor IC L298*. Sedangkan untuk motor servo terhubung langsung pada *Arduino Uno R3*. Rangkaian keseluruhan dengan menggunakan aplikasi fritzing dapat dilihat pada gambar 3.3.

### 3.1.1.1. Perancangan Rangkaian Motor DC



Gambar 3.4. Rangkaian Motor DC

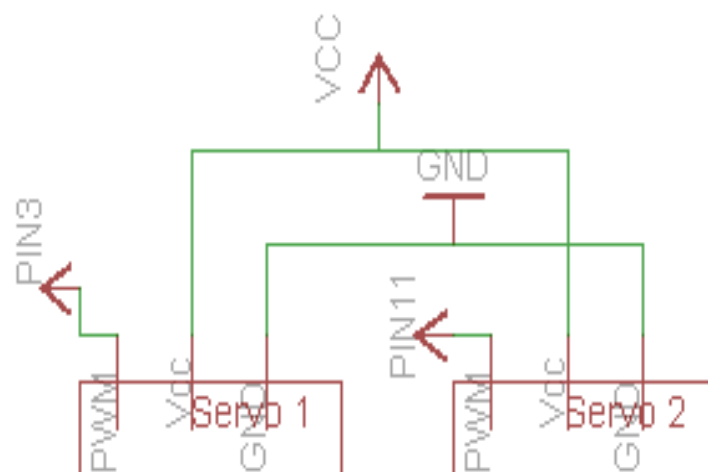
Motor DC digunakan untuk menjalankan telerobot dalam kondisi bergerak maju, mundur, belok kanan, belok kiri dan *stop*. Motor DC yang digunakan adalah motor DC dengan ukuran tegangan 6V dan kecepatan 100 RPM. Karena telerobot membutuhkan roda untuk berjalan, maka dibutuhkan dua buah motor DC yang terhubung dengan roda telerobot. Pada gambar 3.4 untuk menjalankan motor DC menggunakan IC L298 sebagai *driver* penggerak motor. IC L298 memiliki 15 pin, yang terletak pada bagian depan 8 pin dan bagian belakang 7 pin.

Pada bagian *Arduino Uno R3*, bagian dari IC L298 yang terhubung dengan *arduino* adalah pin 4, 5, 6, 7, 10, 5V dan GND. Dimana pin 4 di *arduino* terhubung dengan kaki pin 5 IC L298,

untuk pin 5 terhubung ke kaki pin 7, untuk pin 6 terhubung ke kaki pin 10 dan untuk pin 7 terhubung ke kaki pin 12. Keempat kaki pin tersebut merupakan pin input dari IC L298. Sedangkan pada kaki pin 6 dan 11 IC L298 terhubung ke pin 10 *arduino*, dimana pin ini merupakan pin *enable*. Selain itu yang masih terhubung dengan pin *arduino* lain adalah untuk *Vcc* dan *Ground*, dimana pin *Vcc arduino* terhubung dengan kaki pin 9 IC L298 dan pin *ground arduino* terhubung dengan kaki pin 1, 8 dan 15 IC L298. Untuk kaki pin 2, 3, 13, 14 terhubung pada dua buah motor DC, karena kaki pin ini berfungsi sebagai *output* dari IC L298. Serta kaki pin 4 yang berfungsi untuk *Vs* dihubungkan langsung ke *Vs* di baterai Lipo.

Kecepatan yang dipergunakan untuk kedua motor DC tersebut menggunakan batas kecepatan maksimal yaitu saat tegangan maksimum berupa desimal 255 dan bata kecepatan minimum yaitu desimal 0. Dengan pengaturan yang diatur untuk kecepatan maksimal adalah desimal 200.

### 3.1.1.2. Perancangan Rangkaian Motor Servo Sg90



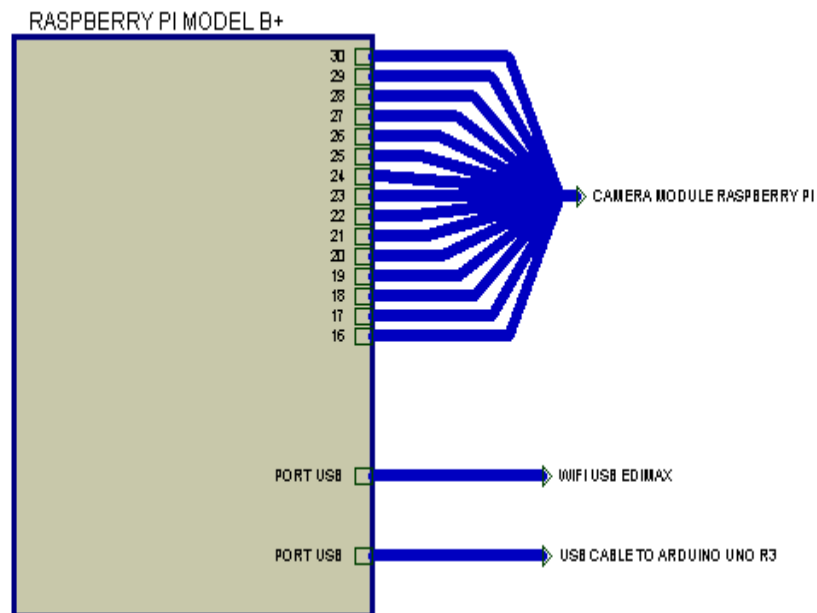
Gambar 3.5. Rangkaian Motor Servo SG90

Motor Servo pada gambar 3.5 digunakan untuk mengatur pergerakan kamera dengan aksi pergeseran ke kanan, kiri, atas dan bawah. Jenis motor servo yang digunakan adalah *micro motor*

*servo* SG90, dengan jumlah pin yang dimiliki sebanyak 3 buah terdiri dari pin PWM pada kabel berwarna oranye, pin Vcc pada kabel berwarna merah dan pin *Ground* pada kabel berwarna coklat. Pin PWM motor servo pertama dihubungkan pada pin 3 arduino dan untuk motor servo kedua dihubungkan pada pin 11. Untuk pin Vcc kedua motor servo tersebut dihubungkan dengan pin Vcc 5V pada pin *arduino Uno R3*, sedangkan *ground* terhubung satu sama lain dengan pin ground di pin *arduino Uno R3*.

Motor servo pertama digunakan untuk melakukan pergerakan kamera ke arah kanan, dan kiri dengan sudut masing-masing pergerakan sudut adalah  $90^\circ$ , begitu juga untuk pergerakan kamera ke arah atas dan bawah menggunakan pergerakan sudut  $90^\circ$ .

### 3.1.2. Perangkat Keras *Raspberry Pi Model B+*



Gambar 3.6. Skematik keseluruhan *Raspberry Pi Model B+*

Gambar 3.6 merupakan skematik keseluruhan dari *Microprocessor Raspberry Pi model B+*. Bagian-bagian yang terhubung dengan *Raspberry Pi model B+* ini adalah Modul Kamera *Raspberry Pi*, *WiFi USB Edimax*

serta kabel USB ke *Arduino Uno R3*. Modul Kamera dihubungkan menggunakan kabel khusus dari modul kamera tersebut ke *port* khusus modul kamera di *Raspberry Pi*. *Port* modul kamera raspberry terdiri dari 15 pin, pin yang terhubung adalah pin 15 hingga pin 30. Pada *Raspberry Pi* terdapat 4 buah *port*, namun *port* yang digunakan hanya dua buah *port* USB. *Port* USB *Raspberry Pi* pertama dihubungkan dengan *WiFi* USB Edimax, dan *port* USB *Raspberry Pi* kedua dihubungkan dengan kabel USB yang terhubung dengan *Arduino Uno R3*.

## 3.2. PERANGKAT LUNAK

### 3.2.1. Flowchart Motor DC

Motor DC pada perancangan perangkat lunak dilakukan pada perangkat lunak *Arduino IDE*. Proses yang dilakukan pada *Arduino IDE* dijelaskan pada *flowchart* gambar 3.7

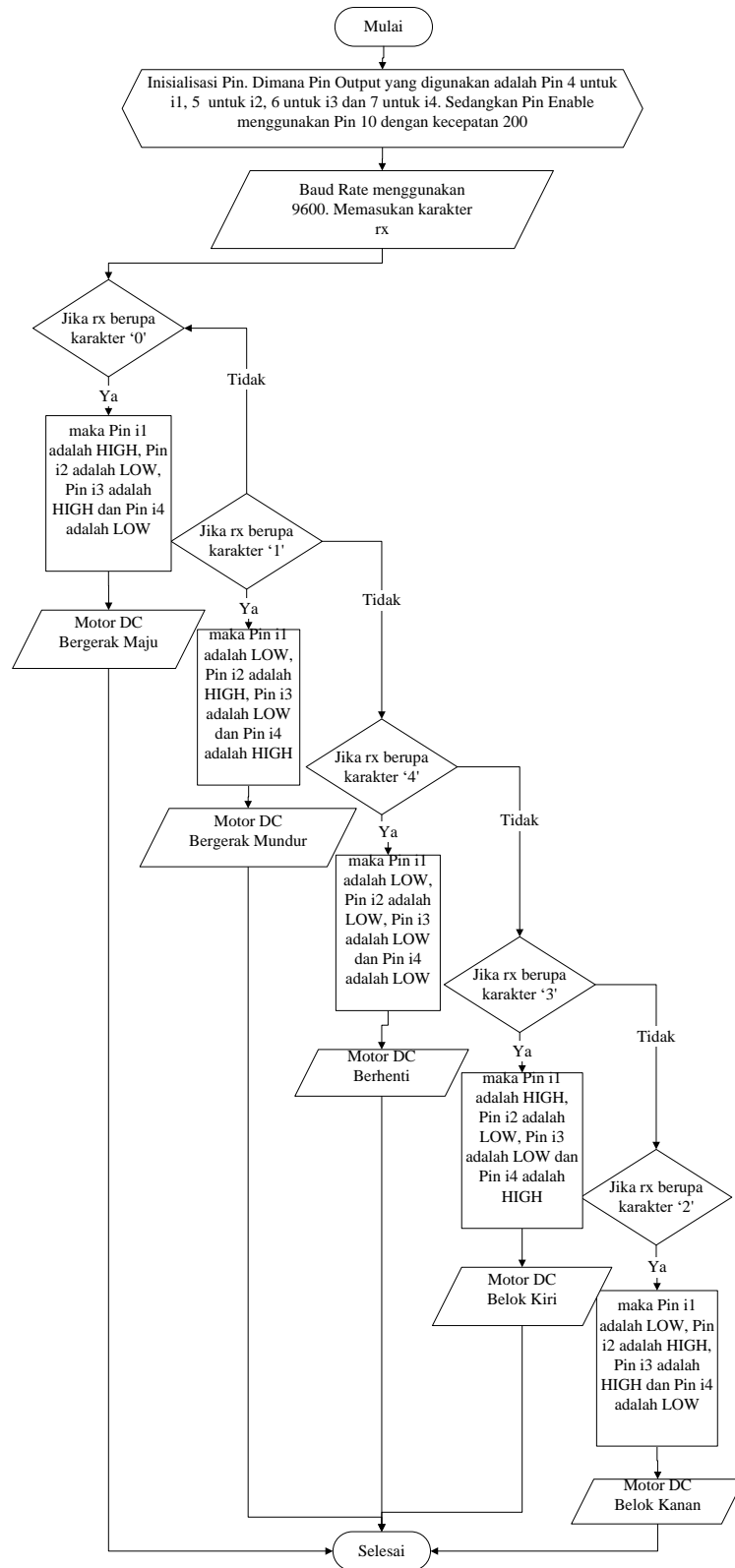
*Flowchart* gambar 3.7 pertama dimulai dengan memasukan data awal Pin i1, Pin i2, Pin i3 dan Pin i4. *Baud rate* yang digunakan adalah 9600. Perulangan pertama yang dilakukan adalah untuk menghasilkan motor DC bergerak maju, berikut ini potongan *syntax* yang digunakan untuk pergerakan maju motor DC.

```
if (rx=='0') { //UNTUK MAJU MOTOR DC
    i1=HIGH;
    i2=LOW;
    i3=HIGH;
    i4=LOW;
}
```

Agar motor DC dapat bergerak maju, maka harus mengirimkan karakter 0, dan *syntax* yang digunakan agar motor DC bergerak mundur adalah sebagai berikut.

```
else if (rx=='1') { //UNTUK MUNDUR MOTOR DC
    i1=LOW;
    i2=HIGH;
    i3=LOW;
    i4=HIGH;
```

}



Gambar 3.7. Flowchart Arduino untuk Motor DC



Untuk berikutnya, jika motor DC berhenti, maka mengirimkan karakter 4, dengan *syntax* seperti dibawah ini.

```
else if (rx=='4') { //UNTUK BERHENTI MOTOR DC
    i1=LOW;
    i2=LOW;
    i3=LOW;
    i4=LOW;
}
```

Saat motor DC melakukan pergerakan ke kiri, maka karakter yang dikirimkan adalah 3, dengan *syntax* pada *arduino* IDE sebagai berikut.

```
else if (rx=='3') { //UNTUK BELOK KIRI MOTOR DC
    i1=HIGH;
    i2=LOW;
    i3=LOW;
    i4=HIGH;
}
```

Sedangkan untuk pergerakan motor DC ke kanan, maka karakter yang dikirimkan adalah 2, dengan *syntax* seperti dibawah ini.

```
else if (rx=='2') { //UNTUK BELOK KANAN MOTOR DC
    i1=LOW;
    i2=HIGH;
    i3=HIGH;
    i4=LOW;
}
```

Data serial berupa karakter yang dikirimkan dari *arduino* ini kemudian dikirimkan lagi ke *Raspberry Pi*, dimana pada bagian *Raspberry Pi* data serial di cocokan dengan bahasa pemrograman *Python*. Prinsip *Python* ini dijadikan sebagai antarmuka antara *Arduino* dengan *Raspberry Pi*. *Flowchart* yang digunakan untuk Motor DC pada bahasa pemrograman *python* dapat dilihat pada gambar 3.8.

Pertama pada halaman web terdapat data berupa tombol-tombol, dimana masing- masing tombol tersebut memiliki fungsi masing-masing

yang sudah disinkronkan dengan python serta *Arduino* IDE. Pada fungsi pertama ketika motor DC difungsikan untuk maju, maka menggunakan syntax sebagai berikut.

```
if message == "ud":
```

```
c = "0";
```

dengan pesan ud yang berarti *up dc*, sedangkan untuk proses selanjutnya seperti pada *syntax* berikut ini.

```
print c
```

```
if c == '0' :
```

```
print "Running Forward"
```

Kedua, apabila motor DC ingin difungsikan untuk mundur, maka *syntax* yang digunakan sebagai berikut.

```
if message == "dd":
```

```
c = "1"
```

dengan d berarti *down dc*, atau difungsikan untuk mundur, dan karakter yang dikirimkan ke *arduino* IDE adalah 1. *Syntax* selanjutnya di *python* untuk membuat maju motor DC menggunakan *syntax* berikut ini.

```
elif c == '1' :
```

```
print "Running Reverse"
```

Ketiga, untuk membuat fungsi belok kiri pada motor DC menggunakan *syntax* berikut sebagai inisialisasi.

```
if message == "ld":
```

```
c = "3"
```

dengan pesan ld merupakan left dc, dan karakter serial yang dikirimkan adalah 3. *Syntax* lanjutan yang membuat fungsi motor DC bergerak ke kiri sebagai berikut.

```
elif c == '3' :
```

```
print "Turning Left"
```

Keempat, fungsi yang digunakan adalah untuk membuat motor DC bergerak ke kanan. *Syntax* identifikasi yang dipergunakan adalah sebagai berikut.

```
if message == "rd":
```

```
c = "2"
```

pesan yang dikirimkan untuk pergerakan motor DC ke kanan adalah rd yang merupakan *right dc* atau kanan, serta karakter serial yang dikirimkan adalah 2 untuk disinkronkan dengan program di *Arduino IDE*. Syntax selanjutnya pada Python untuk pergerakan ke kanan seperti dibawah ini.

```
elif c == '2' :
```

```
    print "Turning Right"
```

Terakhir, untuk membuat motor DC berhenti maka menggunakan *syntax* sebagai berikut.

```
if message == "sd":
```

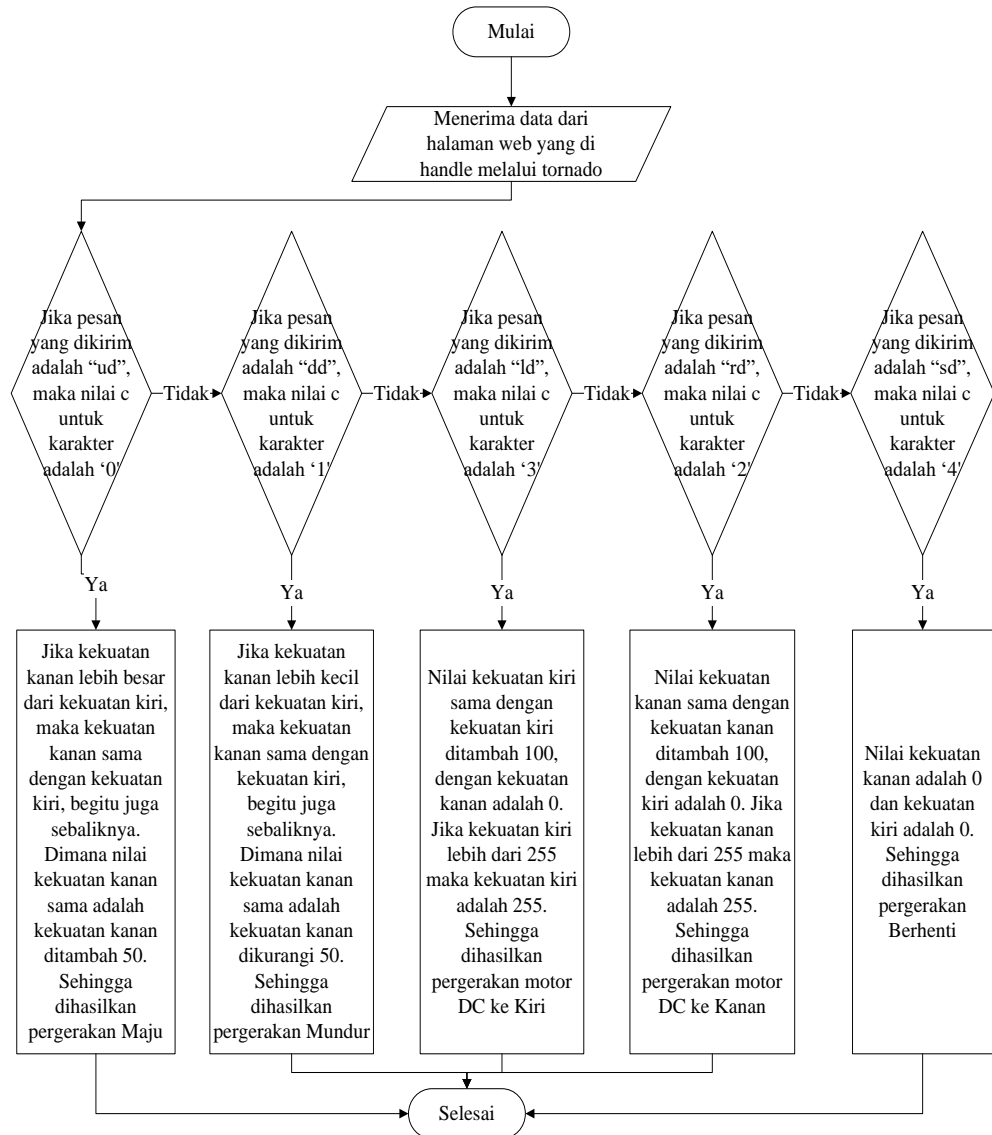
```
c = "4"
```

pesan sd pada *syntax* identifikasi tersebut adalah *stop dc* atau berhenti, sehingga pada halaman *web* yang dikirimkan adalah huruf sd, sedangkan pada *python* ke *Arduino IDE* mengirimkan angka serial berupa karakter 4 agar keduanya saling sinkron. Syntax lanjutan untuk membuat berhenti motor DC adalah sebagai berikut.

```
elif c == '4' :
```

```
    print "Stopped"
```

Pada Tabel 3.1 merupakan pemetaan yang dilakukan untuk mempermudah pembuatan perancangan komunikasi dari halaman *web* ke *arduino uno* melalui *raspberry pi*. Pemetaan dilakukan untuk data karakter, halaman web serta python, dimana pemetaan tersebut memiliki fungsi masing-masing. Data karakter yang digunakan berupa angka untuk menggerakkan motor DC. Angka-angka tersebut adalah 0, 1, 2, 3, dan 4. Sedangkan untuk halaman *web* karakter yang dikirimkan berupa huruf. Huruf-huruf tersebut adalah ud, dd, rd, ld, dan sd. Dimana kelimanya berfungsi untuk maju motor DC, mundur motor DC, belok kanan motor DC, belok kiri motor DC serta berhenti motor DC.



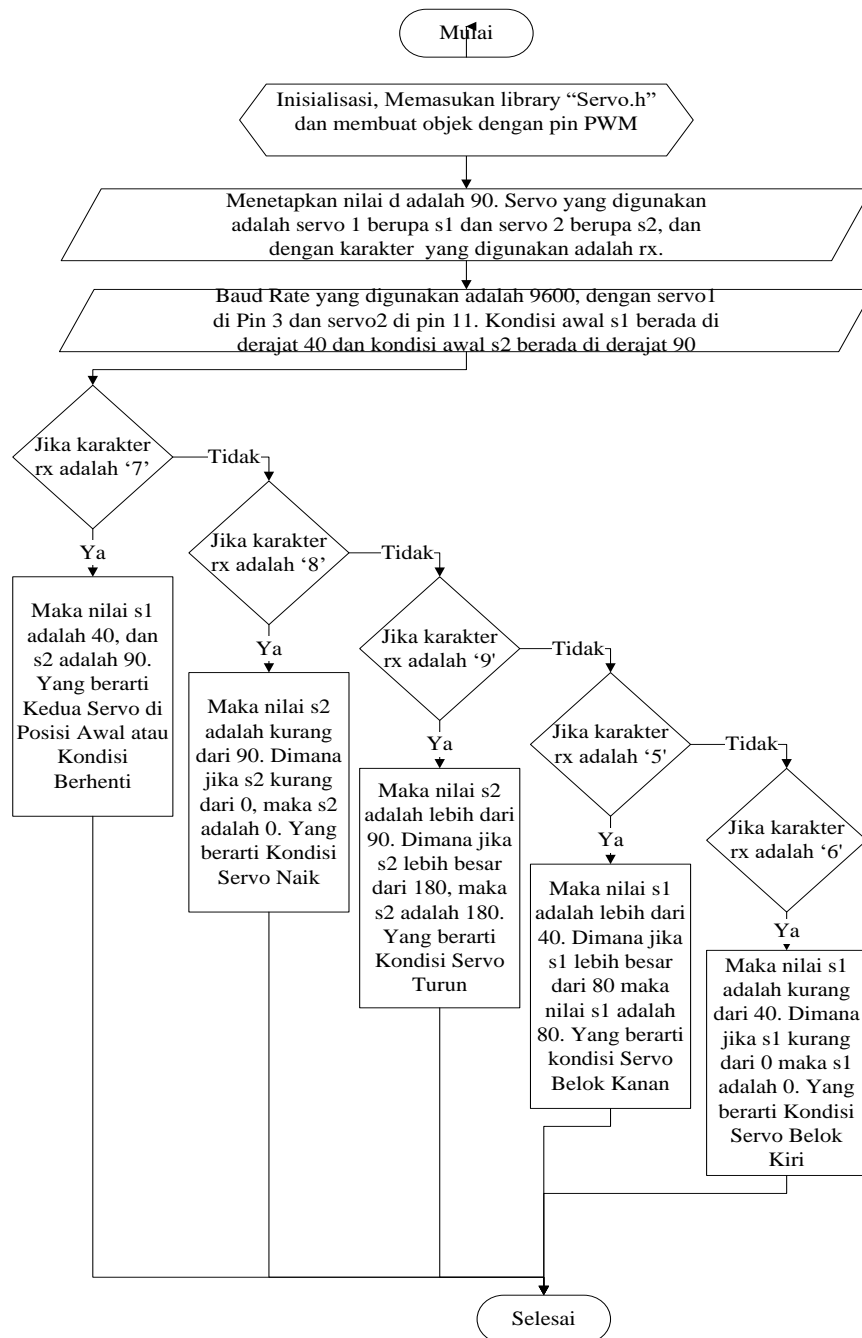
Gambar 3.8 Flowchart motor DC pada bahasa pemrograman Pyton

Tabel 3.1 Pemetaan Data Serial untuk kendali Motor DC

Data (Char)	Fungsi	Halaman Web	Python
0	Maju Motor DC	Ud	ud -> 0
1	Mundur Motor DC	Dd	dd -> 1
2	Belok Kanan Motor DC	Rd	rd -> 2
3	Belok Kiri Motor DC	Ld	ld ->3
4	Berhenti Motor DC	Sd	sd -> 4

### 3.2.2. Flowchart Motor Servo

Dalam mengoperasikan motor Servo dibutuhkan bahasa pemrograman *arduino* untuk mengendalikan motor servo dalam segi perangkat keras, *flowchart* yang menggambarkan proses kerja motor servo pada *arduino* IDE ditunjukkan pada gambar 3.9. Kemudian pemetaan juga terdapat pada tabel 3.2. Pemetaan ini berupa data-data yang dikirim dari Halaman *Web* hingga Servo dapat bekerja dengan kendali dari halaman *web*.



Gambar 3.9. Flowchart Motor Servo pada Arduino

Pada bahasa pemrograman *Arduino* IDE, untuk servo sudah memiliki *library* tersendiri, sehingga hanya memasukan logika yang akan dilakukan oleh motor servo. *Library Servo* tersebut dituliskan dengan *syntax* berikut ini.

```
#include <Servo.h>
```

Agar servo dapat diatur sesuai perintah dari halaman *web*, maka harus ada penyesuaian program antara *Arduino* IDE dengan Halaman *Web*. Pada servo1 agar motor servo dapat bergerak ke kanan sebanyak 90 derajat, dengan pergeseran tiap klik pada halaman *web* menggunakan *syntax* berikut.

```
else if (rx=='5'){ //servo kanan  
    s1++;  
    if (s1>=80) s1=80;  
}
```

Pada *syntax* tersebut karakter rx adalah 5, yang berarti angka 5 di *Arduino* IDE dihubungkan dengan bahasa pemrograman *python* untuk mengidentifikasi pergerakan yang dilakukan dari halaman web. Kemudian untuk servo1, apabila ingin bergerak ke kiri sebesar hingga 90 derajat menggunakan *syntax* berikut.

```
else if (rx=='6'){ //servo kiri  
    s1--;  
    if (s1<=0) s1=0;  
}
```

Untuk program tersebut karakter yang dikirimkan adalah 6. Selanjutnya pada servo1 dan servo 2 untuk membuat servo berhenti dalam kondisi awal maka menggunakan *syntax* berikut ini.

```
if (rx=='7'){ //servo berhenti  
    s1=40;  
    s2=90;  
}
```

Pada *syntax* tersebut nilai servo1 kembali di posisi 40 derajat, dan karakter yang dikirimkan adalah 7. Kemudian untuk servo2 juga dilakukan

hal yang sama untuk pergerakan servo2 agar kembali ke posisi semula maka nilai s2 adalah 90, dan karakter yang dikirimkan adalah 7.

Untuk pergerakan servo ke kanan menggunakan servo 2, dengan karakter yang dikirimkan adalah 5. Berikut ini potongan *syntax* yang digunakan untuk pergerakan servo ke kanan.

```
else if (rx=='5'){ //servo kanan
    s1++;
    if (s1>=80) s1=80;
}
```

Untuk servo 2 agar dapat bergerak ke kiri dengan posisi servo di minus, dan karakter yang dikirimkan adalah 6 seperti pada *syntax* berikut ini.

```
else if (rx=='6'){ //servo kiri
    s1--;
    if (s1<=0) s1=0;
}
```

Selain itu untuk menghubungkan antara program dari *Arduino* IDE ke halaman *web* dibutuhkan bahasa pemrograman *python* sebagai antarmukanya. Gambar 3.10 menjelaskan *flowchart* motor Servo pada *Python*. Pengiriman pesan pertama untuk bahasa *python* ini adalah pesan untuk motor servo 1 agar bergerak ke kiri, dengan menggunakan *syntax* berikut.

```
if message == "ls":
    c = "6";
```

pesan yang dikirimkan adalah ls yang berarti left servo, dan karakter yang dikirimkan adalah 6, sedangkan untuk proses selanjutnya seperti pada *syntax* berikut ini.

```
    print c
    if c == '6' :
        print "Running Forward"
```

Selanjutnya agar motor servo 1 dapat bergerak ke kanan maka *syntax* yang digunakan sebagai berikut.

```
if message == "rs":
```

```
c = "5"
```

pesan yang dikirimkan dari halaman web adalah rs yang berarti right servo, dengan angka yang dikirimkan ke *arduino* IDE adalah 5. Berikut ini *syntax* selanjutnya di *python* untuk menggerakkan motor servo 1 ke kiri.

```
elif c == '5' :
```

```
print "Running Reverse"
```

Kemudian untuk membuat berhenti atau kembali ke posisi semula motor servo 1 dan servo 2, dibutuhkan *syntax* berikut ini.

```
if message == "ss":
```

```
c = "7"
```

pesan ss yang dikirimkan dari halaman web berarti stop servo atau berhenti, sedangkan angka yang dikirimkan ke *arduino* IDE adalah 7. *Syntax* lanjutan untuk membuat motor servo 1 dan servo 2 berhenti adalah sebagai berikut.

```
elif c == '7' :
```

```
print "Stopped"
```

Selanjutnya pada motor servo 2 juga dilakukan pemrograman yang sama, hanya saja pesan serta angka yang dikirimkan berbeda, seperti pada *syntax* berikut ini.

```
if message == "us":
```

```
c = "8"
```

*Syntax* tersebut digunakan untuk membuat motor servo bergerak ke kanan, namun pemasangan yang dilakukan pada servo 2 terletak di atas motor servo 1 sehingga servo 2 dibuat seolah-olah bergerak ke atas dan ke bawah. Sehingga dapat dikatakan pesan us atau *up servo* berarti Naik Servo karena pesan yang dikirimkan adalah us dengan karakter yang dikirimkan ke *Arduino* IDE adalah 8. *Syntax* selanjutnya pada *python* untuk pergerakan motor servo 2 sebagai berikut.

```
elif c == '8' :
```

```
print "Turning Left"
```

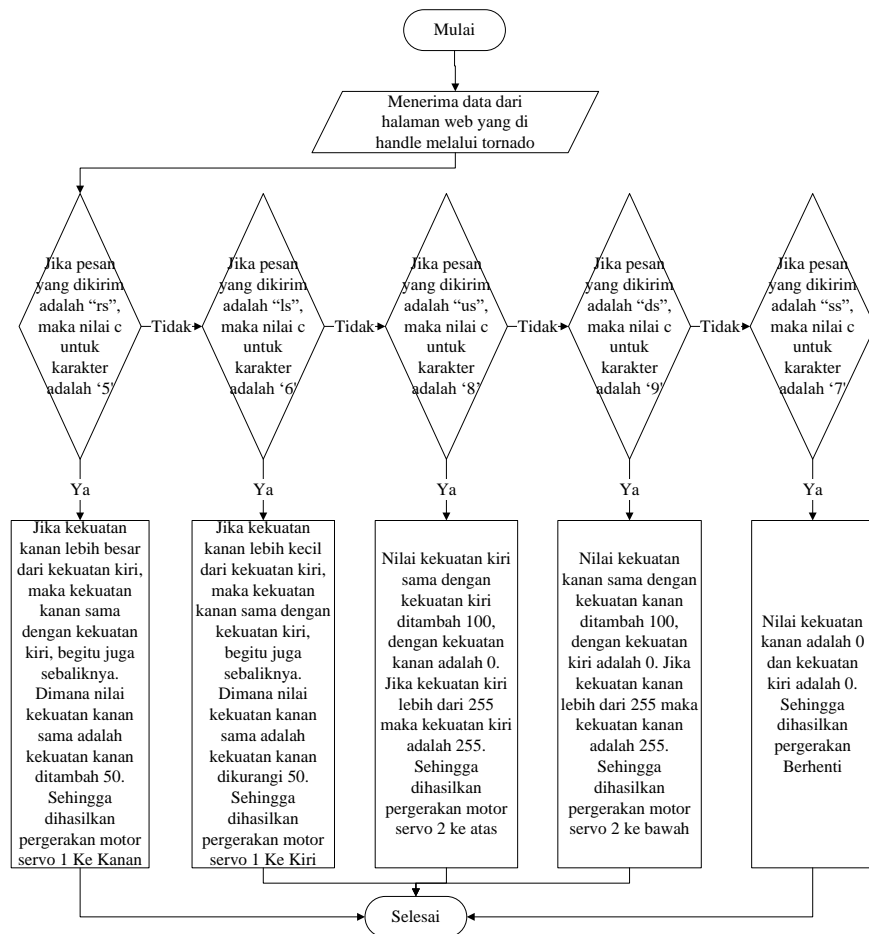


Kemudian agar motor servo 2 dapat melakukan pergerakan ke bawah atau turun, maka dibutuhkan *syntax* berupa ds yang berarti *down servo* dari halaman *web*, serta angka 9 yang kemudian dikirimkan ke Arduino IDE, berikut ini adalah *syntax* pergerakan motor servo 2 ke bawah.

```
if message == "ds":
    c = "9"
```

Selanjutnya berikut ini adalah *syntax* lanjutan untuk pergerakan motor servo 2 ke bawah.

```
elif c == '9':
    print "Turning Right"
```



Gambar 3.10 Flowchart motor servo pada bahasa pemrograman Python

Tabel 3.2 Pemetaan Data Serial untuk kendali Motor Servo

Data (Char)	Fungsi	Halaman Web	Python
5	Kanan Servo 1	Rs	rs -> 0
6	Kiri Servo 1	Ls	ls -> 1

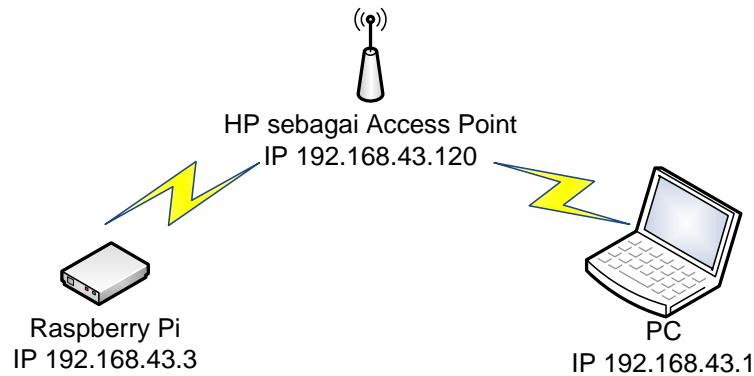
7	Berhenti Servo 1 & 2	Ss	ss -> 2
8	Naik Servo 2	Us	us ->3
9	Turun Servo 2	Ds	ds -> 4

### 3.3 PERANCANGAN JARINGAN WIFI

Pengendalian telorobot pembersih lantai ini memanfaatkan jaringan *WiFi* sebagai media transmisi pengiriman data antara *Raspberry Pi* dengan halaman *web*. Dari sisi perangkat keras yaitu *Arduino Uno R3* berkomunikasi dengan *Raspberry Pi* menggunakan komunikasi serial. Dari *Raspberry Pi* ke halaman *web* menggunakan *Access Point* sebagai *interface* dan jenis jaringan *WiFi* yang digunakan adalah *Wireless Local Area Network (WLAN)*. *Access Point* yang digunakan dapat menggunakan *access point* dari *handphone* atau biasa disebut dengan *thetering* atau dapat menggunakan *personal computer* sebagai *access point*-nya. Pada jaringan *WLAN* ini *IP* yang digunakan agar kedua perangkat dapat saling berkomunikasi menggunakan *IP Static*.

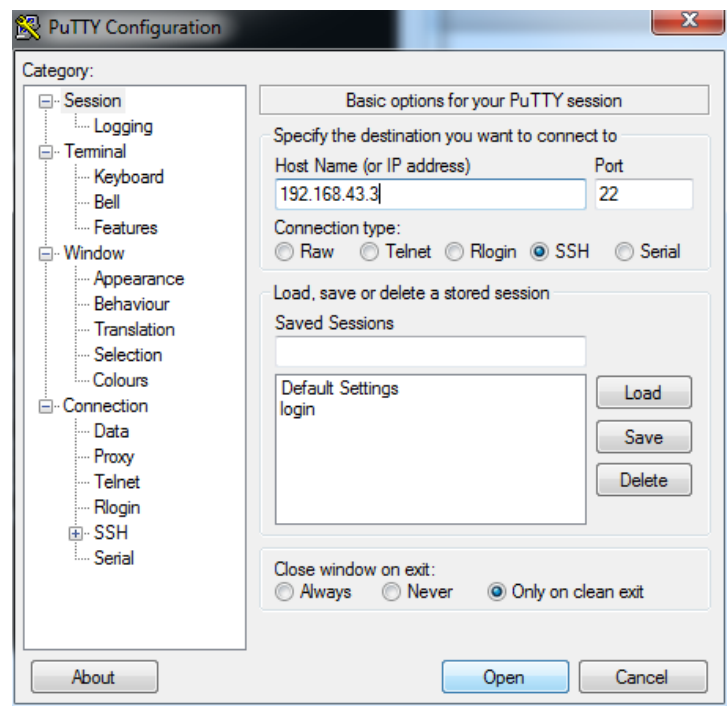
Konfigurasi jaringan yang dipergunakan adalah infrastruktur dengan jumlah client sebanyak satu buah pada laptop. Dengan jaringan infrastruktur ini, maka koneksi yang dibentuk adalah *Point-to-point*. Dimana satu perangkat dihubungkan ke satu perangkat lain melalui media transmisi *wireless*.

Pertama, jika *access point* yang digunakan menggunakan *handphone*, maka struktur jaringan yang digunakan seperti pada gambar 3.10. *IP* yang digunakan pada *Raspberry Pi* adalah 192.168.43.3, untuk PC sebagai pengendali adalah 192.168.43.1 dan untuk *handphone* adalah 192.168.43.120. *IP* pada *handphone static*, sehingga *IP* yang digunakan untuk *Raspberry Pi* dan PC harus mengikuti *network* pada *IP* di *handphone* dan menggunakan jenis *IP Static*. Kedua jika *access point* yang digunakan menggunakan PC maka *IP* yang digunakan juga menggunakan *IP static*. *IP Static* yang digunakan sudah ditentukan pengaturannya dari *raspberry pi*, sehingga alamat *IP* yang digunakan untuk semua perangkat sama dengan alamat *IP* yang digunakan jika menggunakan *handphone* sebagai *access point*. Penggunaan *access point* menggunakan PC merupakan cara lain apabila signal pada *handphone* kurang maksimal.

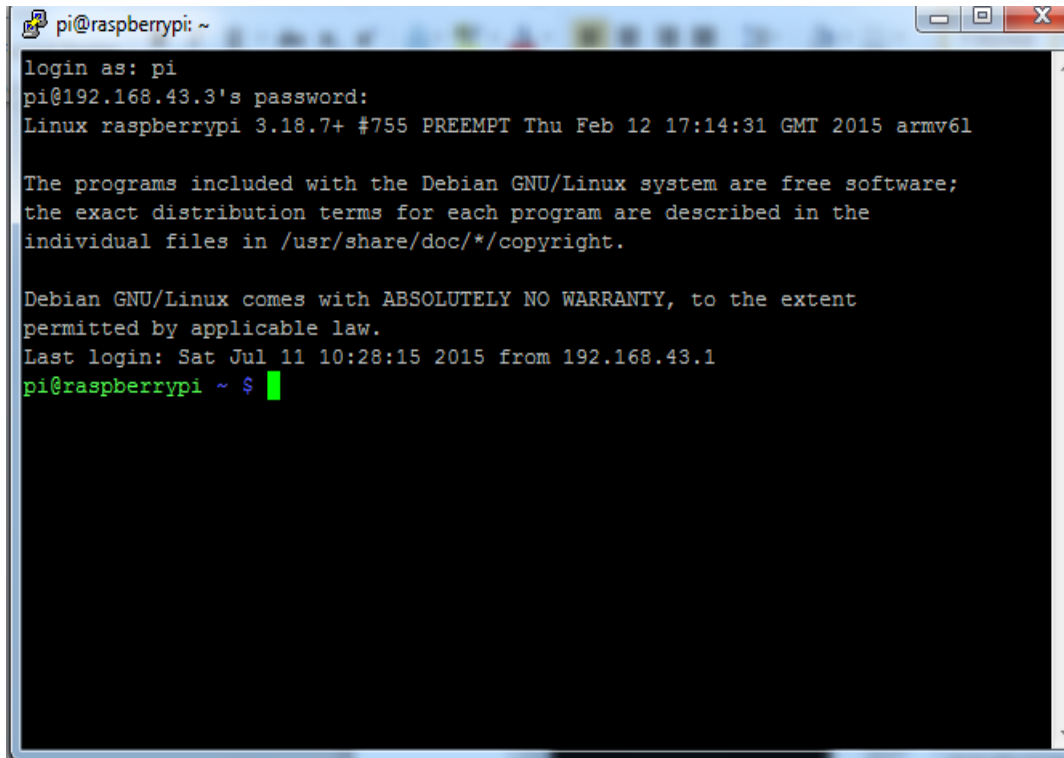


Gambar 3.10 Struktur Jaringan WiFi dengan Handphone sebagai Access Point

Pada gambar 3.11 terdapat konfigurasi PuTTY untuk membuka Raspberry Pi dari windows. PuTTY disini digunakan seagai SSH, dengan memasukan alamat IP Raspberry Pi yaitu 192.168.43.3. Sedangkan pada gambar 3.12 merupakan tampilan awal SSH Raspberry Pi untuk membuka Raspberry Pi. Saat masuk ke Raspberry Pi maka harus memasukan username serta password. Pada login as menggunakan "pi", dan untuk `pi@192.168.43.3's` password yang digunakan menggunakan "raspberry".



Gambar 3.11. PuTTY Configuration



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.43.3's password:  
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv61  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat Jul 11 10:28:15 2015 from 192.168.43.1  
pi@raspberrypi ~ $
```

Gambar 3.12. Tampilan SSH *Raspberry Pi*

