

BAB IV

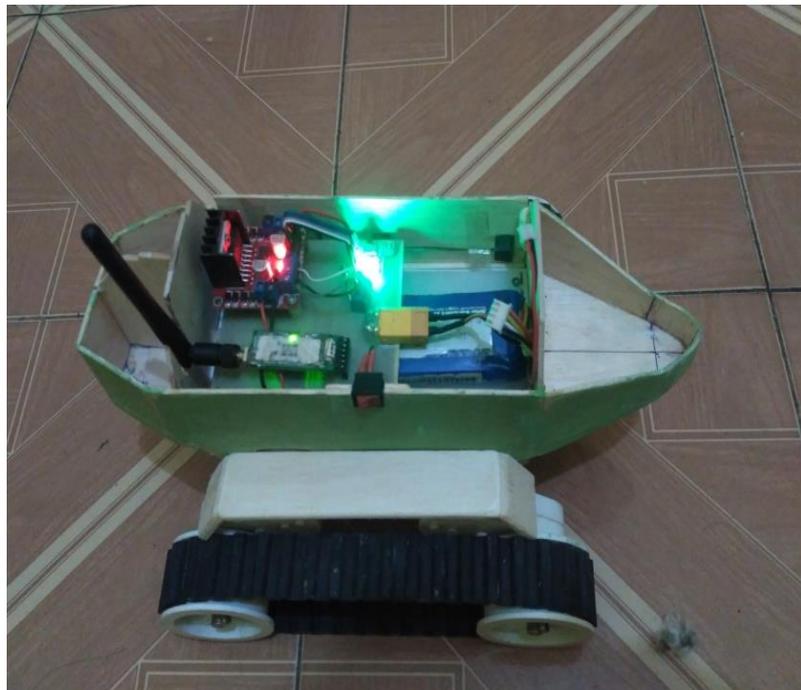
HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

4.1.1 Hasil Tampilan Robot Tank Rover5

Robot tank Rover 5 ini dirancang dari beberapa komponen agar robot dapat beroperasi sebagaimana mestinya, beberapa komponen yang digunakan dalam membangun robot tank adalah sebagai berikut:

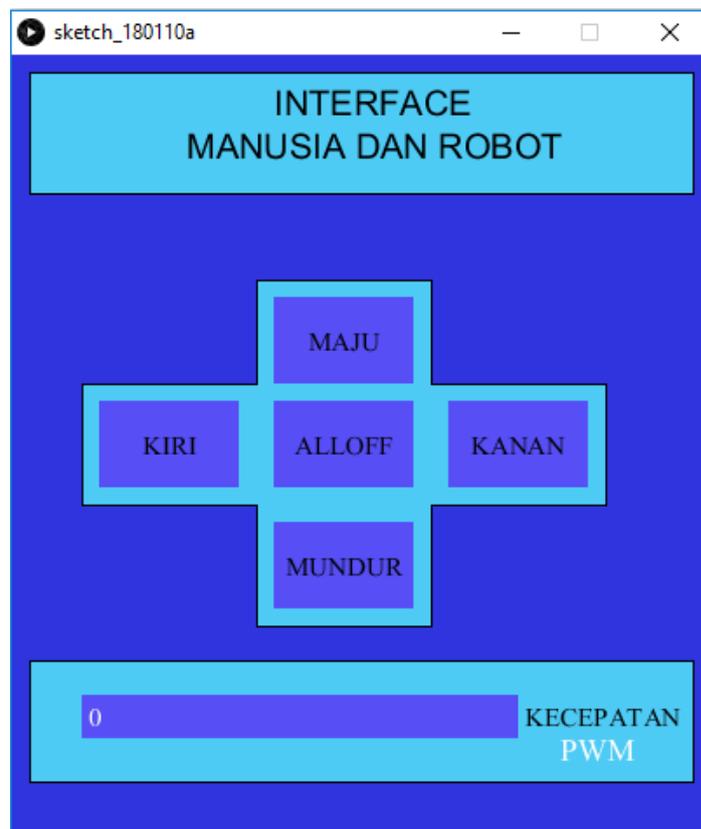
- A. Arduino Uno R3
- B. Module radio telemetry 3DR
- C. 2 buah motor DC
- D. Module Driver L298N
- E. Battery lippo 3 cel 1450 mAh
- F. 1 saklar SPDT
- G. 2 set gear box
- H. Kabel jumper



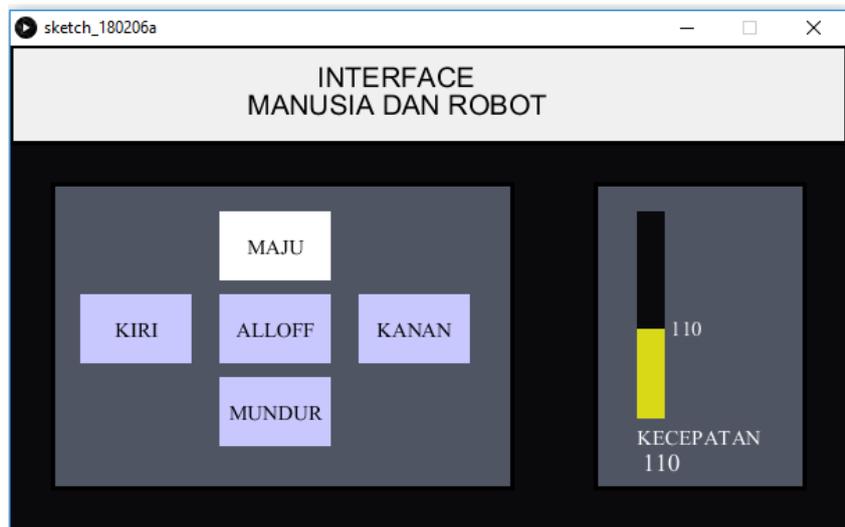
Gambar 4.1 Tampilan Komponen Robot Tank Rover5

4.1.2 Hasil Tampilan Program GUI

Sesuai dengan metode penelitian dalam BAB III bahwa dari tiga versi GUI yang telah dirancang akan dilakukan pengujian terhadap setiap GUI. Sehingga akan didapatkan data yang dapat merepresentasikan GUI mana yang lebih diminati oleh *user*. Berikut merupakan tampilan hasil desain tiga buah versi GUI yang akan disebarakan kepada 25 *user*.



Gambar 4.2 Tampilan GUI Versi 1



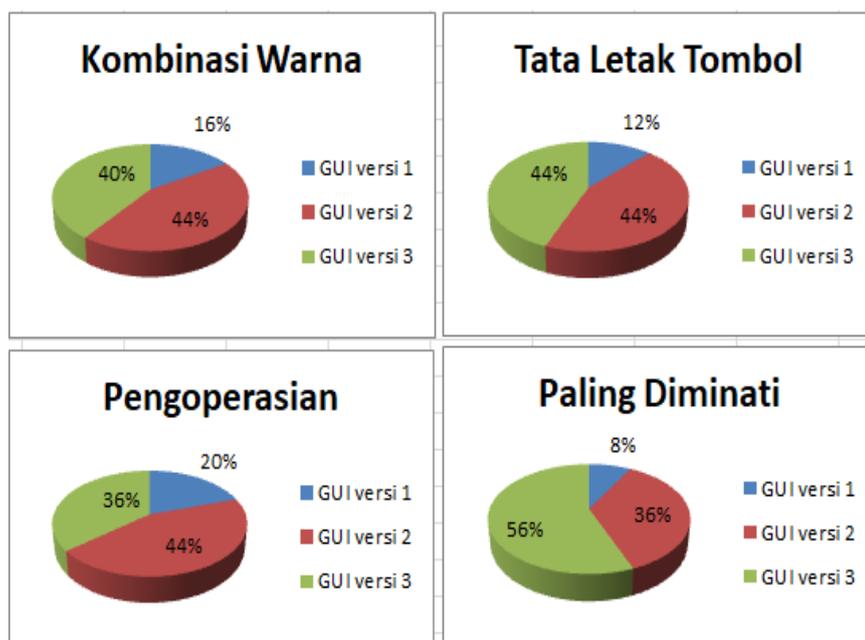
Gambar 4.3 Tampilan GUI Versi 2



Gambar 4.4 Tampilan GUI Versi 3

4.1.3 Hasil Quisioner

Untuk mendapatkan data yang akurat tentang GUI mana yang lebih diminati oleh *user* maka disebarakan quisioner penilaian GUI terhadap 25 orang. Data hasil quisioner tersebut akan disajikan dalam bentuk grafik seperti berikut:



Gambar 4.5 Grafik Hasil Quisioner *User GUI*

Bedasarkan dari tabel 4.5 dapat diamati bahwa mayoritas *user* lebih berminat pada GUI versi 3. Dalam aspek kombinasi warna GUI versi 3 menduduki peminat yang paling banyak yaitu mendapatkan 44 % sedangkan untuk GUI versi 2 memiliki skor 40 % dan untuk GUI versi 1 memiliki jumlah peminat yang paling sedikit hanya 16 % saja.

Dalam aspek tata letak tombol GUI versi 2 dan GUI versi 3 mempunyai nilai yang setara yaitu 44 % sedangkan untuk GUI versi 1 memiliki jumlah peminat yang sedikit hanya sebesar 12% saja.

Walaupun GUI versi 3 lebih banyak diminati diantara GUI-GUI lainnya, namun beberapa *user* menjelaskan bahwa GUI versi 3 lebih sulit dalam pengoperasiannya. Hal ini diakibatkan karena GUI versi 3 menggunakan knop untuk megatur kecepatan. Beberapa *user* lebih mudah menggunakan Slider untuk mengatur kecepatan. Sehingga dalam aspek kemudahan pengoperasian mayoritas *user*

memilih GUI 2 yaitu sebesar 44 %., Sedangkan untuk GUI versi 3 memiliki peminat sebesar 36%., dan GUI versi 1 memiliki peminat sejumlah 20 %.

4.1.4 Hasil Pengujian Tombol GUI

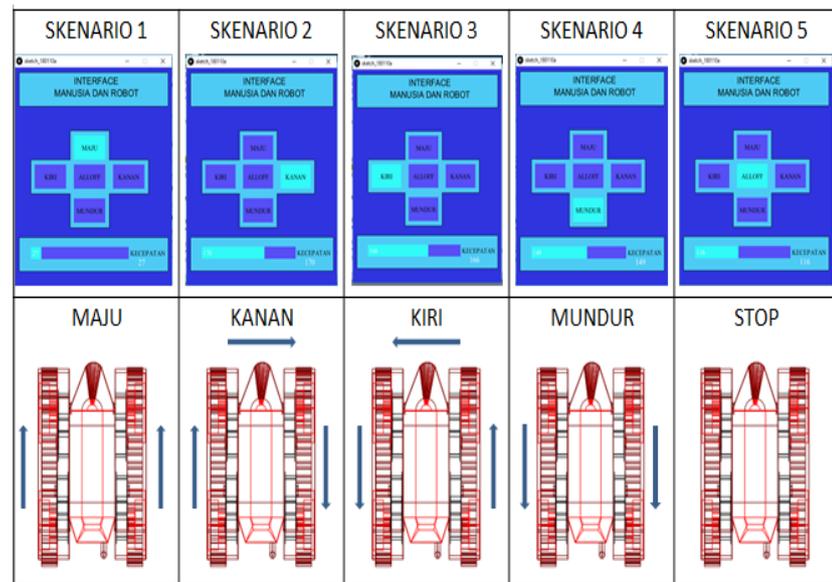
Dalam tahap ini akan dilakukan pengujian kinerja dari GUI. Dalam pengujiannya, akan dilakukan beberapa skenario pengujian. Skenario pengujian 1 adalah pengujian GUI dimana tombol maju GUI akan ditekan dan diberikan beberapa Inputan nilai PWM mulai dari rendah hingga tinggi. Pada skenario ini robot akan membaca perintah dari GUI yang dikirimkan, kemudian melakukan respon untuk bergerak maju dengan kecepatan sesuai dengan nilai kecepatan yang diinginkan.

Skenario ke 2, adalah pengujian tombol kanan pada GUI, tombol kanan akan ditekan dengan memberikan beberapa kondisi nilai PWM dari rendah hingga tinggi dan dilakukan pengamatan pada kondisi putaran *tracking wheels*. Dalam skenario ke dua robot akan bergerak belok ke kanan . Skenario selanjutnya yaitu skenario ke 3. Dalam skenario ke 3 GUI akan ditekan tombol kiri sehingga akan memerintahkan robot untuk bergerak belok ke kiri. Pengujian GUI dilakukan dengan memberikan beberapa nilai PWM.

Skenario ke 4 adalah dengan menekan tombol mundur pada GUI, sehingga akan mengirimkan perintah mundur kepada robot. Sama seperti pengujian sebelumnya, pada pengujian tombol mundur juga akan diberikan beberapa nilai PWM dari rendah hingga tinggi dan diamati kondisi putaran pada *tracking wheels* robot.

Skenario ke 5 adalah skenario terakhir dimana dalam skenario ini akan dilakukan pengujian tombol alloff pada GUI. dengan menekan tombol alloff maka GUI akan memberikan

perintah kepada robot untuk berhenti. Kondisi alloff ini juga berfungsi sebagai perintah *break* pada robot. Dalam gambar 4.6 dijelaskan gambaran pengujian skenario diatas:



Gambar 4.6 Skenario Pengujian

4.1.5 Hasil Skenario Pengujian

A. Skenario 1 Pengujian Tombol Maju

Tombol maju digunakan untuk memeberikan perintah kepada robot untuk dapat bergerak maju. Dalam perancangannya tombol ini memiliki *event mousePressed*. Dengan demikian dapat diartikan bahwa *event mousePressed* memberikan kondisi dimana tombol akan bekerja ketika mouse disorot pada tombol dan ditekan.

Penambahan tombol dalam GUI menggunakan library ControlP5 tombol dapat diatur sesuai dengan keinginan baik dalam segi ukuran, warna, dan font tulisan, warna tombol saat standby dapat dibedakan dengan warna tombol saat disorot maupun ditekan. Gambar 4.7 menjelaskan perbedaan warna tombol maju dalam setiap kondisi. Membedakan warna tombol dapat dilakukan dengan mengisikan nilai

R, G, B sesuai dengan keinginan. Berikut merupakan perintah yang digunakan:

```
cp5.setColorForeground(color(R, G, B));
cp5.setColorActive(color(R, G, B));
cp5.setColorBackground(color(R, G, B));
```



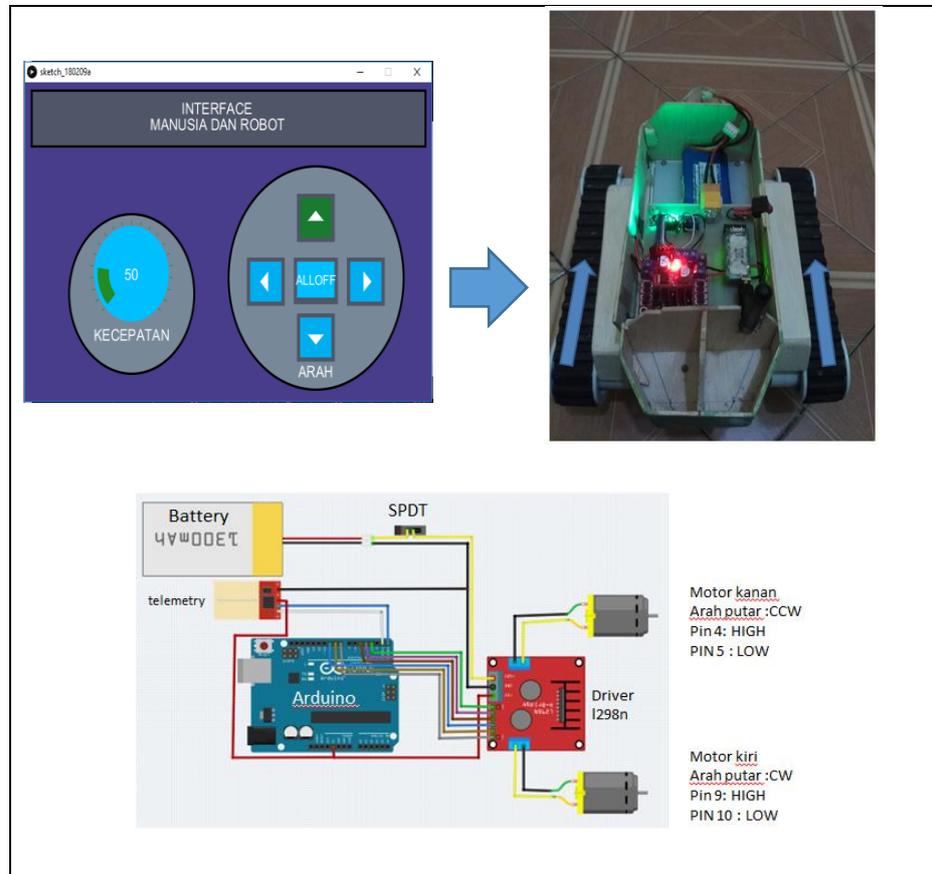
Gambar 4.7 Warna Icon Tombol

Dalam tombol maju terdapat sub rutin program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol maju ditekan terdapat rutin program untuk mengirimkan data berupa karakter “F” sekaligus mengirimkan data nilai slider pada saat itu, dengan menggunakan perintah sebagai berikut:

```
void maju() { B= "F,"+ A; port.write(B); println(B) ;}
```

Perintah di atas akan mengirimkan data karakter “F” dan juga nilai slider melalui *Serial Port* pada PC menuju ke arduino melalui media module radio telemetry 3DR. Agar dapat mengendalikan robot untuk bergerak maju, maka dalam arduino juga terdapat sub rutin yang dapat membaca data inputan “F” dan menggerakkan putaran *tracking wheels* untuk bergerak maju. Oleh sebab itu dalam arduino dimasukan program dimana saat terdeteksi “F” maka PIN 4 diberikan logika (**HIGH**) dan PIN 5 diberikan logika (**LOW**) sehingga *tracking wheels* kanan akan berputar berlawanan arah jarum jam (CCW). Selain itu pada PIN 9 diberikan logika (**HIGH**) dan PIN 10 (**LOW**) sehingga *tracking wheels* akan berputar ke searah jarum jam (CW). Dengan demikian maka robot akan bergerak maju, *tracking wheels* kanan berputar ke depan dan *tracking wheels* kiri berputar ke depan.

Gambar 4.8 menunjukkan kondisi putaran *tracking wheels* saat tombol maju pada GUI ditekan.



Gambar 4.8 Prinsip Kerja TombolMaju

Tabel 4.19 Hasil Pengujian PWM Tombol Maju

No	Nilai PWM	Kondisi Motor Kanan	Kondisi Motor Kiri
1	0-30	Berhenti	Berhenti
2	31-60	Berhenti	CW (<i>clockwise</i>)
3	61-90	Berhenti	CW (<i>clockwise</i>)
4	91-120	Berhenti	CW (<i>clockwise</i>)
5	121-150	CCW (<i>counterclockwise</i>)	CW (<i>clockwise</i>)
6	151-180	CCW (<i>counterclockwise</i>)	CW (<i>clockwise</i>)
7	181-210	CCW (<i>counterclockwise</i>)	CW (<i>clockwise</i>)
8	211-230	CCW (<i>counterclockwise</i>)	CW (<i>clockwise</i>)
9	231-255	CCW (<i>counterclockwise</i>)	CW (<i>clockwise</i>)

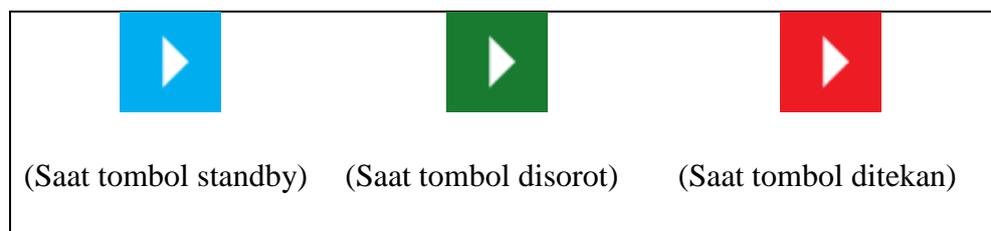
Dari data pengujian tabel 4.1 dapat diamati bahwa nilai PWM akan berpengaruh terhadap putaran motor, semakin besar nilai PWM maka motor akan berputar semakin cepat. Dari data di atas menunjukkan pada saat PWM terlalu rendah motor tidak berputar. Pada saat nilai PWM ≤ 130 maka hanya 1 motor yang akan berputar. Dari hasil pengujian kedua buah motor akan berputar bersamaan pada saat nilai PWM ≥ 130 .

B. Skenario 2 Pengujian Tombol Kanan

Tombol kanan berfungsi memberikan perintah kepada robot untuk dapat bergerak berbelok ke kanan. Dalam perancangannya tombol ini memiliki *event mousePressed*. Artinya, *event mousePressed* memberikan kondisi dimana tombol akan bekerja ketika mouse disorot pada tombol dan ditekan.

Sama halnya dengan tombol maju, pada tombol kanan pembuatan tombol dalam GUI menggunakan library ControlP5 tombol dapat diatur sesuai dengan keinginan dalam segi ukuran, warna, dan font tulisan, warna tombol saat standby dapat dibedakan dengan warna tombol saat disorot maupun ditekan. Gambar 4.9 menjelaskan perbedaan warna tombol kanan dalam setiap kondisi. Membedakan warna tombol dapat dilakukan dengan mengisi nilai R, G, B sesuai dengan keinginan. Berikut merupakan perintah yang digunakan:

```
cp5.setColorForeground(color(R, G, B));
cp5.setColorActive(color(R, G, B));
cp5.setColorBackground(color(R, G, B));
```

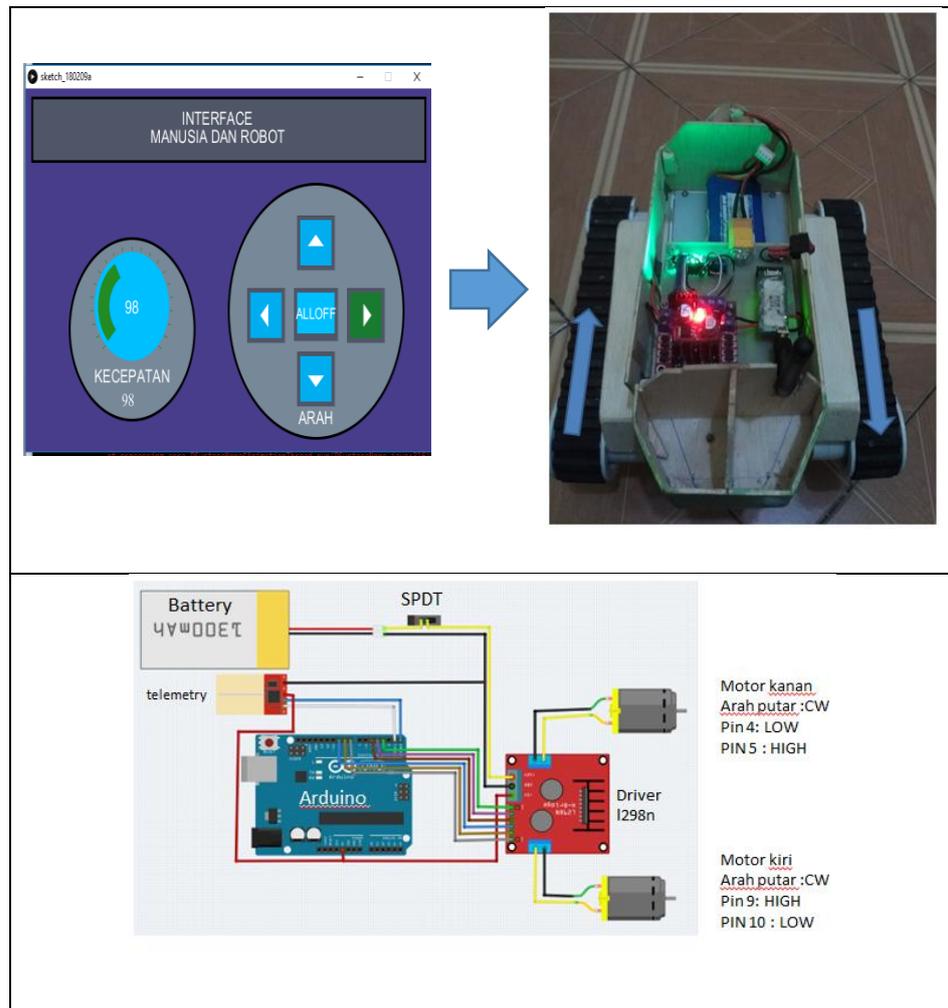


Gambar 4.9 Warna Icon Tombol Kanan

Dalam tombol kanan juga terdapat sub rutin program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol kanan ditekan terdapat rutin program untuk mengirimkan data *Right* berupa karakter “R” sekaligus mengirimkan data nilai slider pada saat itu, dengan menggunakan perintah berikut:

```
void kanan () { B= "R,"+ A; port.write(B); println(B) ;}
```

Perintah di atas akan mengirimkan data karakter “R” dan juga nilai slider melalui *Serial Port* pada PC menuju ke arduino melalui media module radio telemetry 3DR. Agar dapat mengendalikan robot untuk bergerak belok kanan, maka dalam arduino juga terdapat sub rutin yang dapat membaca data inputan “R” dan menggerakkan putaran *tracking wheels* untuk bergerak berbelok kanan . Oleh sebab itu dalam arduino dimasukan program dimana saat terdeteksi “R” maka PIN 4 diberikan logika (**LOW**) dan PIN 5 diberikan logika (**HIGH**) sehingga *tracking wheels* kanan akan berputar searah jarum jam (CW). Selain itu pada PIN 9 diberikan logika (**HIGH**) dan PIN 10 (**LOW**) sehingga *tracking wheels* kiri akan berputar searah jarum jam (CW). Dengan demikian maka robot akan bergerak berputar belok ke kanan. Dengan kondisi *tracking wheels* kanan berputar ke belakang dan *tracking wheels* kiri berputar ke depan. Gambar 4.10 menunjukkan kondisi putaran *tracking wheels* saat tombol kanan pada GUI ditekan.



Gambar 4.10 Prinsip Kerja Tombol Kanan

Tabel 4.20 Hasil Pengujian PWM Tombol Kanan

No	Nilai PWM	Kondisi Motor Kanan	Kondisi Motor Kiri
1	0-30	Berhenti	Berhenti
2	31-60	Berhenti	CW (<i>clockwise</i>)
3	61-90	Berhenti	CW (<i>clockwise</i>)
4	91-130	Behenti	CW (<i>clockwise</i>)
5	131-150	CW (<i>clockwise</i>)	CW (<i>clockwise</i>)
6	151-180	CW (<i>clockwise</i>)	CW (<i>clockwise</i>)
7	181-210	CW (<i>clockwise</i>)	CW (<i>clockwise</i>)
8	211-230	CW (<i>clockwise</i>)	CW (<i>clockwise</i>)
9	231-255	CW (<i>clockwise</i>)	CW (<i>clockwise</i>)

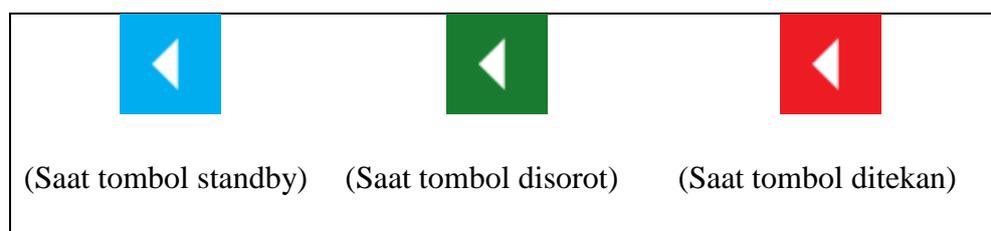
Dari data pengujian tabel 4.2 dapat diamati bahwa nilai PWM akan berpengaruh terhadap putaran motor, semakin besar nilai PWM maka motor akan berputar semakin cepat. Dari data di atas menunjukkan pada saat PWM terlalu rendah motor tidak berputar. Pada saat nilai PWM ≤ 130 maka hanya 1 motor yang akan berputar. Dari hasil pengujian kedua buah motor akan berputar bersamaan pada saat nilai PWM ≥ 130 .

C. Skenario 3 Pengujian Tombol Kiri

Tombol kiri berfungsi untuk memberikan perintah kepada robot untuk dapat bergerak berbelok ke kiri. Dalam perancangannya tombol ini memiliki *event mousePressed*. Artinya, *event mousePressed* memberikan kondisi dimana tombol akan bekerja ketika mouse disorot pada tombol dan ditekan.

Sama halnya dengan fungsi tombol sebelumnya, pada tombol kiri pembuatan tombol dalam GUI menggunakan library ControlP5 tombol dapat diatur sesuai dengan keinginan dalam ukuran, warna, dan font tulisan, warna tombol saat standby dapat dibedakan dengan warna tombol saat disorot maupun ditekan. Gambar 4.11 menjelaskan perbedaan warna tombol kiri dalam setiap kondisi. Membedakan warna tombol dapat dilakukan dengan mengisikan nilai R, G, B sesuai dengan keinginan. Berikut merupakan perintah yang digunakan:

```
cp5.setColorForeground(color(R, G, B));
cp5.setColorActive(color(R, G, B));
cp5.setColorBackground(color(R, G, B));
```

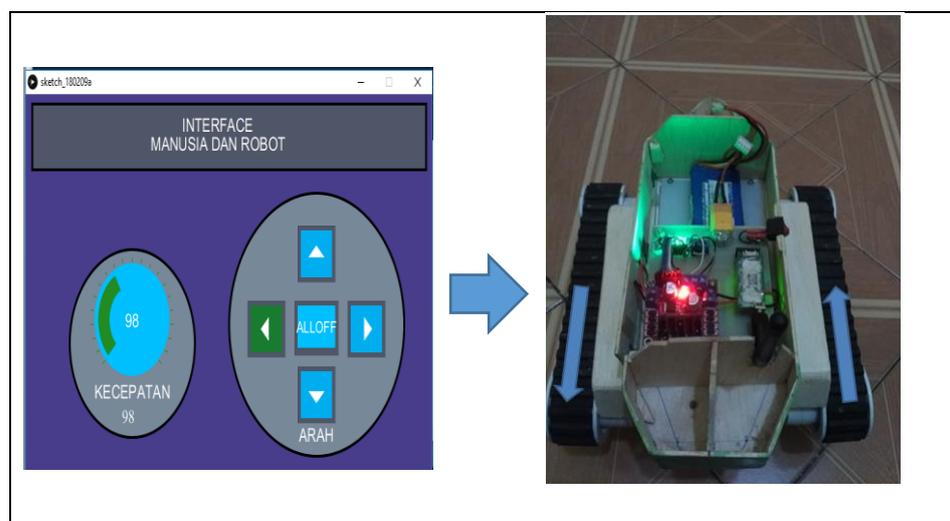


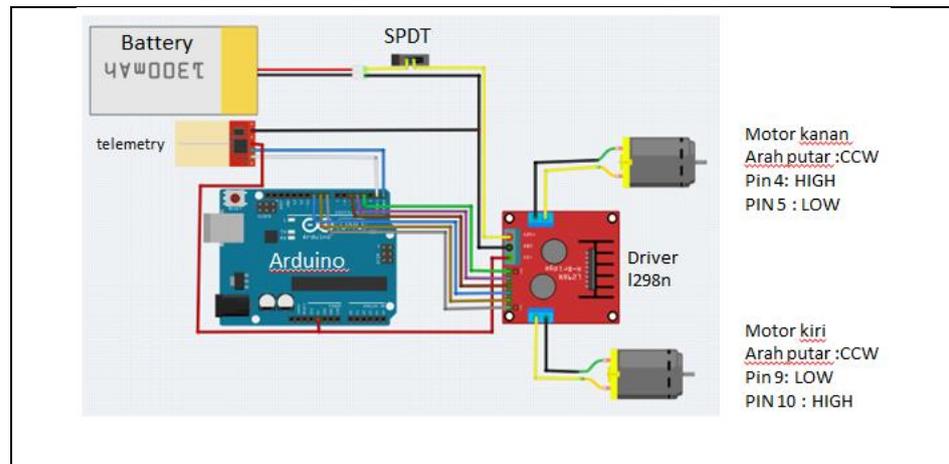
Gambar 4.11 Warna Icon Tombol Kiri

Tombol kiri juga terdapat sub rutin program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol kiri ditekan terdapat rutin program untuk mengirimkan data *Left* yang diwakili dengan karakter “L” sekaligus mengirimkan data nilai slider pada saat itu, dengan menggunakan perintah berikut:

```
void kiri () { B= "L,"+ A; port.write(B); println(B) ;}
```

Perintah diatas akan mengirimkan data karakter “L” dan juga nilai slider melalui *Serial Port* pada PC menuju ke arduino melalui media module radio telemetry 3DR. Agar dapat mengendalikan robot untuk bergerak belok kiri, maka dalam arduino juga terdapat sub rutin yang dapat membaca data inputan “L” dan menggerakkan putaran *tracking wheels* untuk bergerak berbelok kekiri . Oleh sebab itu dalam arduino dimasukan program dimana saat terdeteksi “L” maka PIN 4 diberikan logika (**HIGH**) dan PIN 5 diberikan logika (**LOW**) sehingga *tracking wheels* kanan akan berputar berlawanan arah jarum jam (CCW). Selain itu pada PIN 9 diberikan logika (**LOW**) dan PIN 10 (**HIGH**) sehingga *tracking wheels* kiri akan berputar ke berlawanan arah jarum jam (CCW). Dengan demikian maka robot akan bergerak berputar belok ke kiri. Dengan kondisi *tracking wheels* kanan berputar ke depan dan *tracking wheels* kiri berputar ke belakang. Gambar 4.12 menunjukkan kondisi putaran *tracking wheels* saat tombol kiri pada GUI ditekan.





Gambar 4.12 Prinsip Kerja Tombol kiri

Tabel 4.21 Hasil Pengujian PWM Tombol Kiri

No	Nilai PWM	Kondisi Motor Kanan	Kondisi Motor Kiri
1	0-30	Berhenti	Berhenti
2	31-60	Berhenti	CCW (<i>counterclockwise</i>)
3	61-90	Berhenti	CCW (<i>counterclockwise</i>)
4	91-120	Behenti	CCW (<i>counterclockwise</i>)
5	121-150	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
6	151-180	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
7	181-210	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
8	211-230	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
9	231-255	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)

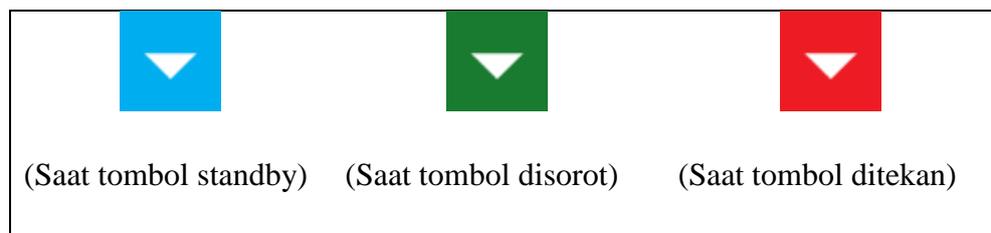
Dari data pengujian tabel 4.3 dapat diamati bahwa nilai PWM akan berpengaruh terhadap putaran motor, semakin besar nilai PWM maka motor akan berputar semakin cepat. Dari data di atas menunjukkan pada saat PWM terlalu rendah motor tidak berputar. Pada saat nilai PWM ≤ 130 maka hanya 1 motor yang akan berputar. Dari hasil pengujian kedua buah motor akan berputar bersamaan pada saat nilai PWM ≥ 130 .

D. Skenario 4 Pengujian Tombol Mundur

Tombol mundur berfungsi untuk memberikan perintah pada robot untuk dapat bergerak mundur. Dalam perancangannya tombol ini memiliki *event mousePressed*. Artinya, *event mousePressed* memberikan kondisi dimana tombol akan bekerja ketika mouse disorot pada tombol dan ditekan.

Sama halnya dengan tombol maju, pada tombol mundur pembuatan tombol dalam GUI menggunakan library ControlP5 tombol dapat diatur sesuai dengan keinginan dalam segi ukuran, warna, dan font tulisan, warna tombol saat standby dapat dibedakan dengan warna tombol saat disorot maupun ditekan. Gambar 4.13 menjelaskan perbedaan warna tombol mundur dalam setiap kondisi. Membedakan warna tombol dapat dilakukan dengan mengisikan nilai R, G, B sesuai dengan keinginan. Berikut merupakan perintah yang digunakan:

```
cp5.setColorForeground(color(R, G, B));
cp5.setColorActive(color(R, G, B));
cp5.setColorBackground(color(R, G, B));
```



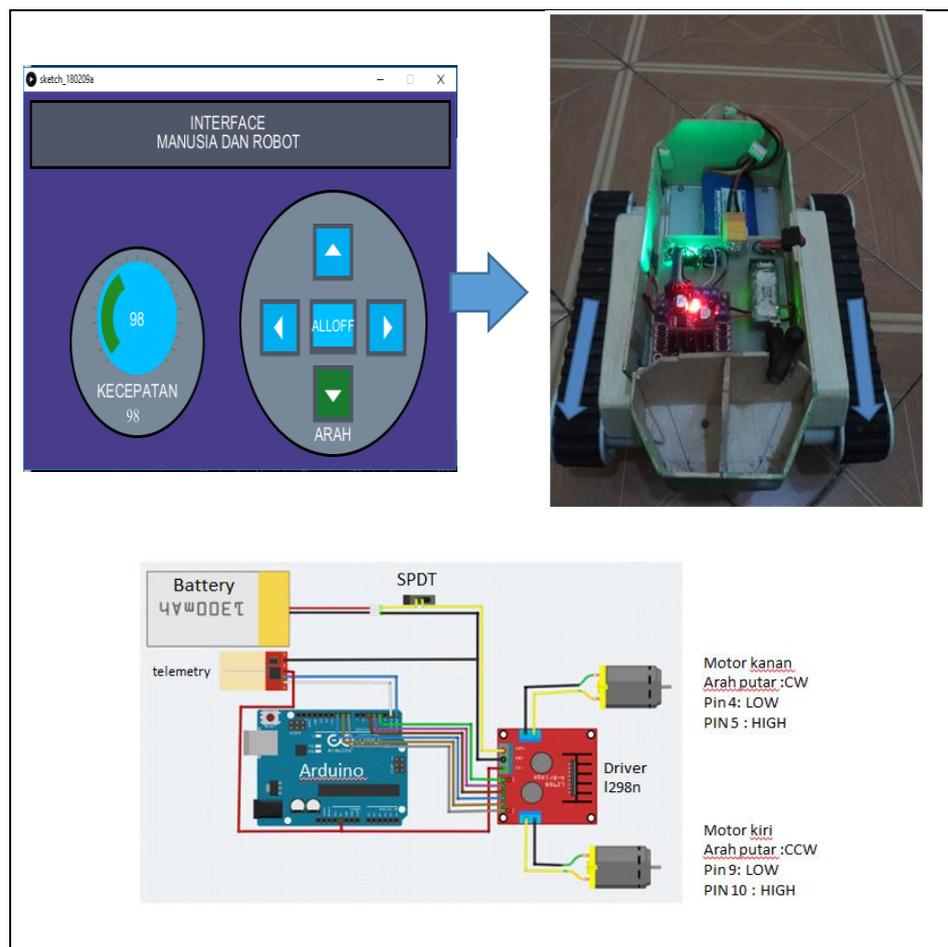
Gambar 4.13 Warna Icon Tombol Mundur

Dalam tombol mundur juga terdapat sub rutin program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol mundur ditekan terdapat rutin program untuk mengirimkan data berupa karakter “B” sekaligus mengirimkan data nilai slider pada saat itu, dengan menggunakan perintah

```
void mundur() { B= "B,"+ A; port.write(B); println(B) ;}
```

Perintah diatas akan mengirimkan data karakter “B” dan juga nilai slider melalui *Serial Port* pada PC menuju ke arduino melalui media

module radio telemetry 3DR. Agar dapat mengendalikan robot untuk bergerak mundur, maka dalam arduino juga terdapat sub rutin yang dapat membaca data inputan “B” dan menggerakkan putaran *tracking wheels* untuk bergerak mundur. Oleh sebab itu dalam arduino dimasukan program dimana saat terdeteksi “B” maka PIN 4 diberikan logika (**LOW**) dan PIN 5 diberikan logika (**HIGH**) sehingga *tracking wheels* kanan akan berputar searah jarum jam (CW). Selain itu pada PIN 9 diberikan logika (**LOW**) dan PIN 10 (**HIGH**) sehingga *tracking wheels* kiri akan berputar berlawanan arah jarum jam (CCW). Dengan demikian maka robot akan bergerak mundur *tracking wheels* kanan berputar ke belakang dan *tracking wheels* kiri berputar ke belakang. Gambar 4.14 menunjukkan kondisi putaran *tracking wheels* saat tombol mundur pada GUI ditekan.



Gambar 4.14 Prinsip Kerja Tombol Mundur

Tabel 4.22 Hasil Pengujian PWM Tombol Mundur

No	Nilai PWM	Kondisi Motor Kanan	Kondisi Motor Kiri
1	0-30	Berhenti	Berhenti
2	31-60	Berhenti	CCW (<i>counterclockwise</i>)
3	61-90	Berhenti	CCW (<i>counterclockwise</i>)
4	91-120	Behenti	CCW (<i>counterclockwise</i>)
5	121-150	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
6	151-180	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
7	181-210	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
8	211-230	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)
9	231-255	CCW (<i>counterclockwise</i>)	CCW (<i>counterclockwise</i>)

Dari data pengujian tabel 4.4 dapat diamati bahwa nilai PWM akan berpengaruh terhadap putaran motor, semakin besar nilai PWM maka motor akan berputar semakin cepat. Dari data di atas menunjukkan pada saat PWM terlalu rendah motor tidak berputar. Pada saat nilai PWM ≤ 130 maka hanya 1 motor yang akan berputar. Dari hasil pengujian kedua buah motor akan berputar bersamaan pada saat nilai PWM ≥ 130 .

E. Skenario 5 Pengujian Tombol Alloff

Tombol Alloff berfungsi untuk memberikan perintah kepada robot untuk dapat berhenti bergerak. Dalam perancangannya tombol ini memiliki *event mousePressed*. Artinya, *event mousePressed* memberikan kondisi dimana tombol akan bekerja ketika mouse disorot pada tombol dan ditekan.

Sama halnya dengan fungsi tombol sebelumnya, pada tombol Alloff pembuatan tombol dalam GUI menggunakan library ControlP5 tombol dapat diatur sesuai dengan keinginan dalam ukuran, warna, dan font tulisan, warna tombol saat standby dapat dibedakan dengan warna tombol saat disorot maupun ditekan. Gambar 4.15 menjelaskan

perbedaan warna tombol alloff dalam setiap kondisi. Membedakan warna tombol dapat dilakukan dengan mengisikan nilai R, G, B sesuai dengan keinginan. Berikut merupakan perintah yang digunakan:

```
cp5.setColorForeground(color(R, G, B));
cp5.setColorActive(color(R, G, B));
cp5.setColorBackground(color(R, G, B));
```



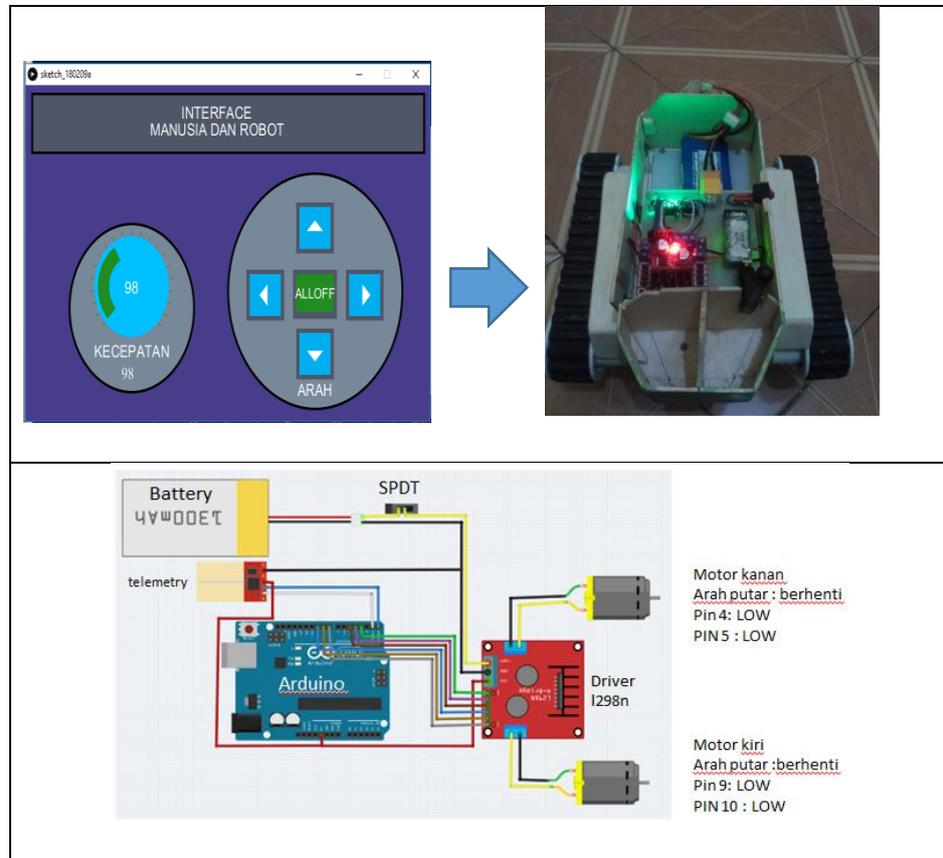
Gambar 4.15 Warna Icon Tombol Alloff

Tombol Alloff juga terdapat sub rutin program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol kiri ditekan terdapat rutin program untuk mengirimkan data *Stop* yang diwakili dengan karakter “S” sekaligus mengirimkan data nilai slider pada saat itu, dengan menggunakan perintah berikut:

```
void alloff () { B= "S,0" ; port.write(B); println(B) ;}
```

Perintah di atas akan mengirimkan data karakter “S” dan juga nilai slider melalui *Serial Port* pada PC menuju ke arduino melalui media module radio telemetry 3DR. Walaupun dalam GUI terdapat nilai PWM yang dikirim namun saat ditekan tombol Alloff maka nilai PWM akan tetap dikirimkan 0. Agar dapat mengendalikan robot untuk berhenti bergerak, maka dalam arduino juga terdapat sub rutin yang dapat membaca data inputan “S” dan menggerakkan putaran *tracking wheels* untuk berhenti bergerak. Oleh sebab itu dalam arduino dimasukan program dimana saat terdeteksi “S” maka PIN 4 diberikan logika (**LOW**) dan PIN 5 diberikan logika (**LOW**) sehingga *tracking wheels* kanan akan berhenti berputar. Selain itu pada PIN 9 diberikan logika (**LOW**) dan PIN 10 (**LOW**) sehingga *tracking wheels* kiri akan berhenti berputar. Dengan demikian maka robot akan berhenti. Gambar

4.16 menunjukkan kondisi putaran *tracking wheels* saat tombol allof pada GUI ditekan.



Gambar 4.16 Prinsip Kerja Tombol Stop

Tabel 4.23 Hasil Pengujian Tombol Alloff

No	Nilai PWM	Kondisi Motor Kanan	Kondisi Motor Kiri
1	0-30	Berhenti/break	Berhenti/break
2	31-60	Berhenti/break	Berhenti/break
3	61-90	Berhenti/break	Berhenti/break
4	91-120	Berhenti/break	Berhenti/break
5	121-150	Berhenti/break	Berhenti/break
6	151-180	Berhenti/break	Berhenti/break
7	181-210	Berhenti/break	Berhenti/break
8	211-230	Berhenti/break	Berhenti/break
9	231-255	Berhenti/break	Berhenti/break

Dari data pengujian tabel 4.5 dapat diamati bahwa pada saat tombol alloff ditekan maka GUI akan mengirimkan data berupa karakter "S" dan nilai PWM 0, sehingga ketika slider pada GUI diatur pada rentang nilai tertentu, namun nilai PWM yang diterima pada Arduino akan tetap 0.