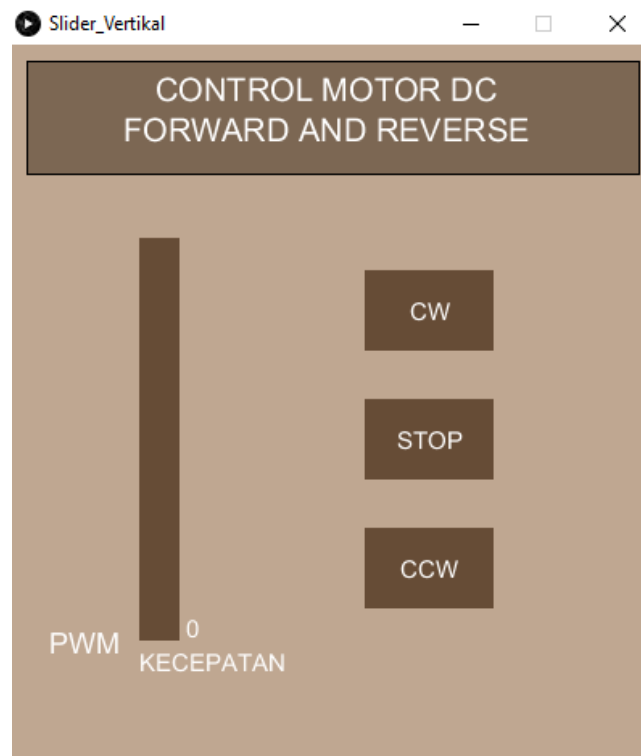


BAB IV HASIL DAN PEMBAHASAN

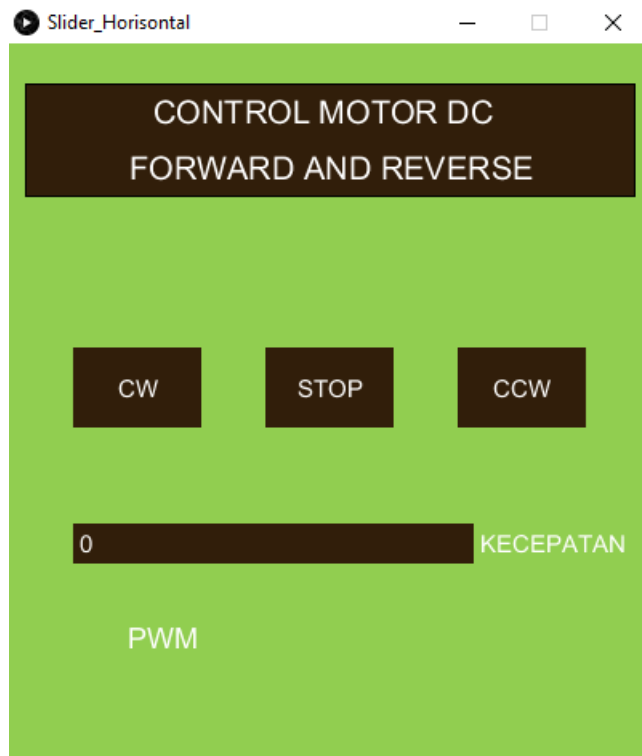
4.1 Hasil Perancangan

4.1.1 Tampilan Program GUI

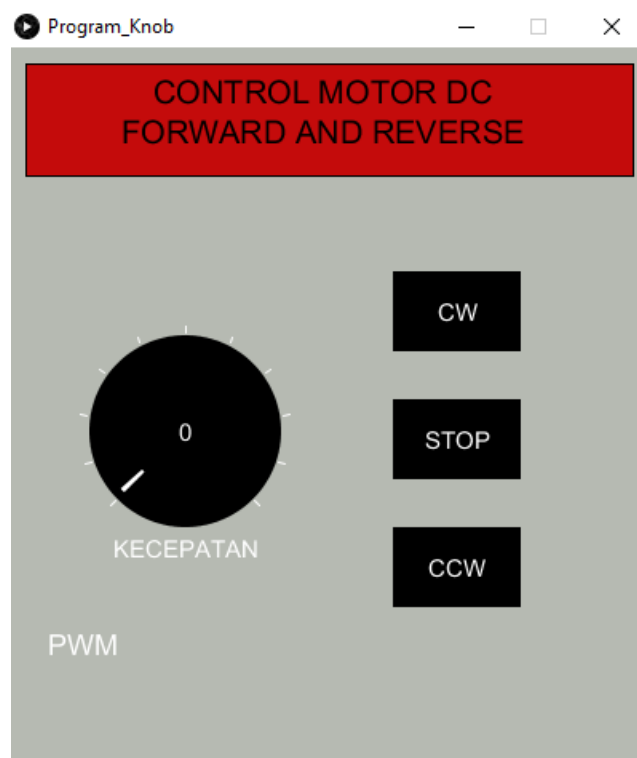
Berdasarkan pada metode perancangan Bab III, perancangan ini membentuk 3 (tiga) versi desain tampilan program *Graphical User Interface* (GUI). Pada ketiga versi desain ini telah dilakukan pengujian dengan cara membagikan kuesioner ke beberapa responden yaitu mahasiswa yang berkuliah di Fakultas Teknik Universitas Muhammadiyah Yogyakarta, dimana memang sasaran utama pengguna (*user*) program GUI ini adalah mereka yang berkuliah di jurusan keteknikan, khususnya Teknik Elektro. Desain tampilan program *Graphical User Interface* (GUI) ini akan ditunjukkan seperti pada gambar 4.1, gambar 4.2, dan gambar 4.3.



Gambar 4.1 GUI Versi 1



Gambar 4.2 GUI Versi 2



Gambar 4.3 GUI Versi 3

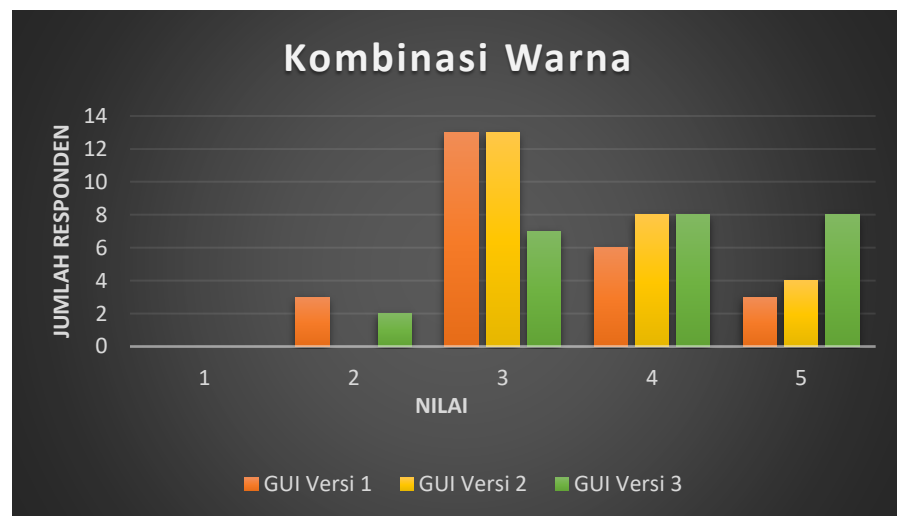
Pada perancangan ketiga desain program *Graphical User Interface* (GUI) di atas (gambar 4.1, gambar 4.2, gambar 4.3) dirancang dengan mempertimbangkan metode kombinasi warna terbaik dan kombinasi warna terburuk. Pemilihan kombinasi warna pada desain program GUI ini dipilih agar terlihat lebih menarik dan juga pengguna (*user*) akan merasa nyaman, mata tidak merasa cepat kelelahan saat menggunakan program ini.

4.1.2 Hasil Kuesioner

Dari hasil pengujian dengan cara membagikan kuesioner kepada beberapa responden, dapat dilihat hasilnya dalam *chart* berikut:

1. Penilaian untuk Pemilihan Kombinasi Warna

Pada grafik dibawah ini merupakan hasil penilaian responden mengenai pemilihan kombinasi warna pada program GUI Versi 1, Versi 2, dan Versi 3



Gambar 4.4 Hasil Penilaian Pemilihan Kombinasi Warna

Penilaian untuk pemilihan kombinasi warna pada program GUI, dapat dilihat pada diagram bahwa:

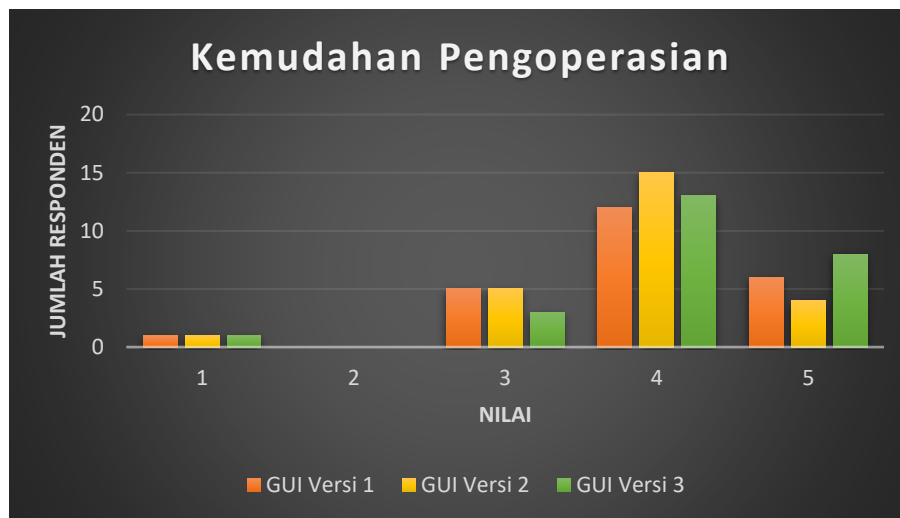
- Pada GUI versi 1, sebanyak 3 (tiga) responden memberikan nilai 2 (Buruk), 13 (tiga belas) responden memberikan nilai 3 (Cukup), 6 (enam) responden memberikan nilai 4 (Baik), dan 3 (tiga) responden memberikan nilai 5 (Sangat Baik). Pada lembar kuesioner, beberapa responden mengatakan bahwa pada GUI

versi satu, kombinasi warna yang dipilih terkesan sangat monoton dan membosankan

- b. Pada GUI versi 2, sebanyak 13 (tiga belas) responden memberikan nilai 3 (Cukup), 8 (delapan) responden memberikan nilai 4 (Baik), dan 4 (empat) responden memberikan nilai 5 (Sangat Baik). Pada lembar kuesioner, responden mengatakan bahwa pemilihan warna pada program GUI versi 2 sudah baik, namun untuk pemilihan warna backgroundnya terlalu terang (*eye catching*), sehingga menyebabkan mata tegang saat melihat tampilannya.
- c. Pada GUI versi 3, sebanyak 2 (dua) responden memberikan nilai 2 (Buruk), 7 (tujuh) responden memberikan nilai 3 (Cukup), 8 (delapan) responden memberikan nilai 4 (Baik), dan 8 (delapan) responden nilai 5 (Sangat Baik). Pada lembar kuesioner, beberapa responden mengatakan bahwa kombinasi warna pada program GUI versi 3 sudah sangat baik.
- d. Berdasarkan diagram diatas, sebanyak 12 (dua belas) responden memberikan nilai 3 (Cukup) pada program GUI versi 1 dan 2, sedangkan untuk program GUI Versi 3 responden banyak memberikan nilai 4 (Baik) dan nilai 5 (Sangat Baik) yaitu sebanyak 8 (delapan) responden pada masing masing nilai. Dilihat dari penilaian responden maka didapat disimpulkan bahwa mayoritas responden lebih menyukai kombinasi warna pada program GUI versi 3. Hanya saja untuk warna tombol CW, CCW, dan stop lebih baik dibedakan.

2. Penilaian untuk Kemudahan Pengoperasian

Pada grafik dibawah ini merupakan hasil penilaian responden mengenai kemudahan dalam pengoperasian pada program GUI Versi 1, Versi 2, dan Versi



Gambar 4.5 Hasil Penilaian Kemudahan Pengoperasian GUI

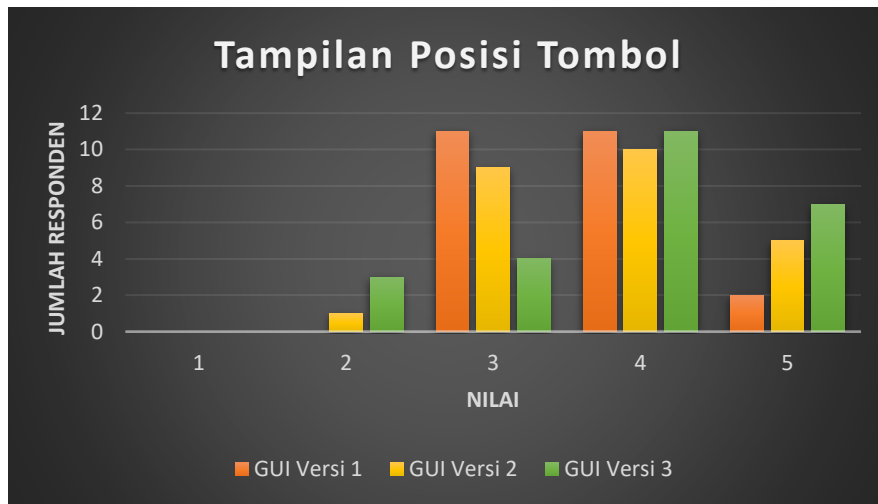
Penilaian untuk kemudahan Pengoperasian pada program GUI, dapat dilihat pada diagram bahwa:

- Pada GUI versi 1, sebanyak 1 (satu) responden memberikan nilai 1 (Sangat Buruk), 5 (lima) responden memberikan nilai 3 (Cukup), 13 (tiga belas) responden memberikan nilai 4 (Baik), dan 6 (tiga) responden memberikan nilai 5 (Sangat Baik).
- Pada GUI versi 2, sebanyak 1 (satu) responden memberikan nilai 1 (Sangat Buruk), 5 (lima) responden memberikan nilai 3 (Cukup), 15 (lima belas) responden memberikan nilai 4 (Baik), dan 4 (empat) responden memberikan nilai 5 (Sangat Baik).
- Pada GUI versi 3, sebanyak 1 (satu) responden memberikan nilai 1 (Sangat Buruk), 3 (tiga) responden memberikan nilai 3 (Cukup), 13 (tiga belas) responden memberikan nilai 4 (Baik), dan 8 (delapan) responden nilai 5 (Sangat Baik).
- Berdasarkan diagram diatas, banyak responden memberikan nilai 4 (baik) dan nilai 5 (Sangat Baik) pada program GUI versi 3. Pada lembar kuesioner, responden mengatakan bahwa pengoperasian GUI sudah baik, dan mudah, dan kebanyakan responden lebih menyukai pemilihan indicator kecepatan dalam bentuk knob, karena terlihat lebih familiar seperti speedometer pada kendaraan serta mudah dalam pengaturannya, lalu untuk tombol CW, CCW dan STOP lebih baik menggunakan simbol saja daripada tulisan (cw, ccw, stop), agar

pengoperasiannya lebih mudah. Maka didapat disimpulkan bahwa mayoritas responden merasa program GUI versi 3 lebih mudah dalam pengoperasiannya.

3. Penilaian untuk Tampilan Posisi Tombol

Pada grafik dibawah ini merupakan hasil penilaian responden mengenai penempatan posisi tombol pada program GUI Versi 1, Versi 2, dan Versi 3



Gambar 4.6 Hasil Penilaian Tampilan Posisi Tombol

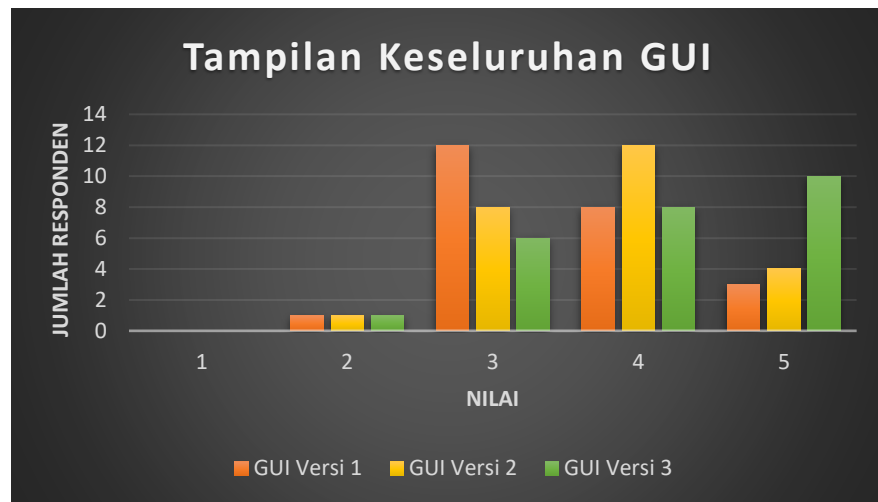
Penilaian untuk Tampilan Posisi Tombol pada program GUI, dapat dilihat pada diagram bahwa:

- Pada GUI versi 1, sebanyak 12 (dua belas) responden memberikan nilai 3 (Cukup), 11 (sebelas) responden memberikan nilai 4 (Baik), dan 2 (dua) responden memberikan nilai 5 (Sangat Baik).
- Pada GUI versi 2, sebanyak 1 (satu) responden memberikan nilai 2 (Buruk), 9 (sembilan) responden memberikan nilai 3 (Cukup), 10 (sepuluh) responden memberikan nilai 4 (Baik), dan 5 (lima) responden memberikan nilai 5 (Sangat Baik).
- Pada GUI versi 3, sebanyak 3 (tiga) responden memberikan nilai 2 (Buruk), 4 (empat) responden memberikan nilai 3 (Cukup), 11 (sebelas) responden memberikan nilai 4 (Baik), dan 7 (tujuh) responden nilai 5 (Sangat Baik).
- Berdasarkan diagram diatas, banyak responden memberikan nilai yang kurang baik dalam penempatan posisi tombol pada program GUI versi satu, (dua belas) responden memberikan nilai 3 (Cukup) pada program GUI versi 1 dan 2, sedangkan untuk program GUI Versi 3 responden banyak memberikan nilai 4

(Baik) dan nilai 5 (Sangat Baik) yaitu sebanyak 8 (delapan) responden pada masing masing nilai. Pada lembar kuesioner, responden menyebutkan bahwa penempatan tombol masih kurang rapi, terlebih pada GUI versi 2. Responden menjelaskan bahwa untuk indicator kecepatan lebih baik di letakkan di atas tombol CW, CCW, dan STOP. Maka didapat disimpulkan bahwa mayoritas responden menilai GUI versi 3.

4. Penilaian GUI secara Keseluruhan

Pada grafik dibawah ini merupakan hasil penilaian responden mengenai tampilan keseluruhan pada program GUI Versi 1, Versi 2, dan Versi



Gambar 4.7 Hasil Penilaian Tampilan Keseluruhan GUI

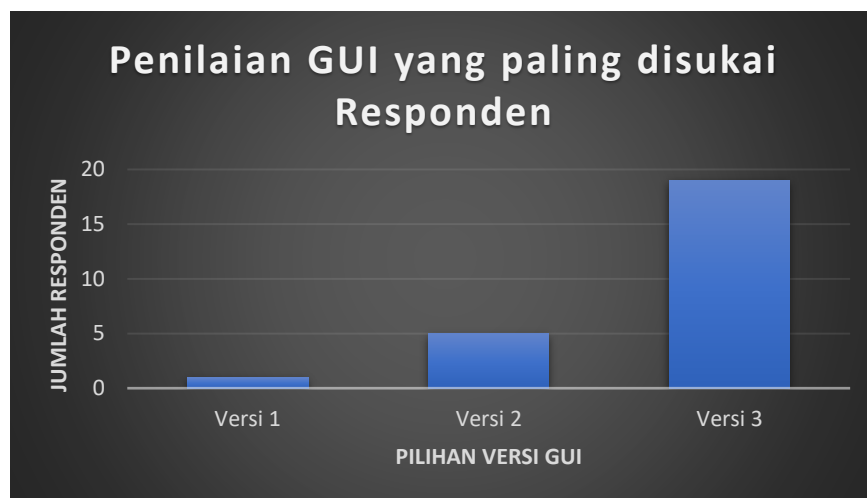
Penilaian untuk Tampilan keseluruhan pada program GUI, dapat dilihat pada diagram bahwa:

- Pada GUI versi 1, sebanyak 1 (satu) responden memberikan nilai 2 (Buruk), 12 (dua belas) responden memberikan nilai 3 (Cukup), 9 (sembilan) responden memberikan nilai 4 (Baik), dan 3 (tiga) responden memberikan nilai 5 (Sangat Baik).
- Pada GUI versi 2, sebanyak 1 (satu) responden memberikan nilai 2 (Buruk), 8 (delapan) responden memberikan nilai 3 (Cukup), 12 (dua belas) responden memberikan nilai 4 (Baik), dan 4 (empat) responden memberikan nilai 5 (Sangat Baik).

- c. Pada GUI versi 3, sebanyak 1 (satu) responden memeberikan nilai 2 (Buruk), 6 (enam) responden memberikan nilai 3 (Cukup), 8 (delapan) responden memberikan nilai 4 (Baik), dan 10 (sepuluh) responden nilai 5 (Sangat Baik).

5. Penilaian GUI yang paling disukai Responden

Pada grafik dibawah ini merupakan hasil penilaian GUI yang paling disukai oleh responden



Gambar 4.8 Hasil Penilaian GUI yang paling disukai

Penilaian untuk program GUI yang paling disukai oleh responden, dapat dilihat pada diagram bahwa:

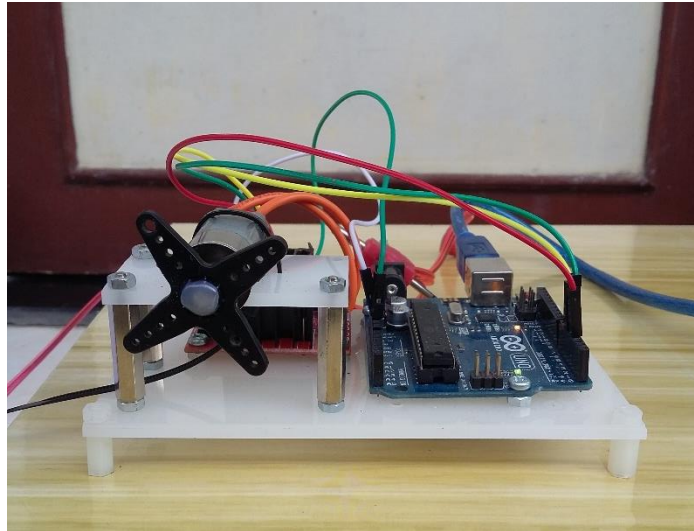
- Sebanyak 1 (satu) responden memilih program GUI versi 1.
- Sebanyak 5 (lima) responden memilih program GUI versi 2.
- Sebanyak 19 (Sembilan belas) responden memilih program GUI versi 3.

4.1.3 Tampilan Rangkaian Motor DC

Rangkaian Motor DC ini dirancang dengan beberapa komponen yaitu:

- 1) Arduino Uno ATmega328
- 2) Driver Motor L298
- 3) 1 buah Motor DC 16mm 1000 RPM
- 4) Adaptor 12V
- 5) Kabel Jumper

- 6) Kabel Ties
- 7) Double Foam
- 8) Acrylic
- 9) Mur & Baut
- 10) Specer



Gambar 4.9 Rangkaian Motor DC

4.2 Pembahasan

4.2.1 Pengujian Tombol

1. Skrip 1 (Pengujian Tombol CW (*clockwise*))

Pada perancangan program *Graphical User Interface* ini terdapat tombol CW (*clockwise*), yaitu merupakan tombol yang digunakan untuk memberi perintah pada Motor DC untuk berputar searah jarum jam. Tombol ini akan mulai bekerja setelah *user* (pengguna) mengatur kecepatan putaran Motor DC pada slider ataupun knob, lalu setelah mengatur kecepatan, *user* akan memilih dan menekan tombol CW tersebut. Pengujian akan dilakukan dengan cara mengatur kecepatan putaran mulai dari minimum kecepatan hingga maksimum kecepatan (Range 0-255).

2. Skrip 2 (Pengujian Tombol CCW (*Counter Clockwise*))

Pada perancangan program *Graphical User Interface* (GUI) ini terdapat pula tombol CCW (*Counter Clockwise*), yaitu merupakan tombol yang digunakan untuk memberi perintah pada Motor DC untuk berputar berlawanan arah jarum jam. Tombol ini akan mulai bekerja setelah *user* (pengguna) mengatur kecepatan putaran

Motor DC pada slider ataupun knob, lalu setelah mengatur kecepatan *user* akan memilih dan menekan tombol CCW tersebut. Pengujian akan dilakukan dengan cara mengatur kecepatan putaran mulai dari minimum kecepatan hingga maksimum kecepatan (Range 0-255).

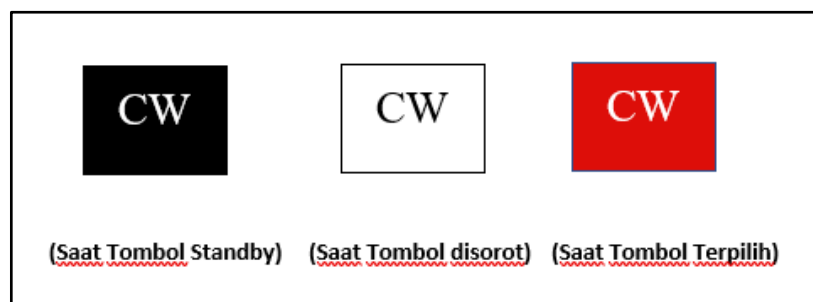
3. Skrip 3 (Pengujian Tombol STOP)

Pada perancangan program *Graphical User Interface* (GUI) ini terdapat tombol STOP (berhenti), yaitu merupakan tombol yang digunakan untuk memberi perintah pada motor DC untuk berhenti berputar. Tombol ini akan bekerja setelah *user* (pengguna) memilih dan menekan tombol STOP tersebut.

4.2.2 Hasil Pengujian Skrip Tombol

1. Skrip 1 (Pengujian Tombol CW (*clockwise*))

Pada tombol CW (*clockwise*) digunakan untuk memberikan perintah agar Motor DC berputar searah jarum jam. Dalam perancangan pembuatan tombol ini *library* yang digunakan yaitu ControlP5. *Library* ini sangat memudahkan untuk pembuatan tombol, baik dari segi bentuk, ukuran maupun pemilihan warnanya. Gambar 4.10 akan menjelaskan warna pada tombol CW dalam berbagai kondisi, yaitu saat keadaan *standby*, saat disorot, dan saat terpilih.



Gambar 4.10 Warna Tombol CW

Pengaturan warna pada tombol seperti ditunjukkan pada gambar 4.10, dapat diatur pada perintah berikut ini dengan memasukan nilai R, G, B sesuai dengan keinginan:

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

Digunakan untuk mengatur warna tombol dalam keadaan disorot.

b. cp5.setColorActive(color(R, G, B));

Digunakan untuk mengatur warna tombol dalam keadaan terpilih.

c. cp5.setColorBackground(color(R, G, B));

Digunakan untuk mengatur warna tombol dalam keadaan tidak terpilih atau *standby*.

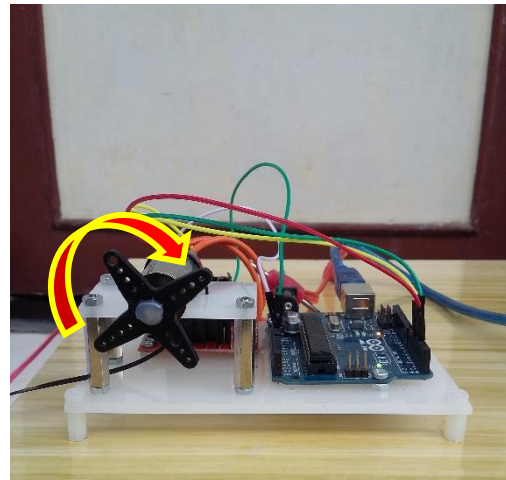
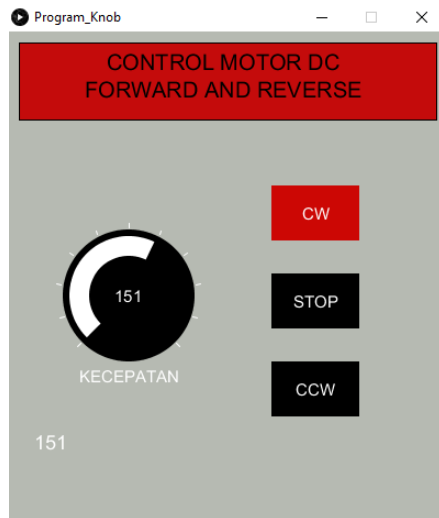
d. cp5.setColorCaptionLabel(color(R, G, B));

Digunakan untuk mengatur warna label atau tulisan.

Pada tombol CW terdapat instruksi perintah program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol CW ditekan terdapat instruksi perintah program untuk mengirimkan data “Kanan” berupa karakter B sekaligus mengirimkan data nilai knob pada saat itu menggunakan perintah:

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

Perintah tersebut diatas mengirimkan data karakter “B” dan juga nilai knob melalui *Serial Port* pada PC menuju ke Arduino. Pada saat mengendalikan Motor DC agar berputar searah jarum jam, maka terdapat instruksi perintah juga pada Arduino yang dapat membaca data inputan “B” dan menggerakkan putaran searah jarum jam. Maka dari itu pada Arduino dimasukan program dimana saat terdeteksi “B” maka PIN 4 diberi logika (LOW) dan PIN 5 diberi logika (HIGH) sehingga arah putarannya akan searah jarum jam atau clockwise (CW). Pada gambar 4.11 akan menunjukkan kondisi putaran saat tombol CW (clockwise) pada GUI ditekan.



Gambar 4.11 Kondisi Putaran saat tombol CW ditekan

Pada tabel 4.1 merupakan hasil dari pengujian tombol CW pada program GUI ke rangkaian Motor DC

Tabel 4.1 Pengujian Tombol CW (clockwise)

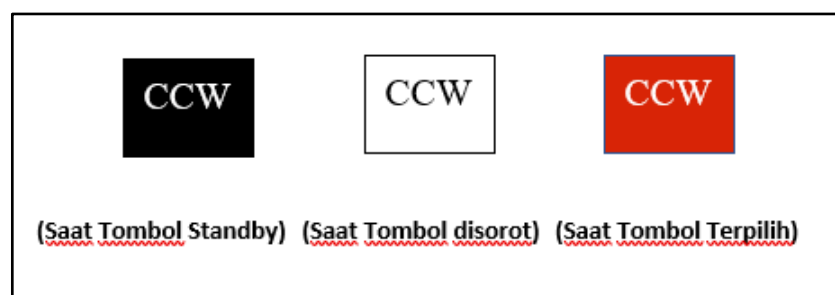
No	Nilai PWM	Kondisi Motor DC	Keterangan	Duty Cycle
1	0-10	Berhenti	Tidak Berputar	0-3.92%
2	11-24	Berhenti	Tidak Berputar	4.31% - 9.41%
3	25-30	CW (<i>clockwise</i>)	Berputar Sangat Pelan	9.80% - 11.76%
4	31-40	CW (<i>clockwise</i>)	Berputar Pelan	12.15% - 15.68%
5	41-50	CW (<i>clockwise</i>)	Berputar Cukup Cepat	10.07% - 19.6%
6	51-60	CW (<i>clockwise</i>)	Berputar Cepat	20% - 23.52%
7	61-90	CW (<i>clockwise</i>)	Berputar Cepat	23.92% - 35.29%
8	91-120	CW (<i>clockwise</i>)	Berputar Lebih Cepat	35.68% - 47.05%
9	121-150	CW (<i>clockwise</i>)	Berputar Lebih Cepat	47.45% - 58.82%

10	151-180	CW (<i>clockwise</i>)	Berputar Sangat Cepat	59.21% - 70.58%
11	181-255	CW (<i>clockwise</i>)	Berputar Sangat Cepat	70.98% - 100%

Dari hasil data pengujian pada table 4.1 dapat dilihat bahwa nilai PWM sangat berpengaruh terhadap putaran motor, ketika nilai PWM kecil maka motor akan berputar lambat bahkan jika dilihat dari table 4.1, saat nilai PWM 0-24 motor dalam kondisi berhenti atau tidak berputar, begitu juga dengan nilai PWM yang semakin besar maka putaran yang ada pada motor akan semakin cepat. Hal ini dapat disimpulkan bahwa nilai PWM berbanding lurus dengan kecepatan putaran pada motor.

2. Skrip 2 (Pengujian Tombol CCW (*counter clockwise*))

Pada tombol CCW (*counter clockwise*) digunakan untuk memberikan perintah agar motor DC berputar berlawanan arah jarum jam. Dalam perancangan pembuatan tombol ini *library* yang digunakan yaitu ControlP5. *Library* ini sangat memudahkan untuk pembuatan tombol, baik dari segi bentuk, ukuran maupun pemilihan warnanya. Gambar 4.12 akan menjelaskan warna pada tombol CCW dalam berbagai kondisi, yaitu saat keadaan *standby*, saat disorot, dan saat terpilih.



Gambar 4.12 Warna Tombol CCW

Pengaturan warna pada tombol seperti ditunjukkan pada gambar 4.12, dapat diatur pada perintah berikut ini dengan memasukan nilai R, G, B sesuai dengan keinginan:

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

Digunakan untuk mengatur warna tombol dalam keadaan disorot.

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

Digunakan untuk mengatur warna tombol dalam keadaan terpilih.

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

Digunakan untuk mengatur warna tombol dalam keadaan tidak terpilih atau *standby*.

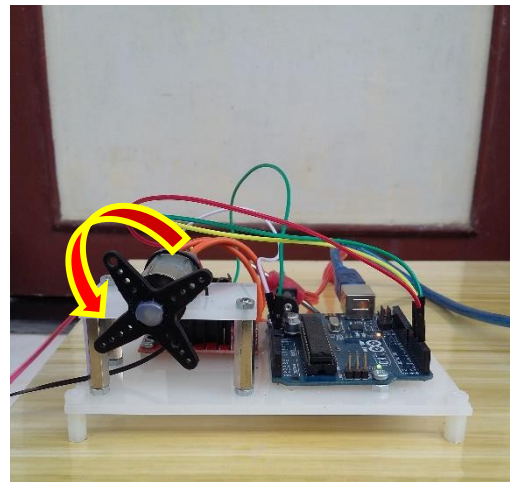
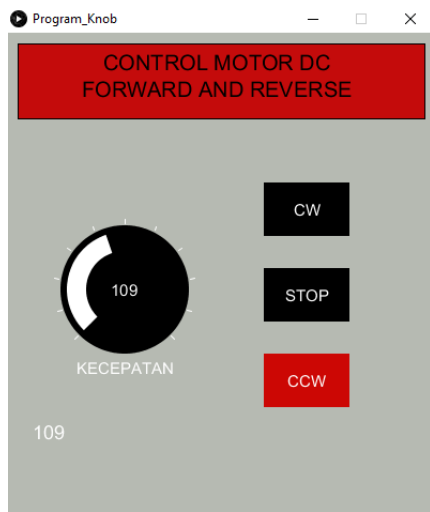
d. `cp5.setColorCaptionLabel(color(R, G, B));`

Digunakan untuk mengatur warna label atau tulisan.

Pada tombol CCW terdapat instruksi perintah program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol CCW ditekan terdapat instruksi perintah program untuk mengirimkan data “Kiri” berupa karakter F sekaligus mengirimkan data nilai knob pada saat itu menggunakan perintah:

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

Perintah tersebut diatas mengirimkan data karakter “F” dan juga nilai knob melalui *Serial Port* pada PC menuju ke Arduino. Pada saat mengendalikan Motor DC agar berputar berlawanan arah jarum jam, maka terdapat instruksi perintah juga pada Arduino yang dapat membaca data inputan “F” dan menggerakkan putaran berlawanan arah jarum jam. Maka dari itu pada Arduino dimasukan program dimana saat terdeteksi “F” maka PIN 4 diberi logika (**HIGH**) dan PIN 5 diberi logika (**LOW**) sehingga arah putarannya akan searah jarum jam atau *counter clockwise* (**CCW**). Pada gambar 4.13 akan menunjukkan kondisi putaran saat tombol CCW (*counter clockwise*) pada GUI ditekan.



Gambar 4.13 Kondisi Putaran saat tombol CCW ditekan

Pada tabel 4.2 merupakan hasil dari pengujian tombol CCW pada program GUI ke rangkaian Motor DC

Tabel 4.2 Pengujian Tombol CCW (*counter clockwise*)

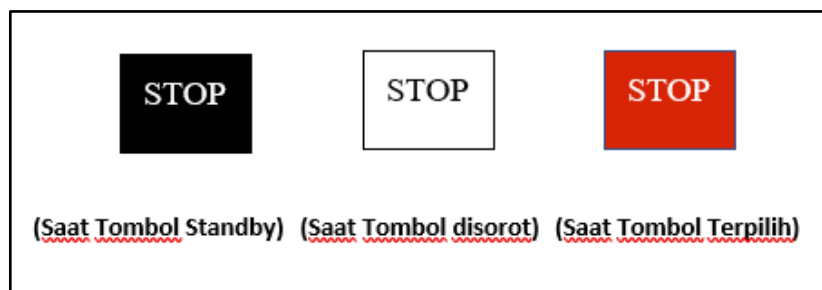
No	Nilai PWM	Kondisi Motor DC	Keterangan	Duty Cycle
1	0-10	Berhenti	Tidak Berputar	0-3.92%
2	11-24	Berhenti	Tidak Berputar	4.31% - 9.41%
3	25-30	CCW (<i>counter clockwise</i>)	Berputar Sangat Pelan	9.80% - 11.76%
4	31-40	CCW (<i>counter clockwise</i>)	Berputar Pelan	12.15% - 15.68%
5	41-50	CCW (<i>counter clockwise</i>)	Berputar Cukup Cepat	10.07% - 19.6%
6	51-60	CCW (<i>counter clockwise</i>)	Berputar Cepat	20% - 23.52%
7	61-90	CCW (<i>counter clockwise</i>)	Berputar Cepat	23.92% - 35.29%

8	91-120	CCW (<i>counter clockwise</i>)	Berputar Lebih Cepat	35.68% - 47.05%
9	121-150	CCW (<i>counter clockwise</i>)	Berputar Lebih Cepat	47.45% - 58.82%
10	151-180	CCW (<i>counter clockwise</i>)	Berputar Sangat Cepat	59.21% - 70.58%
11	181-255	CCW (<i>counter clockwise</i>)	Berputar Sangat Cepat	70.98% - 100%

Dari hasil data pengujian pada table 4.1 dapat dilihat bahwa nilai PWM sangat berpengaruh terhadap putaran motor, ketika nilai PWM kecil maka motor akan berputar lambat bahkan jika dilihat dari table 4.2, saat nilai PWM 0-24 motor dalam kondisi berhenti atau tidak berputar, begitu juga dengan nilai PWM yang semakin besar maka putaran yang ada pada motor akan semakin cepat. Hal ini dapat disimpulkan bahwa nilai PWM berbanding lurus dengan kecepatan putaran pada motor.

3. Skrip 3 (Pengujian Tombol STOP (berhenti/tidak berputar))

Pada tombol STOP (berhenti) digunakan untuk memberikan perintah agar motor DC berhenti berputar. Dalam perancangan pembuatan tombol ini *library* yang digunakan yaitu ControlP5. *Library* ini sangat memudahkan untuk pembuatan tombol, baik dari segi bentuk, ukuran maupun pemilihan warnanya. Gambar 4.14 akan menjelaskan warna pada tombol STOP dalam berbagai kondisi, yaitu saat keadaan *standby*, saat disorot, dan saat terpilih.



Gambar 4.14 Warna Tombol STOP

Pengaturan warna pada tombol seperti ditunjukkan pada gambar 4.14, dapat diatur pada perintah berikut ini dengan memasukan nilai R, G, B sesuai dengan keinginan:

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

Digunakan untuk mengatur warna tombol dalam keadaan disorot.

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

Digunakan untuk mengatur warna tombol dalam keadaan terpilih.

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

Digunakan untuk mengatur warna tombol dalam keadaan tidak terpilih atau *standby*.

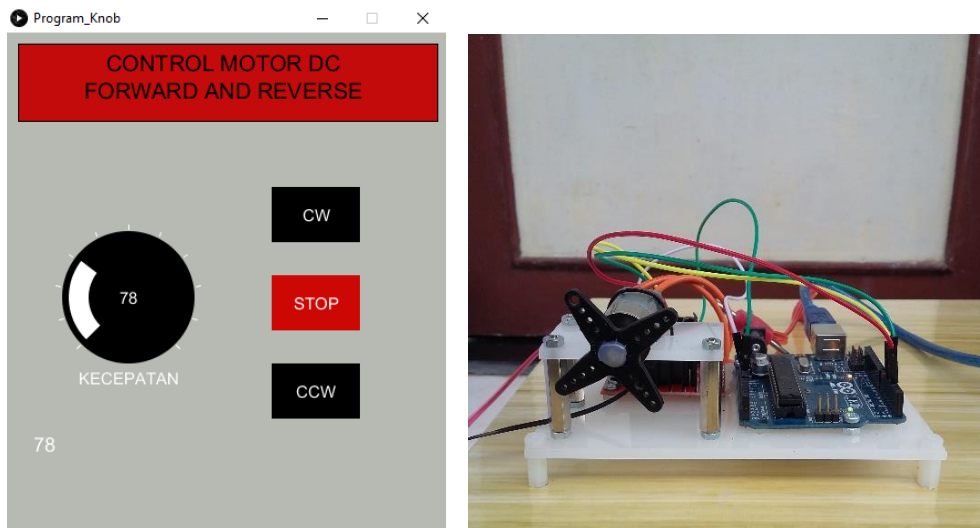
d. `cp5.setColorCaptionLabel(color(R, G, B));`

Digunakan untuk mengatur warna label atau tulisan.

Pada tombol STOP terdapat instruksi perintah program untuk mengirimkan data berupa karakter melalui serial komunikasi. Pada saat tombol STOP ditekan terdapat instruksi perintah program untuk mengirimkan data “Stop” berupa karakter S sekaligus mengirimkan data nilai knob pada saat itu menggunakan perintah:

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

Perintah tersebut diatas mengirimkan data karakter “S” dan juga nilai knob melalui *Serial Port* pada PC menuju ke Arduino. Pada saat mengendalikan Motor DC agar berhenti berputar, maka terdapat instruksi perintah juga pada Arduino yang dapat membaca data inputan “S”. Maka dari itu pada Arduino dimasukan program dimana saat terdeteksi “S” maka PIN 4 diberi logika (**LOW**) dan PIN 5 diberi logika (**LOW**) sehingga putaran yang ada pada motor akan berhenti (**STOP**). Pada gambar 4.15 akan menunjukkan kondisi putaran saat tombol STOP (berhenti) pada GUI ditekan.



Gambar 4.15 Kondisi Putaran saat tombol STOP ditekan

Pada tabel 4.3 merupakan hasil dari pengujian tombol STOP pada program GUI ke rangkaian Motor DC

Tabel 4.3 Pengujian Tombol STOP (berhenti)

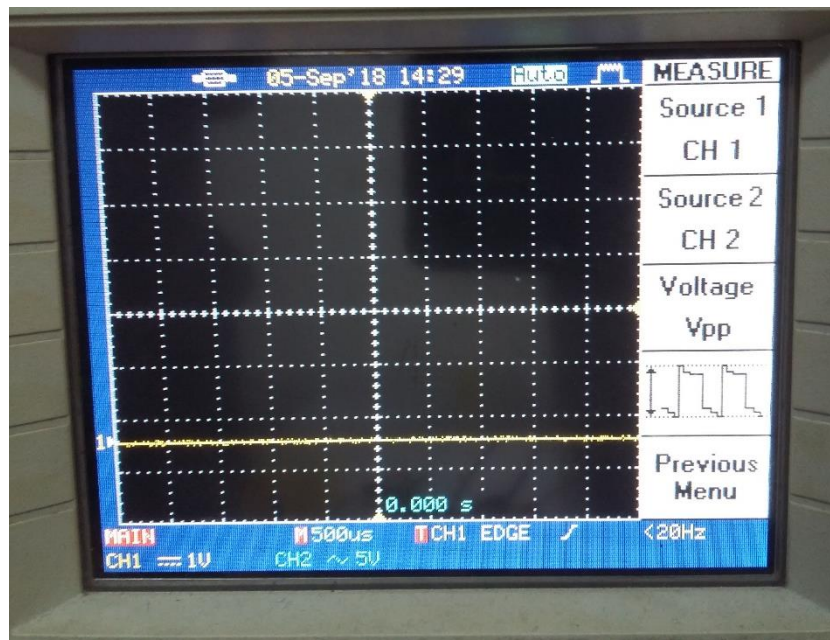
No	Nilai PWM	Kondisi Motor DC	Keterangan
1	0-10	Berhenti	Tidak Berputar
2	11-24	Berhenti	Tidak Berputar
3	25-30	Berhenti	Tidak Berputar
4	31-40	Berhenti	Tidak Berputar
5	41-50	Berhenti	Tidak Berputar
6	51-60	Berhenti	Tidak Berputar
7	61-90	Berhenti	Tidak Berputar
8	91-120	Berhenti	Tidak Berputar
9	121-150	Berhenti	Tidak Berputar

10	151-180	Berhenti	Tidak Berputar
11	181-255	Berhenti	Tidak Berputar

Dari hasil data pengujian pada tabel 4.3 dapat dilihat bahwa saat tombol STOP ditekan maka putaran motor akan berhenti. Hal ini dikarenakan saat tombol STOP ditekan, GUI mengirimkan perintah berupa data karakter “S” dan nilai PWM 0, sehingga berapapun kecepatan yang di atur, nilai PWM yang akan diterima oleh Arduino tetap 0.

4.2.3 Hasil Pengujian Sinyal PWM menggunakan Osiloskop

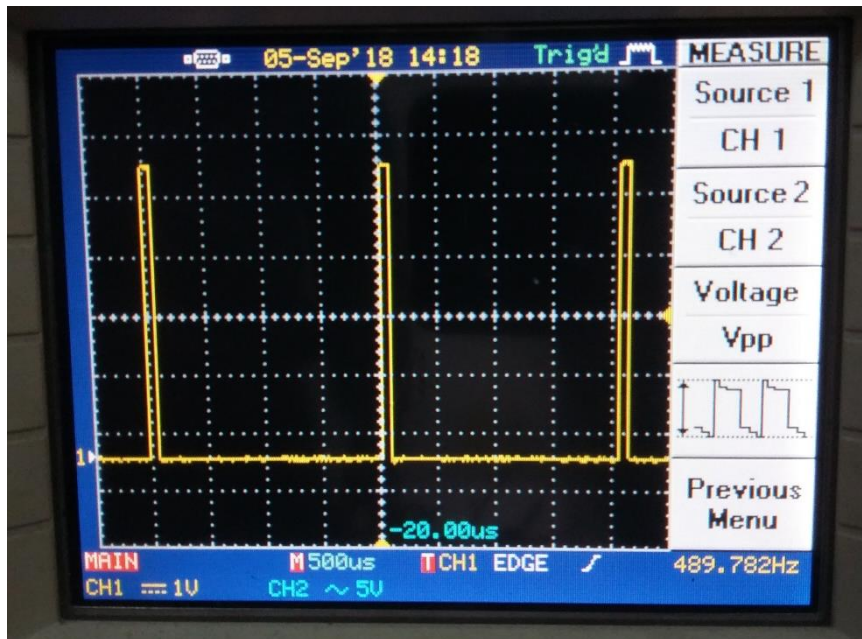
- 1) Telah dilakukan pengujian sinyal PWM dengan osiloskop sebenarnya. Dari pengujian yang telah dilakukan tampak bahwa mula-mula diberikan posisi awal motor tidak bergerak (Nilai PWM 0) tampak bahwa tampilan sinyal di osiloskop tampak datar (horisontal).



Gambar 4.16 Sinyal PWM pada Osiloskop saat Nilai PWM 0

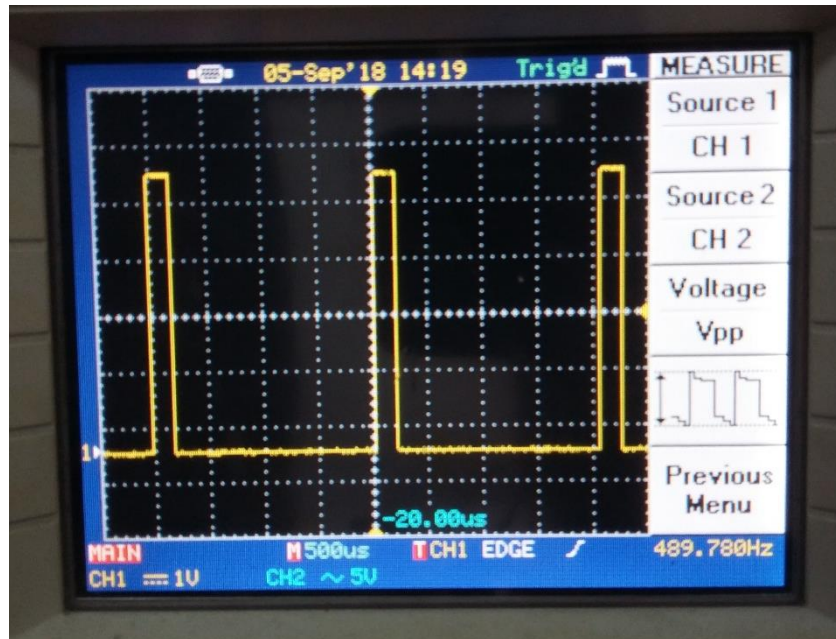
- 2) Seiring penambahan nilai sinyal PWM, maka akan tampak muncul sinyal gelombang kotak di osiloskop. Dengan jumlah kotak vertical sebanyak 5kolom dengan setting 1 V/div. Hal ini membuktikan bahwa sinyal PWM berupa digital

sudah mulai bekerja. Gambar 4.17, Gambar 4.18, Gambar 4.19, Gambar 4.20, Gambar 4.21, Gambar 4.22, Gambar 4.23, Gambar 4.24, Gambar 4.25, Gambar 4.26, Gambar 4.27, Gambar 4.28 menunjukkan adanya perubahan pada sinyal yang ditampilkan pada Osiloskop saat nilai PWM diubah, mulai dari nilai PWM 0 hingga nilai PWM 255.



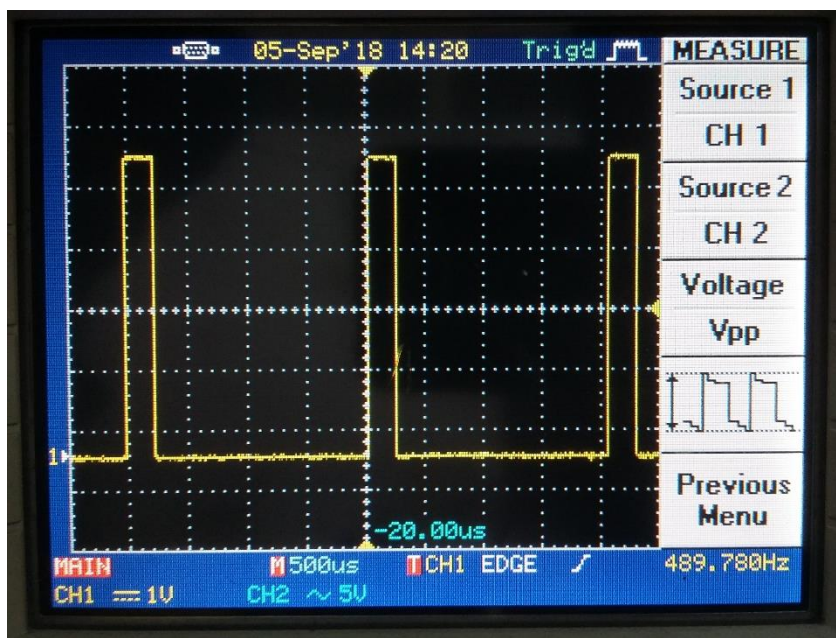
Gambar 4.17 Sinyal PWM pada Osiloskop saat Nilai PWM 10

Berdasarkan pada Gambar 4.17 saat nilai PWM di set menjadi 10, maka duty cyclenya yaitu sebesar 3,92%



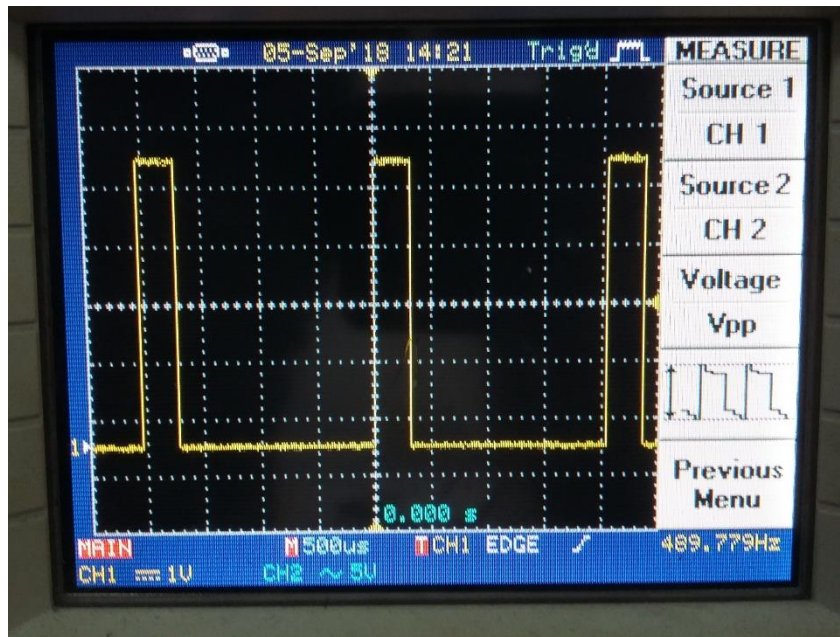
Gambar 4.18 Sinyal PWM pada Osiloskop saat Nilai PWM 24

Berdasarkan pada Gambar 4.18 saat nilai PWM di set menjadi 24, maka duty cyclenya yaitu sebesar 9,41%



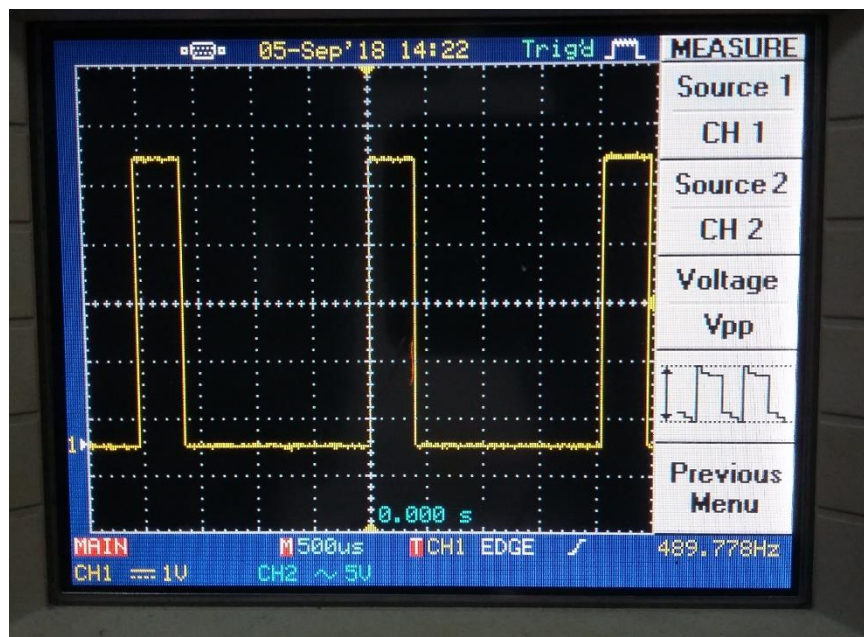
Gambar 4.19 Sinyal PWM pada Osiloskop saat Nilai PWM 30

Berdasarkan pada Gambar 4.19 saat nilai PWM di set menjadi 30, maka duty cyclenya yaitu sebesar 11,76%



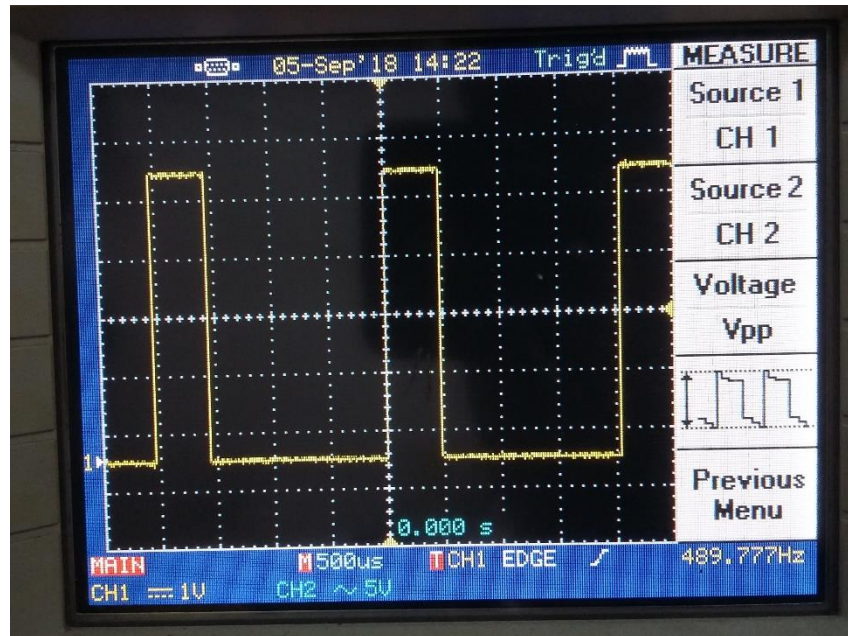
Gambar 4.20 Sinyal PWM pada Osiloskop saat Nilai PWM 40

Berdasarkan pada Gambar 4.20 saat nilai PWM di set menjadi 10, maka duty cyclenya yaitu sebesar 15,68%



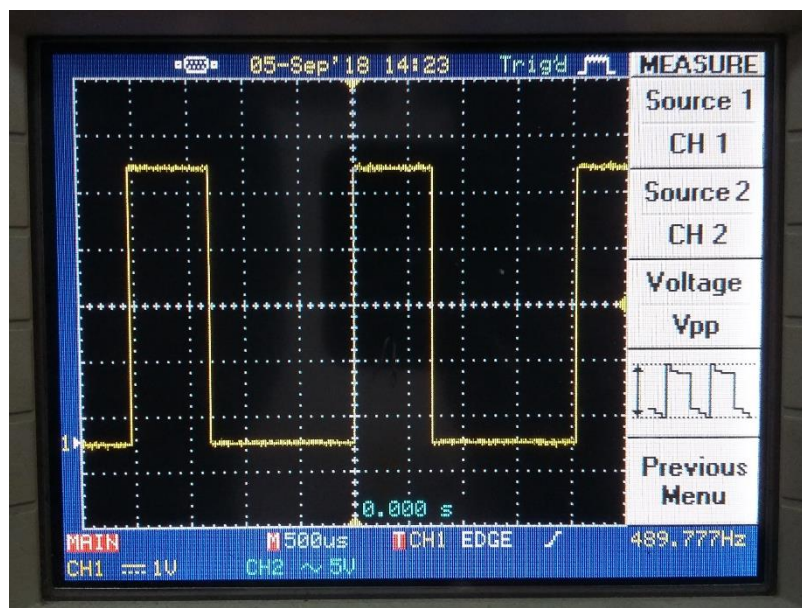
Gambar 3.21 Sinyal PWM pada Osiloskop saat Nilai PWM 49

Berdasarkan pada Gambar 4.21 saat nilai PWM di set menjadi 49, maka duty cyclenya yaitu sebesar 19,21%



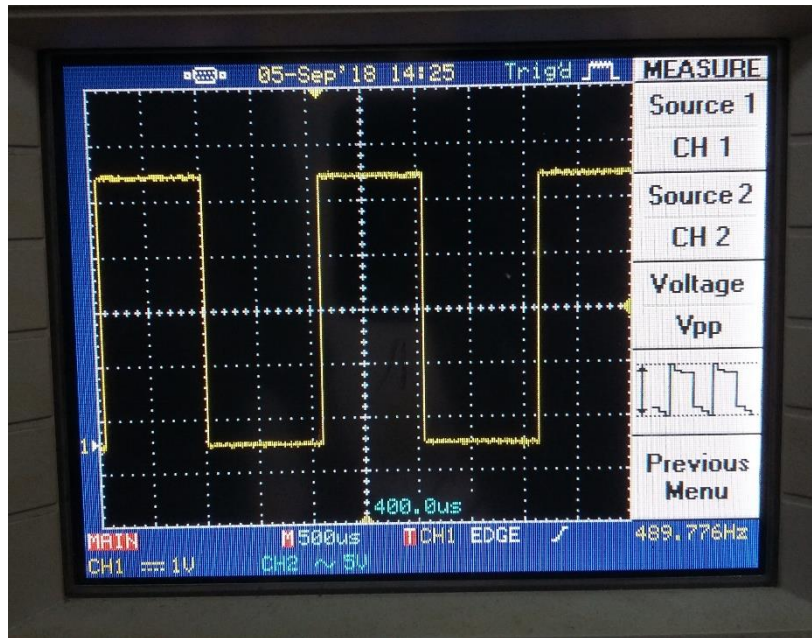
Gambar 4.22 Sinyal PWM pada Osiloskop saat Nilai PWM 60

Berdasarkan pada Gambar 4.22 saat nilai PWM di set menjadi 60, maka duty cyclenya yaitu sebesar 23,52%



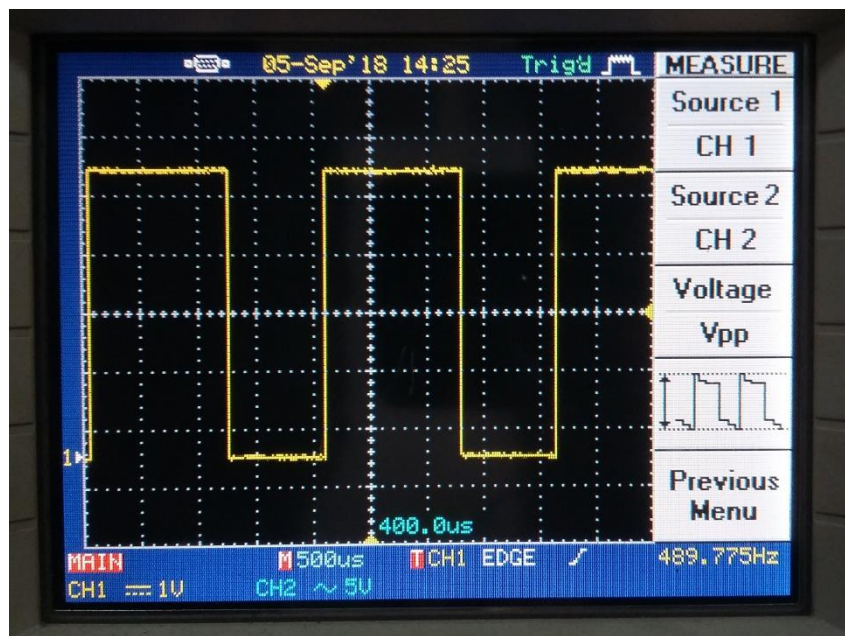
Gambar 4.23 Sinyal PWM pada Osiloskop saat Nilai PWM 120

Berdasarkan pada Gambar 4.23 saat nilai PWM di set menjadi 120, maka duty cyclenya yaitu sebesar 47,05%



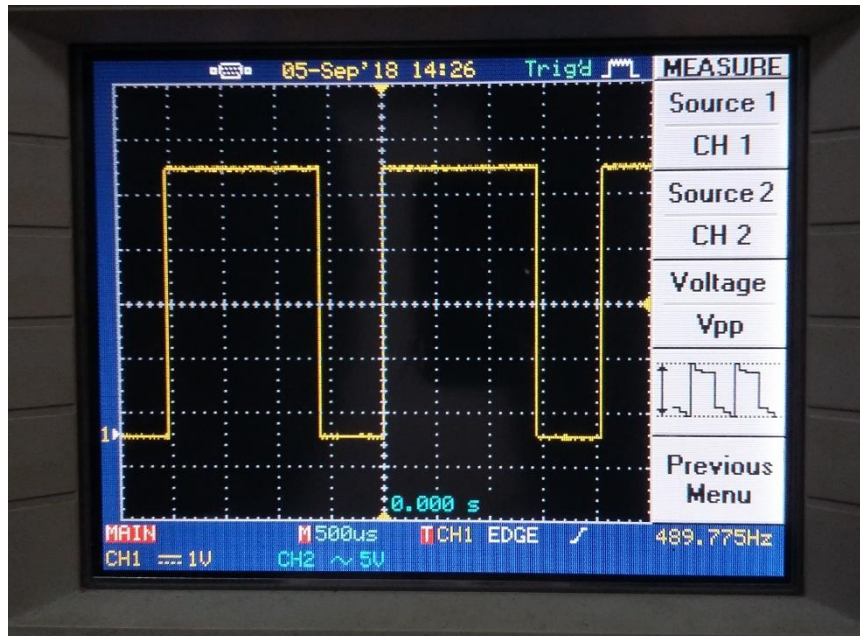
Gambar 4.24 Sinyal PWM pada Osiloskop saat Nilai PWM 130

Berdasarkan pada Gambar 4.24 saat nilai PWM di set menjadi 10, maka duty cyclenya yaitu sebesar 50,98%



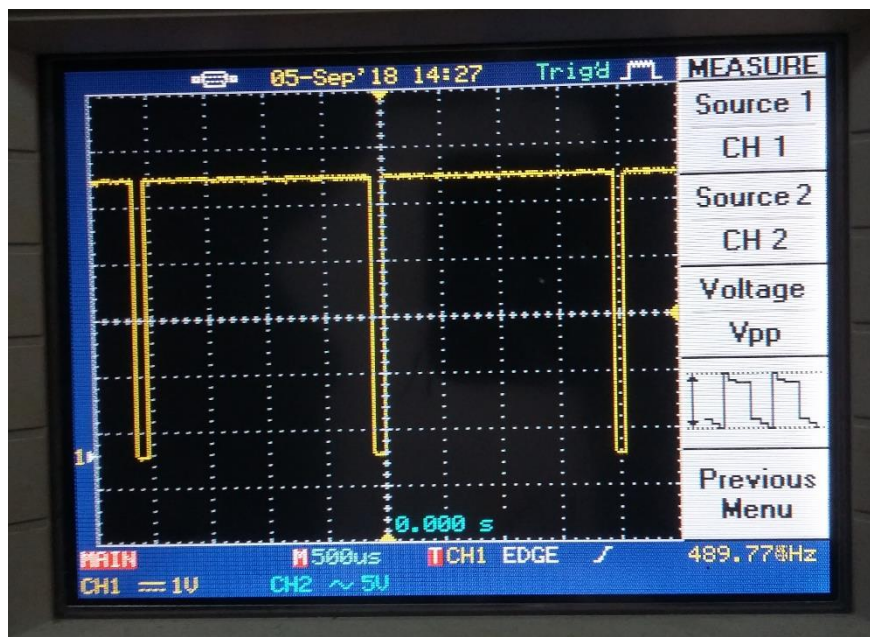
Gambar 4.25 Sinyal PWM pada Osiloskop saat Nilai PWM 150

Berdasarkan pada Gambar 4.25 saat nilai PWM di set menjadi 150, maka duty cyclenya yaitu sebesar 58,82%



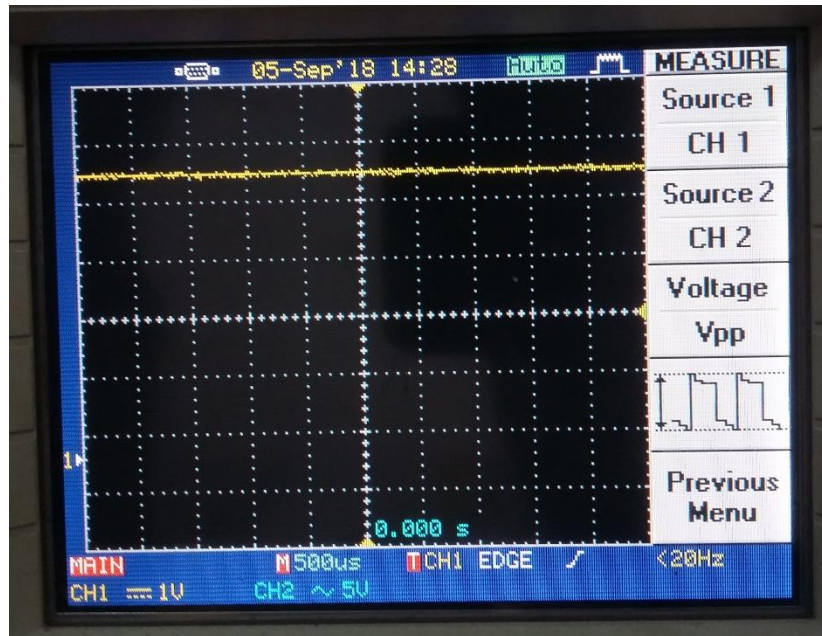
Gambar 4.26 Sinyal PWM pada Osiloskop saat Nilai PWM 180

Berdasarkan pada Gambar 4.26 saat nilai PWM di set menjadi 180, maka duty cyclenya yaitu sebesar 70,58%



Gambar 4.27 Sinyal PWM pada Osiloskop saat Nilai PWM 243

Berdasarkan pada Gambar 4.27 saat nilai PWM di set menjadi 10, maka duty cyclenya yaitu sebesar 95,29%



Gambar 4.28 Sinyal PWM pada Osiloskop saat Nilai PWM 255

Berdasarkan pada Gambar 4.17 saat nilai PWM di set menjadi 10, maka duty cyclenya yaitu sebesar 100%

- 3) Seiring bertambahnya nilai PWM , dapat terlihat juga bahwa duty cycle dari gelombang tersebut tampak semakin melebar, hal ini dilihat dari pembacaan periode pada osiloskop.