BAB IV

PENGUJIAN DAN ANALISIS

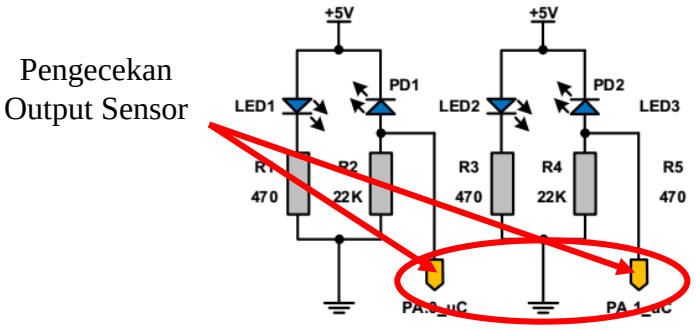
Dalam bab ini akan dijelaskan mengenai proses pengujian dari sistem yang telah dirancang sebelumnya. Dimana dalam pengujian yang akan dilakukan ini nantinya berupa pengukuran terhadap komponen dari setiap perangkat mulai dari bagian masukan (*input*), proses (*process*) dan keluaran (*output*) yang terdapat di dalam sistem yang dirancang. Setelah melakukan proses pengujian komponen, berikutnya dilanjutkan dengan menganalisa setiap hasil pengecekan serta pengukuran. Hal tersebut dilakukan dengan tujuan untuk melihat dan memastikan bahwa sistem yang dirancang dapat sesuai dengan apa yang diharapkan.

4.1 Pengujian dan Analisis Perangkat Masukan (*Input*)

Pengujian dan analisis bagian masukan (*input*) ini terdiri dari pengujian pendeteksian sensor *photodiode*, *keypad* 3x4 dan 2-*push button switch*. Pengujian ini dilakukan untuk mendapatkan parameter dari setiap komponen masukan tersebut. Nilai parameter ini dapat menjadi acuan untuk melihat apakah komponen masukan tersebut dapat bekerja sesuai dengan fungsinya atau tidak.

4.1.1 Pengujian Sensor *Photodiode*

Sensor *photodiode* merupakan bagian komponen yang paling penting dari robot, dimana sensor ini difungsikan untuk mendeteksi garis (*line*) pada lintasan. Pengujian pada sensor *photodiode* ini dilakukan untuk mengetahui nilai tegangan keluaran dari sensor saat kondisi aktif pada saat mendeteksi garis. Berikut ini adalah tabel pengujian terhadap sensor *photodiode*.



Gambar 4.1 Pengujian Sensor *Photodiode*

Tabel 4.1 Pengujian Deteksi Sensor *Photodiode* terhadap Garis

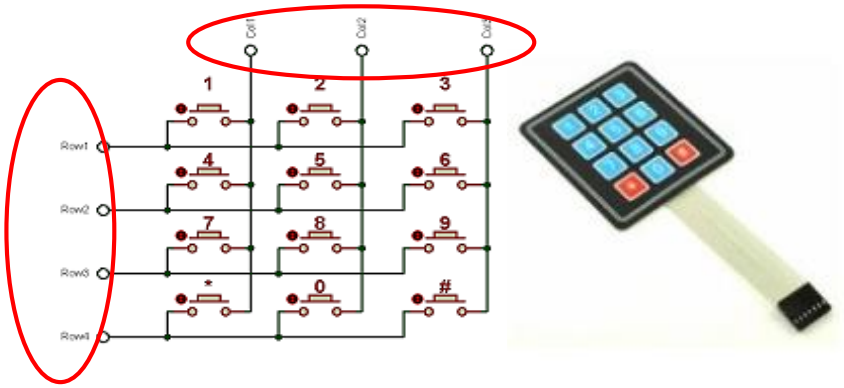
Jenis Sensor	Kondisi	Logika	Hasil
Photodiode (Kiri2)	Aktif	1	2,27 *Volt*
	Tidak Aktif	0	0,02 *Volt*
Photodiode (Kiri1)	Aktif	1	2,32 *Volt*
	Tidak Aktif	0	0,02 *Volt*
Photodiode (Tengah)	Aktif	1	2,40 *Volt*
	Tidak Aktif	0	0,01 *Volt*
Photodiode (Kanan1)	Aktif	1	2,32 *Volt*
	Tidak Aktif	0	0,03 *Volt*
Photodiode (Kanan2)	Aktif	1	2,27 *Volt*
	Tidak Aktif	0	0,01 *Volt*

Berdasarkan dari hasil pengujian terhadap sensor *photodiode*, dapat dianalisa bahwa ketika sensor *photodiode* ini mendeteksi garis berwarna hitam, maka akan berlogika 1 (*high*) dan memiliki nilai tegangan rata-rata sekitar 2 *Volt* saat mendeteksi garis. Nilai parameter inilah yang nantinya akan diolah oleh mikrokontroler melalui masukan *port* ADC.

4.1.2 *Keypad* 3x4

Dilihat dari prinsip kerjanya, *keypad* ukuran 3x4 ini difungsikan sebagai masukan (*input*) ke bagian pemroses data yaitu mikrokontroler. Mikrokontroler akan membaca data dari *keypad* melalui proses *scanning* pada *keypad*. Proses *scanning* ini maksudnya adalah bahwa mikrokontroler akan men-*scan* dan

mendeteksi ada atau tidaknya penekanan tombol pada *keypad* yang akan dijadikan sebagai masukan (*input*).



Gambar 4.2 Pengujian Penekanan Tombol pada *Keypad* Matriks 3x4

Setiap penekanan salah satu tombol pada *keypad* akan terjadi kombinasi antara *pin input* (baris) dan *pin output* (kolom) pada susunan matriks *keypad*. Berikut ini adalah pengujian penekanan tombol pada *keypad* berdasarkan kombinasi baris dan kolom dari tombol-tombol yang ada pada *keypad*.

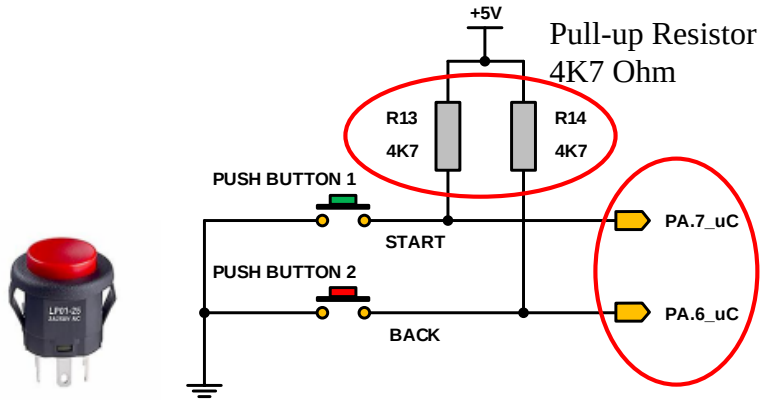
Tabel 4.2 Pengujian Penekanan Tombol *Keypad*

Jenis Pengujian	Kombinasi (Baris-Kolom)	Kondisi
Penekanan Tombol 1	B1 dan K1	Terhubung
Penekanan Tombol 2	B1 dan K2	Terhubung
Penekanan Tombol 3	B1 dan K3	Terhubung
Penekanan Tombol 4	B2 dan K1	Terhubung
Penekanan Tombol 5	B2 dan K2	Terhubung
Penekanan Tombol 6	B2 dan K3	Terhubung
Penekanan Tombol 7	B3 dan K1	Terhubung
Penekanan Tombol 8	B3 dan K2	Terhubung
Penekanan Tombol 9	B3 dan K3	Terhubung
Penekanan Tombol *	B4 dan K1	Terhubung
Penekanan Tombol 0	B4 dan K2	Terhubung
Penekanan Tombol #	B4 dan K3	Terhubung

Berdasarkan dari hasil pengujian penekanan tombol-tombol pada *keypad* bahwa keseluruhan tombol pada *keypad* dapat berfungsi dengan baik, sehingga dapat digunakan sebagai tombol masukan nomor meja pelanggan.

4.1.3 Sakelar Tekan (*Push Button Switch*)

Secara umum, sakelar tekan atau *push button switch* ini merupakan jenis sakelar yang dapat menghubungkan aliran listrik hanya sesaat saja saat ditekan dan setelah dilepas maka akan kembali ke posisi *off*. Pada perancangan *prototype* robot pengantar makanan ini terdapat 2 buah *push button switch* yang berfungsi sebagai tombol (*start*) untuk memulai pergerakan robot untuk mengantarkan makanan menuju meja pelanggan. Sedangkan tombol (*back*) untuk memerintahkan robot untuk kembali posisi awalnya (*home*).



Gambar 4.3 Pengujian Tombol *Push Button Switch*

Tabel 4.3 Pengujian Penekanan Tombol *Push Button Switch*

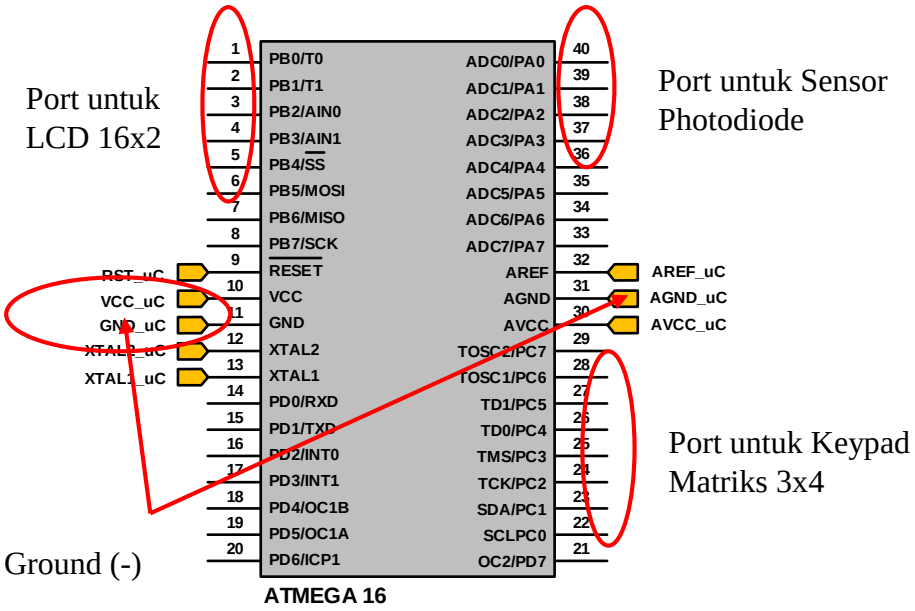
Jenis Pengujian	Posisi Tombol	Logika	Kondisi
Tombol Hijau	Ditekan	1	*High*
	Tidak Ditekan	0	*Low*
Tombol Merah	Ditekan	1	*High*
	Tidak Ditekan	0	*Low*

Pada gambar di atas dapat dilihat bahwa rangkaian kedua *push button switch* ini menggunakan *resistor pull-up* dengan nilai 4K7 Ohm yang salah satu bagian *pin* nya terhubung langsung ke tegangan sumber +5 Vdc sedangkan bagian *pin* yang satunya ke *pin* sakelar (*switch*). Kedua tombol *push button switch* ini bila dalam kondisi aktif atau adanya tombol yang ditekan, maka *pin* keluaran dari sakelar akan berlogika '1' atau berada dalam kondisi (*high*). Apabila tidakada tombol yang ditekan, maka akan berlogika '0' atau berada dalam kondisi (*low*).

4.2 Pengujian dan Analisis Perangkat Proses (*Process*)

Pengujian rangkaian sistem minimum mikrokontroler AVR ATmega 16 ini ditujukan untuk menentukan parameter nilai tegangan *Input-Output* (I/O) dari mikrokontroler. Berdasarkan *datasheet*, mikrokontroler ATmega 16 dapat beroperasi pada rentang tegangan dari 2,7 – 5,5 *Volt*.

Pengujian dilakukan dengan memberikan tegangan masukan sebesar 5Vdc pada mikrokontroler ATmega 16 melalui *pin* 10 (VCC) serta *pin* 11 dan *pin* 31 (Ground/GND) pada mikrokontroler ATmega 16. Pengujian kedua yaitu dengan melakukan pengetesan keempat *port* yang ada pada mikrokontroler ATmega 16 agar nantinya dapat digunakan sebagaimana mestinya.



Gambar 4.4 Pengujian Kinerja Mikrokontroler ATmega 16

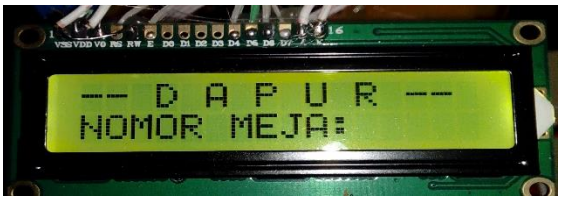
4.3 Pengujian dan Analisis Perangkat Keluaran (*Output*)

Pengujian dan analisis yang dilakukan selanjutnya adalah melakukan pengukuran serta pengujian terhadap perangkat keluaran (*output*). Bagian keluaran yang akan diuji terdiri pengujian *Liquid Crystal Display* (LCD), *driver* motor L293D dan juga motor DC penggerak roda. Pengujian serta analisis yang dilakukan ini bertujuan untuk mendapatkan parameter-parameter yang diperlukan yang diperlukan dari setiap komponen keluaran tersebut. Disamping itu juga untuk mengetahui kinerja komponen-komponen dari perangkat keluaran ini agar dapat bekerja sesuai dengan fungsinya.

4.3.1 *Liquid Crystal Display* (LCD) *Character* 16x2

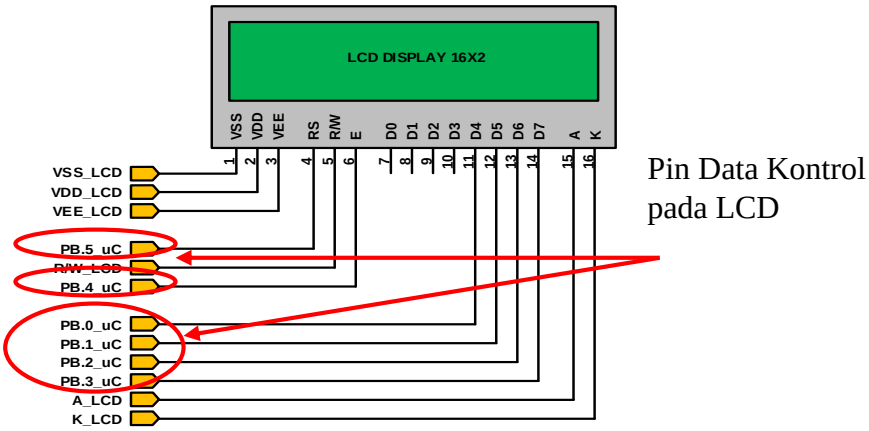
Pengujian serta analisis mengenai *Liquid Crystal Display* (LCD) *character* 16x2 ini dilakukan agar LCD ini dapat menampilkan karakter huruf dan angka yang sesuai dengan keinginan. Jika ingin menampilkan beberapa karakter huruf dan

angka pada layar LCD, maka mikrokontroler harus diprogram dengan program yang khusus digunakan untuk mengontrol tampilan karakter pada LCD.



Gambar 4.5 Tampilan Karakter Huruf pada LCD 16x2

Untuk dapat digunakan sebagai penampil karakter huruf dan angka, terlebih dahulu harus dikonfigurasikan beberapa *pin* pada LCD menuju ke mikrokontroler. Di dalam perancangan, rangkaian LCD ini memiliki beberapa *pin* kontrol yang dapat digunakan antara lain *pin* RS, E, 4-bit data (D4, D5, D6, D7) agar lebih jelas dapat dilihat pada Gambar 4.7.



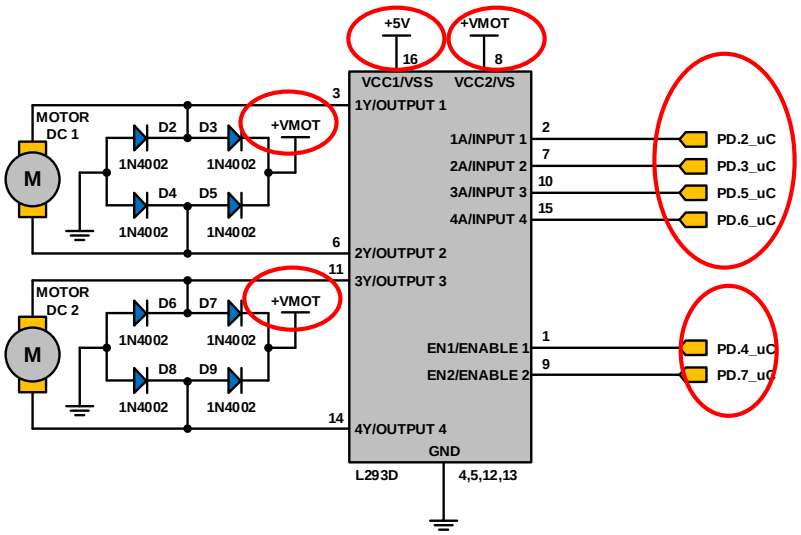
Gambar 4.6 Pengujian *pin-pin* Pengontrol Data pada LCD 16x2

4.3.2 *Driver* Motor L293D

Untuk mengontrol atau mengendalikan arah pergerakan motor DC dengan menggunakan mikrokontroler diperlukan sebuah rangkaian *driver* motor DC. Sebenarnya mikrokontroler tidak bisa mengontrol/ menggerakkan motor secara

langsung, karena motor DC memerlukan sumber tegangan yang cukup besar untuk dapat bergerak, sehingga penggunaan *driver* motor ini sangat disarankan agar dapat mengontrol seluruh aktifitas pergerakan motor DC. *Driver* motor DC yang digunakan dalam perancangan sebelumnya adalah menggunakan IC *driver* L293D. Dimana *driver* motor ini memiliki kemampuan mengalirkan arus 1 ampere pada setiap *driver*-nya. Sehingga dapat digunakan untuk membuat rangkaian *driver Half Bridge* (*H-Bridge*) untuk mengontrol 2 buah motor DC.

Pengujian serta analisis ini dilakukan untuk mengetahui apakah komponen *driver* motor DC ini bekerja dengan baik atau tidak. Pengujian diawali dengan mengonfigurasikan *pin input driver* motor dengan mikrokontroler yang dapat dilihat pada Gambar 4.7.



Gambar 4.7 Konfigurasi *Pin Driver* Motor dengan Mikrokontroler

Bila dilihat pada gambar di atas, terdapat dua sumber tegangan *input* yang berbeda diantaranya adalah *pin* (Vss) digunakan untuk tegangan sumber *driver* motor DC yaitu sebesar 5 Vdc. Sedangkan *pin* (Vs) adalah jalur *input* sumber tegangan untuk

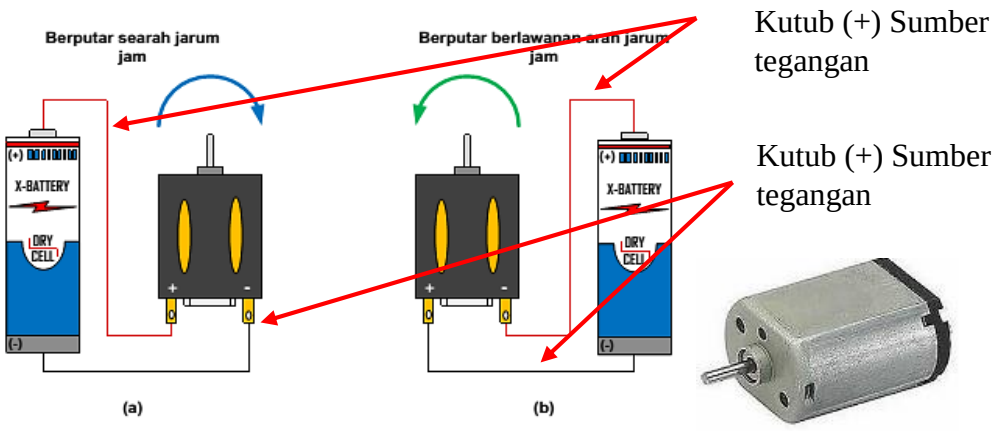
motor DC yang dikendalikan berkisar antara 12 Vdc tergantung dari jenis motor DC yang digunakan.

4.3.3 Motor DC

Pengujian dan analisis selanjutnya ditujukan pada motor DC sebagai penggerak roda pada *prototype* robot pengantar makanan. Untuk mengetahui arah pergerakan motor DC yang benar maka harus dilakukan pengujian arah putaran motor DC, agar nantinya saat digunakan dapat berjalan dengan baik. Pengujian motor DC ini dilakukan dengan cara menghubungkan sumber tegangan dengan kutub (+) dan kutub (-) pada motor DC.

Terdapat dua pengujian pergerakan motor DC yaitu, pengujian pertama dengan mengubungkan sumber tegangan dengan motor DC dengan benar seperti sumber tegangan positif (+) dihubungkan ke bagian kutub (+) motor. Sedangkan sumber tegangan negatif (-) ke bagian kutub (-) agar motor DC dapat bergerak normal (searah jarum jam).

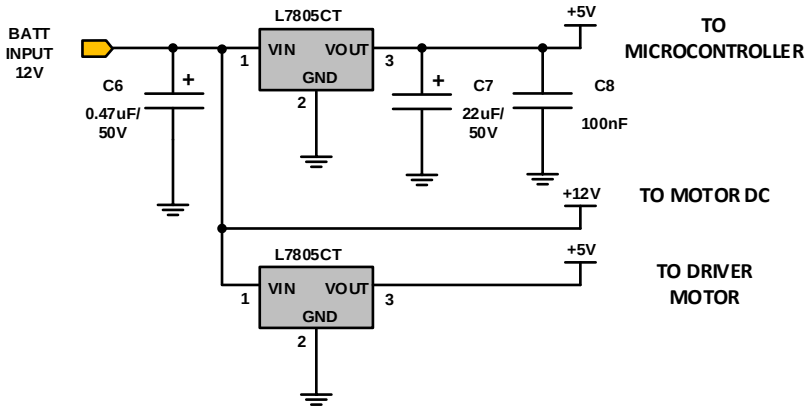
Pengujian kedua, dengan menghubungkan kebalikannya yaitu menghubungkan sumber tegangan positif (+) ke bagian kutub (-) pada motor DC. Sedangkan untuk sumber tegangan negatif (-) dihubungkan ke bagian kutub (+) pada motor DC, sehingga motor DC akan bergerak kebalikannya dari sebelumnya dikarenakan poralitas yang terbalik (berlawanan arah jarum jam). Berikut ini merupakan pengujian pergerakan serta arah putaran pada motor DC yang ditunjukkan pada Gambar 4.8.



Gambar 4.8 Pengujian Arah Pergerakan pada Motor DC

4.4 Pengujian dan Analisis Catu Daya (*Power Supply*)

Pengujian rangkaian catu daya ini bertujuan untuk mengetahui kinerja dari catu daya dalam mensuplai perangkat *prototype* robot pengantar makanan. Pemilihan catu daya yang benar sangat disarankan karena nantinya akan mempengaruhi kinerja dari prototype robot ketika akan dijalankan. Sumber tegangan untuk mensuplai rangkaian robot ini berasal dari sebuah baterai *rechargeable* (*accumulator*) dengan spesifikasi tegangan sebesar 12 Volt dan arus sebesar 1,2Ah.



Gambar 4.9 Pembagian Tegangan Keluaran pada Catu Daya

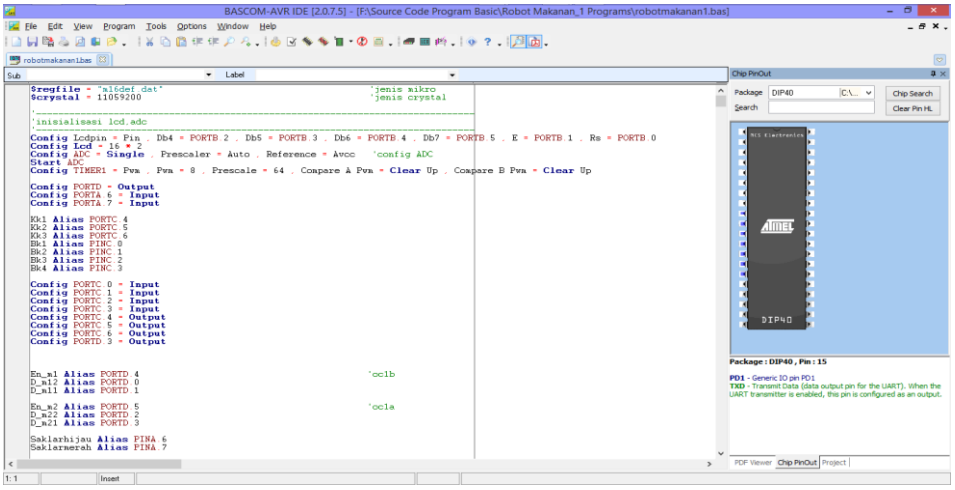
Tabel 4.4 Pengujian Tegangan Keluaran Pada Catu Daya

Jenis Regulator	Sumber Tegangan	Baterai
L7805 (1)	*Input* (Vdc)	12,03 *Volt*
	Output (Vdc)	5,00 *Volt*
-	*Input* (Vdc)	12,03 *Volt*
	Output (Vdc)	12,03 *Volt*
L7805 (2)	*Input* (Vdc)	12,03 *Volt*
	Output (Vdc)	5,00 *Volt*

4.5 Pengujian Perangkat Lunak (*Software*)

Pengujian perangkat lunak (*Software*) ini bertujuan untuk mengetahui apakah *source code* program yang telah dibuat tersebut masih terdapat kesalahan (*error*) atau tidak. Kemudian pengujian ini juga untuk menguji perangkat lunak tersebut dapat sesuai dengan kinerja dari perangkat keras (*hardware*) yang diharapkan atau tidak. Oleh sebab itu, maka akan dijelaskan selengkapnya mengenai langkah-langkah pengujian perangkat lunak sebagai berikut :

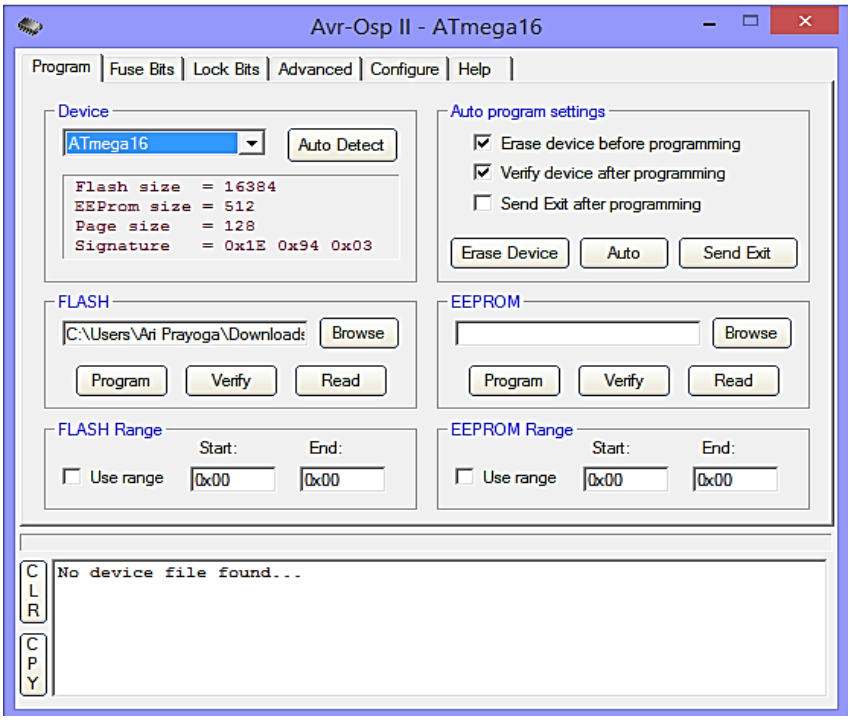
1. Pertama, menjalankan program editor *Basic Compiler* (*Bascom*-AVR) maka akan muncul tampilan program *Bascom*-AVR.



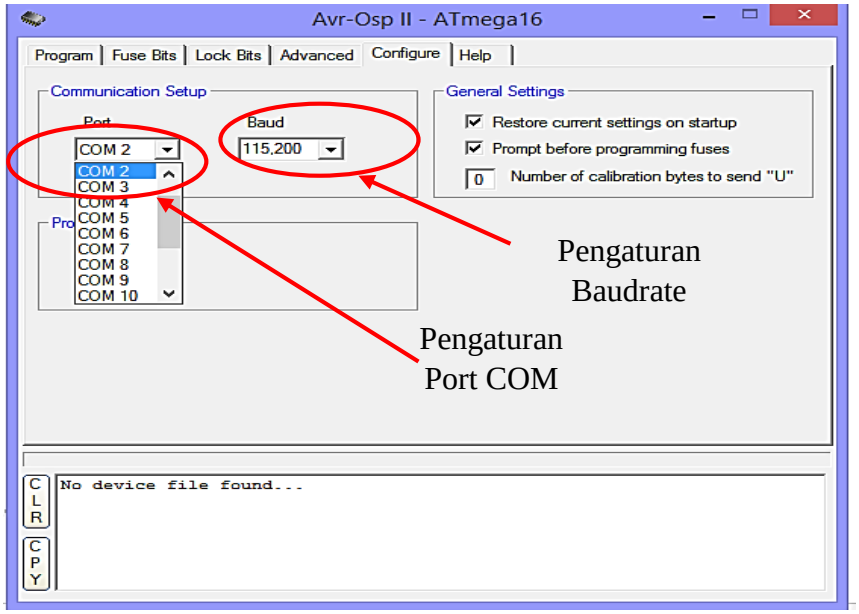
Gambar 4.10 Program *Bascom*-AVR

Bascom-AVR merupakan sebuah program yang dikhususkan untuk melakukan pemrograman bahasa *basic*. Dimana *Bascom*-AVR ini dapat digunakan untuk memprogram mikrokontroler AVR. Pada program *Bascom* ini kita dapat membuat dan melakukan pengeditan program dengan kode *basic* dan juga dapat melakukan kompilasi terhadap program yang dibuat. Apabila pada saat melakukan kompilasi terjadi kesalahan penulisan *syntax* program, maka kita akan diberitahukan oleh program tersebut bahwa terjadi kesalahan atau ada baris program yang *error*. Biasanya pesan *error* ini dapat dilihat pada bagian bawah program.

2. Jika pengeditan program telah selesai dilakukan, maka selanjutnya akan dilakukan kompilasi (*compile*) program yaitu mengubah sebuah program yang sebelumnya berekstensi (.bas) menjadi bentuk file heksa (.hex). dimana file ini nantinya akan dimasukkan atau dalam istilah lain yaitu di *download* ke dalam *chip* mikrokontroler. Untuk dapat memasukkan program .hex ini ke dalam mikrokontroler diperlukan sebuah program downloader AVROSP II. Program ini berguna untuk mengisikan file .hex yang telah siap tadi ke dalam *chip* mikrokontroler. Berikut ini adalah tampilan program AVROSP II yang dtunjukkan pada Gambar 4.11



Gambar 4.11 Program AVROSP II



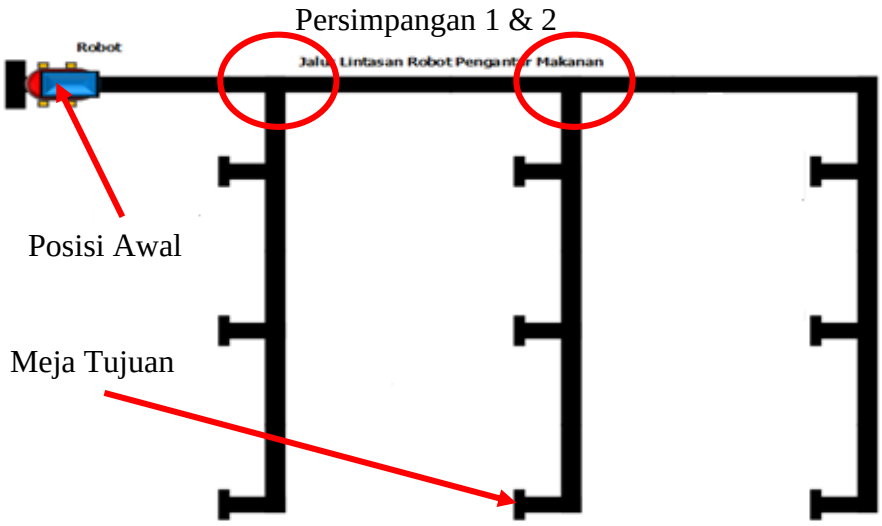
Gambar 4.12 Pengaturan Konfigurasi *Port* COM dan *Baudrate*

Pada Gambar 4.11 di atas adalah tampilan program dari AVROSP II, dimana dengan program ini para *programmer* dapat dengan mudah melakukan proses

pengisian program ke dalam mikrokontroler. Tetapi sebelum melakukan proses pengisian program, ada hal yang perlu diperhatikan pada Gambar 4.12 di atas bahwa sebelumnya kita harus mengatur *Port* COM dan *baudrate* yang digunakan sehingga bila sudah benar maka proses pengisian program dapat berjalan dengan lancar.

4.6 Pengujian Pergerakan Robot pada Media Garis Lintasan

Dari hasil pengujian sebelumnya yang telah dilakukan, pengujian tahap ini adalah pengujian terakhir yang dilakukan untuk mengecek pergerakan dari robot, apakah berjalan sesuai dengan yang diharapkan atau tidak. Uji coba ini dilakukan untuk menentukan tingkat keberhasilan dari robot yang dalam aplikasinya sebagai robot pengantar makanan yang ditujukan ke meja pelanggan. Berikut ini adalah media garis lintasan yang akan di uji coba dengan menggunakan *prototype* robot pengantar makanan.



Gambar 4.13 Uji Coba Robot Pengantar Makanan pada Lintasan

Tabel 4.5 Pengujian Pergerakan Robot Pengantar Makanan

Masukan Perintah	Uji Coba	Waktu Antar	Waktu Kembali	Keterangan
Robot Antar ke Meja 1	Ke-1	42,20 detik	46,04 detik	Uji Coba Berhasil
	Ke-2	42,19 detik	46,01 detik	Uji Coba Berhasil
	Ke-3	41, 98 detik	45,97 detik	Uji Coba Berhasil
Robot Antar ke Meja 2	Ke-1	50,06 detik	54,03 detik	Uji Coba Berhasil
	Ke-2	49,99 detik	54,01 detik	Uji Coba Berhasil
	Ke-3	49,95 detik	51,98 detik	Uji Coba Berhasil
Robot Antar ke Meja 3	Ke-1	57,68 detik	1,03 menit	Uji Coba Berhasil
	Ke-2	57,55 detik	59,99 detik	Uji Coba Berhasil
	Ke-3	57,53 detik	59,97 detik	Uji Coba Berhasil
Robot Antar ke Meja 4	Ke-1	50,04 detik	59,26 detik	Uji Coba Berhasil
	Ke-2	50,01 detik	59,24 detik	Uji Coba Berhasil
	Ke-3	49,98 detik	59,20 detik	Uji Coba Berhasil
Robot Antar ke Meja 5	Ke-1	58,10 detik	1,02 menit	Uji Coba Berhasil
	Ke-2	58,07 detik	1,02 menit	Uji Coba Berhasil
	Ke-3	58,05 detik	1,00 menit	Uji Coba Berhasil
Robot Antar ke Meja 6	Ke-1	1,08 menit	1,50 menit	Uji Coba Berhasil
	Ke-2	1,06 menit	1,48 menit	Uji Coba Berhasil
	Ke-3	1,04 menit	1,45 menit	Uji Coba Berhasil
Robot Antar ke Meja 7	Ke-1	1,02 menit	1,50 menit	Uji Coba Berhasil
	Ke-2	1,01 menit	1,45 menit	Uji Coba Berhasil
	Ke-3	1,00 menit	1,40 menit	Uji Coba Berhasil
Robot Antar ke Meja 8	Ke-1	1,08 menit	1,79 menit	Uji Coba Berhasil
	Ke-2	1,07 menit	1,78 menit	Uji Coba Berhasil
	Ke-3	1,07 menit	1,76 menit	Uji Coba Berhasil
Robot Antar ke Meja 9	Ke-1	1,14 menit	1,97 menit	Uji Coba Berhasil
	Ke-2	1,13 menit	1,96 menit	Uji Coba Berhasil
	Ke-3	1,13 menit	1,96 menit	Uji Coba Berhasil

Dari tabel di atas diketahui bahwa untuk keseluruhan pengantaran sudah cukup berhasil dilakukan oleh *prototype* robot pengantar makanan. Kemudian disini terdapat selisih perbedaan waktu tempuh, hal ini dikarenakan oleh beberapa kemungkinan-kemungkinan yang bisa saja terjadi diantaranya karena adanya faktor teknis dari sistem penggerak roda, permukaan lintasan tidak datar, atau bisa juga karena sumber tegangan catu daya tidak stabil sehingga mempengaruhi kinerja dari robot saat berjalan di garis lintasan.