

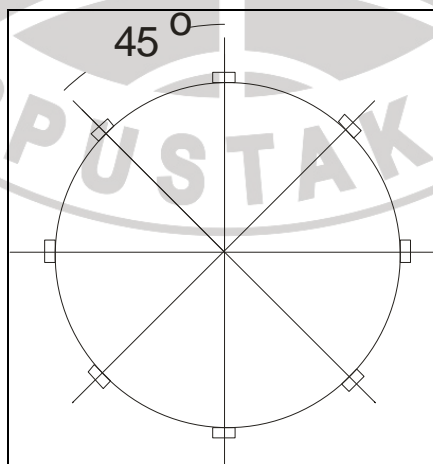
BAB III

PERANCANGAN PEMBUATAN PERANGKAT KERAS DAN PERANGKAT LUNAK

3.1 Perancangan dan Pembuatan Perangkat Keras (*Hardware*)

Perangkat keras yang akan dibuat adalah :

- Delapan sensor inframerah yang akan dipasang melingkar pada robot mobil dengan jarak antar sensor adalah 45° seperti yang ditunjukkan pada gambar 3.1.
- Delapan buah pembangkit pulsa (pengirim) dan delapan buah penerima inframerah. Sistem sensor infra merah yang menggunakan timer 555 untuk pulsa pengirimnya dan menghasil frekuensi keluaran sebesar 40 KHz.
- Pengendali sistem sensor inframerah menggunakan mikrokontroler AT89S51 dengan frekuensi clock 11,0592 MHz yang dihasilkan dari oscillator Kristal.
- Driver motor DC sebagai pengatur pergerakan motorik dari robot.



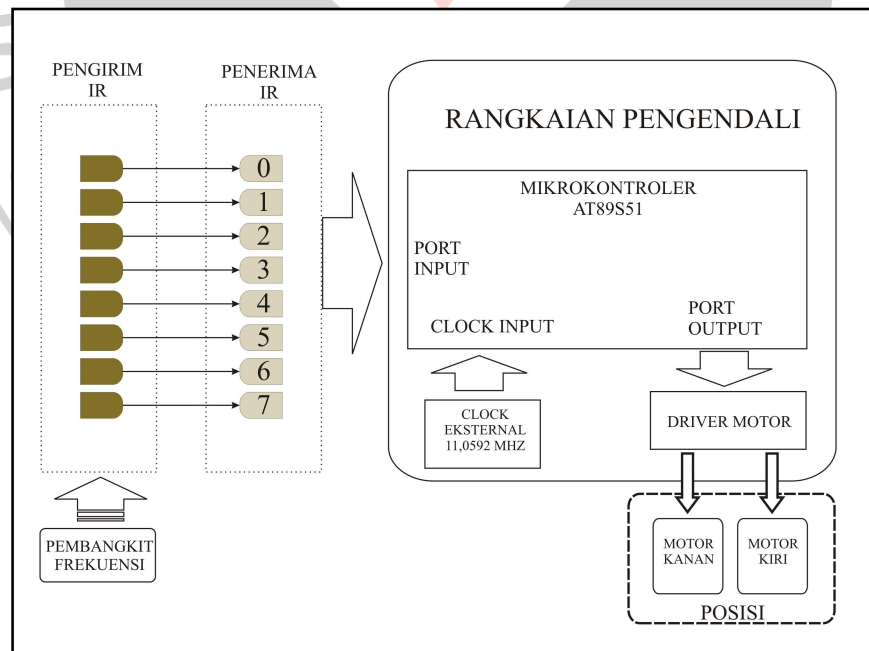
Gambar 3.1 Konfigurasi peletakan sensor pada Robot Avoider

3.1.1 Sistem Sensor Infra Merah

Secara umum sistem sensor infra merah yang dibuat digambarkan pada gambar

3.2. Cara kerja dari sistem tersebut dapat dijelaskan sebagai berikut :

1. Sensor dirancang dengan meletakkan bagian *transmitter* inframerah yang saling berseberangan dengan *receiver* inframerah. Transmitter inframerah terdiri dari rangkaian IC 555 yang berfungsi sebagai penghasil gelombang frekuensi 40 KHz.
2. Ketika cahaya inframerah terhalang oleh benda atau halangan, maka penerima akan mengirimkan sinyal ke mikrokontroler sebagai masukan sensor.
3. Mikrokontroler akan memproses sinyal yang masuk dari receiver inframerah untuk menggerakkan posisi motor.



Gambar 3.2 Diagram Blok sistem Infra merah

3.1.2 Rangkaian Pengirim Infra Merah

Rangkaian pengirim infra merah menggunakan pembangkit pulsa IC 555 dengan konfigurasi Astable multivibrator. Frekuensi sinyal yang digunakan adalah 40 KHz. Maksud dari frekuensi tersebut yaitu supaya pancarannya jauh dan tidak terpengaruhi *noise* dari luar. Frekuensi tersebut didapat dengan mengatur besar hambatan satu R_1 , hambatan dua R_2 , dan kapasitor satu C_1 . Besarnya R_1 , R_2 , dan C_1 yang digunakan untuk mendapatkan Frekuensi 40 KHz adalah (*National Semiconductor Corporation: 2000*) :

$$f = \frac{1.44}{(R_4 + 2R_1)C_1}$$

$$(R_4 + 2R_1)C_1 = 36 \times 10^{-6} \Omega F$$

Jika $R_1 = 10 \text{ K}\Omega$ dan $C_1 = 1 \text{ nF}$ maka

$$(R_4 + 2.10000 \Omega)1 \times 10^{-9} = 36 \times 10^{-6} \Omega F$$

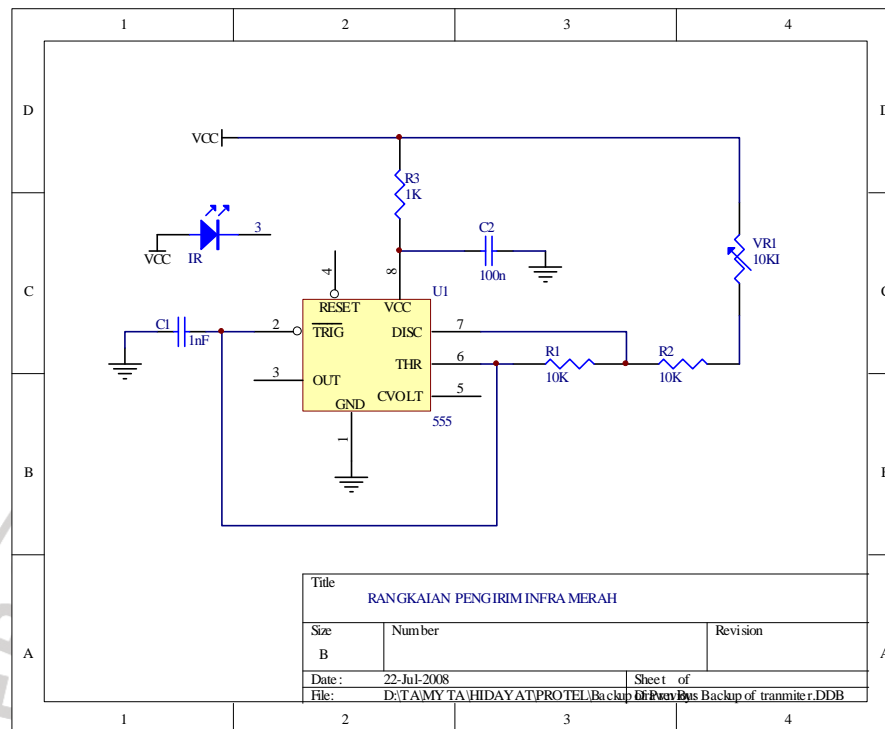
$$R_4 = \frac{36 \times 10^{-6} \Omega F - 2 \times 10^{-5} \Omega F}{1 \times 10^{-9} F}$$

$$R_4 = 16 \text{ K}\Omega \text{ dan}$$

$$VR_1 = R_4 - R_2 = 16 \text{ K}\Omega - 10 \text{ K}\Omega = 6 \text{ K}\Omega$$

Sehingga digunakan variabel resistor sebesar 10 K Ω .

Berikut adalah rangkaian pengirim infra merah :

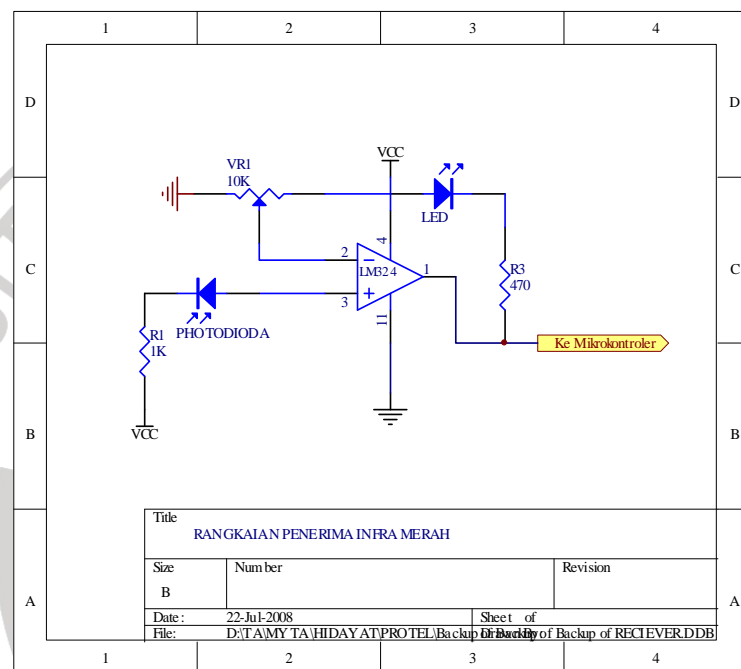


Gambar 3.3. Rangkaian Pengirim Infra Merah

3.1.3 Rangkaian Penerima Infra Merah

Pada Gambar 3.4 adalah rangkaian penerima infra merah yang dapat menangkap sinyal IR yang frekuensinya disetting antara 30 - 50 KHz, hal ini dilakukan supaya data yang dipakai dapat mengendalikan ada atau tidaknya pancaran sinar infra merah. Setelah diterima dalam bentuk pulsa kemudian akan dimasukkan ke komparator tegangan LM324, komparator ini sebagai pembanding antara tegangan reverensi dengan tegangan yang terukur pada potodioda. Proses pembacaan sensor dengan cara mendeteksi warna putih, maka output pada komparator berlogika '1'.

Fungsi VR1 pada sensor untuk mengeset tegangan yang digunakan sebagai referensi. Karena tiap tempat mempunyai pencahayaan yang berbeda-beda. Jadi ketika tegangan pada photodiode melebihi tegangan referensi maka output pada komparator akan berlogika '1'.



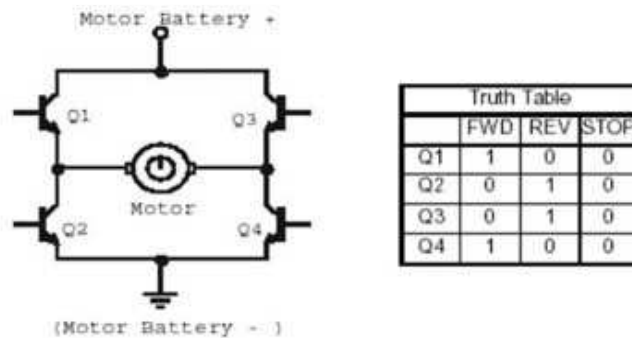
Gambar 3.4. Rangkaian penerima Infra Merah

3.1.4 Sistem Penggerak Motor

Driver motor yang digunakan yaitu IC L298 yang memiliki kemampuan menggerakkan motor DC sampai arus 2A dan tegangan maksimum 40 VoltDC untuk satu kanalnya. Driver motor ini memiliki dua jenis pin,yaitu Pin *Enable* untuk mengendalikan jalan atau kecepatan motor, dan pin Input untuk mengendalikan arah putaran. Pin Enable diberi VCC 5 Volt untuk kecepatan

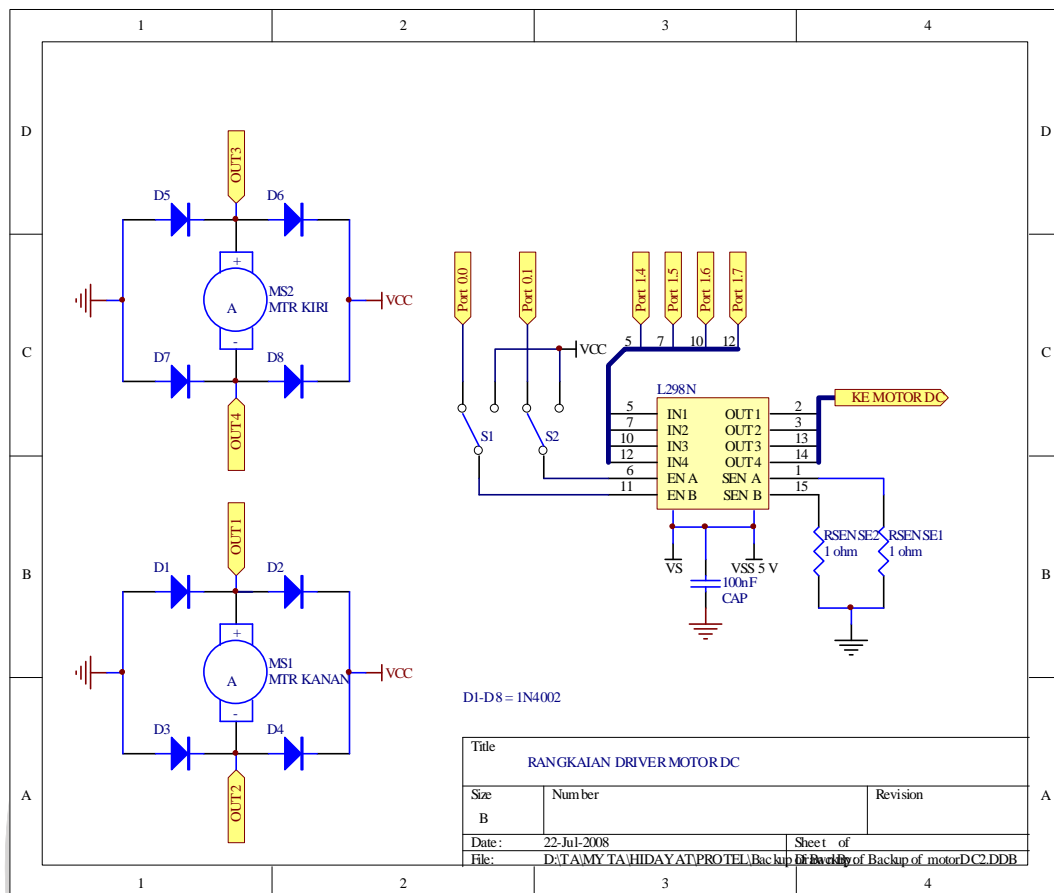
penuh dan PWM (*Pulse Width Modulation*) untuk kecepatan rotasi yang bervariasi tergantung dari level highnya.

Berikut ilustrasi pengendalian motor didalam IC L298 (ST Microelectronic: 2000):



Gambar 3.5. Ilustrasi Pengendalian Motor didalam IC L298

Didalam IC L298, untuk mengendalikan arah putaran motor digunakan metode bridge-H dari kombinasi transistor, jadi dengan metode demikian arus yang mengalir kemotor polaritasnya dapat diatur dengan memberikan logika ke transistor Q1 sampai Q4. Pengaturannya seperti tabel kebenaran disamping gambar 3.5. Kondisi high untuk semua input tidak diizinkan, sebab akan mengakibatkan semua transistor aktif dan akan merusakkan transistor karena secara otomatis arus dari kolektor Q1 dan Q2 langsung mengalir ke Q2 dan Q3 sehingga arus sangat besar tanpa melalui beban motor DC.



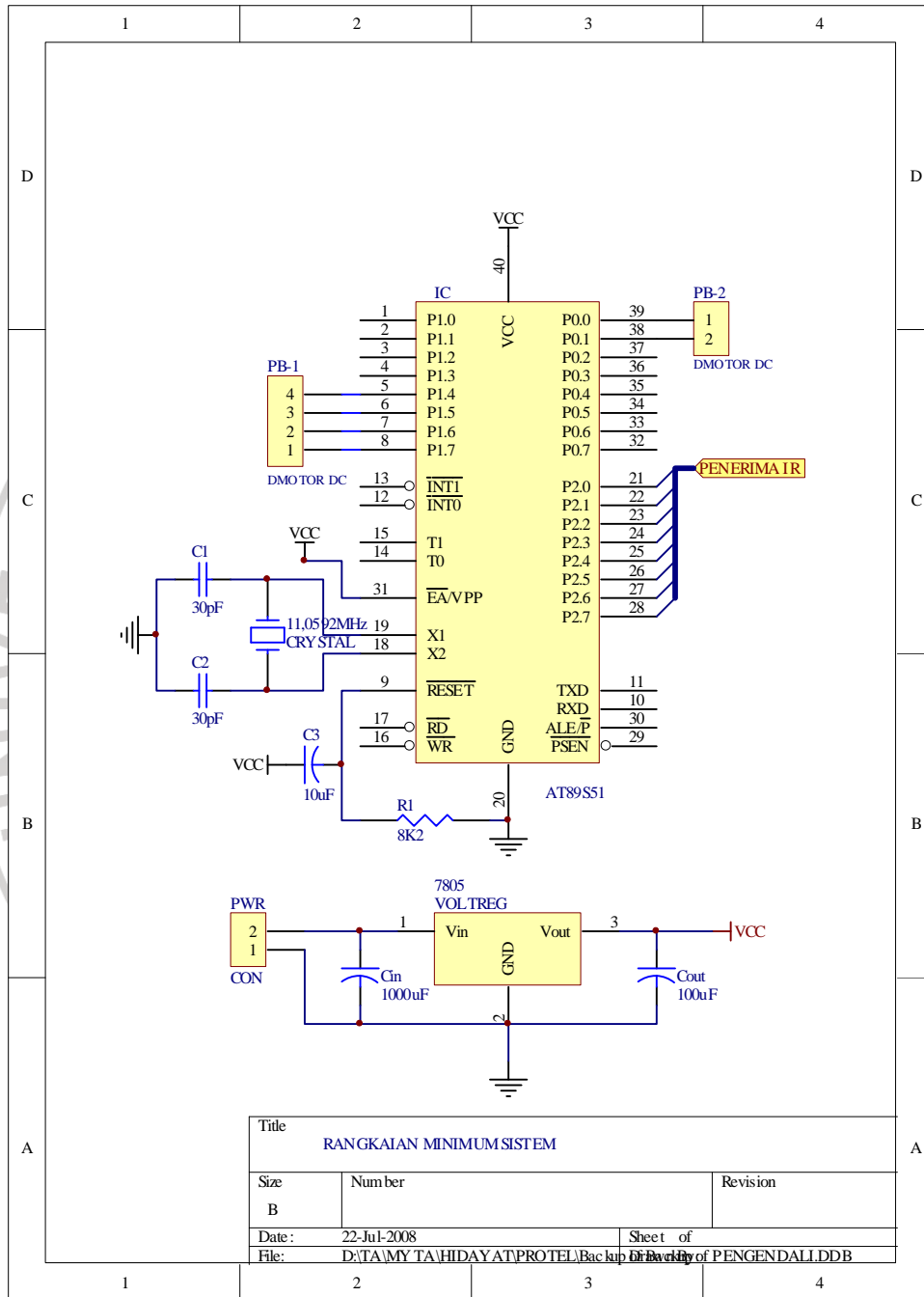
Gambar 3.6. Rangkaian Penggerak Motor DC

3.1.5 Rangkaian Pengendali

Pengendali sistem sensor dan sistem penggerak yang digunakan adalah sebuah mikrokontroler AT89S51 yang rangkaiannya ditunjukkan pada gambar 3.8. Mikrokontroler bekerja dengan frekuensi clock 11,0592 MHz yang diperoleh dari oscillator Kristal.

Program pengendali semuanya disimpan hanya pada memori flash yang ada didalam mikrokontroler dan tidak menggunakan tambahan memori eksternal. *Flash Memory* dapat diprogram sebesar 4Kbytes, 128 x 8 bit RAM internal. Jadi

dengan menggunakan mikro tipe ini akan didapat desain yang cukup kompak dan pemrogramannya relatif lebih mudah.

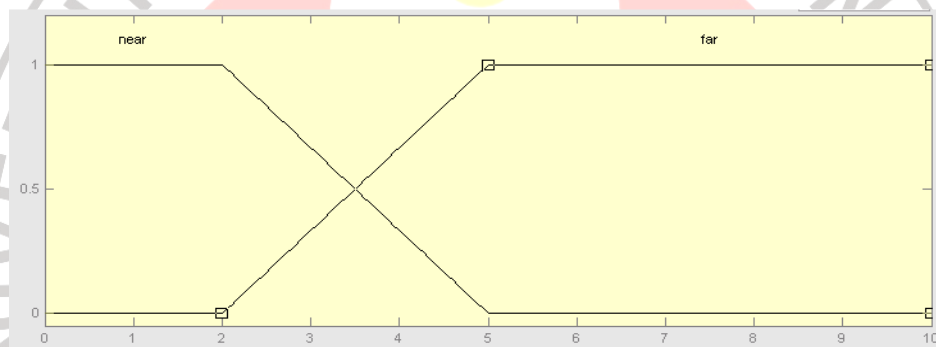


Gambar 3.7. Rangkaian Minimum Sistem AT89S51

3.2 Perancangan dan Pembuatan Perangkat Lunak (*Software*)

3.2.1 Sistem Kendali Fuzzy

Dalam system logika fuzzy, sebagai crisp inputnya adalah jarak antara robot mobil dan halangan. *Crisp* masukan ini dirancang dalam delapan arah sudut pandang dari robot avoider ini sesuai dengan konfigurasi peletakan sensor (gambar 3.1). Delapan arah tersebut adalah *Front*, *More Left*, *Left*, *Back Left*, *Back*, *Back Right*, *Right*, *More Right*. Sedangkan *Membership Function* (MF) untuk jarak robot ke halangan mempunyai dua label yaitu *Near* dan *Far*.

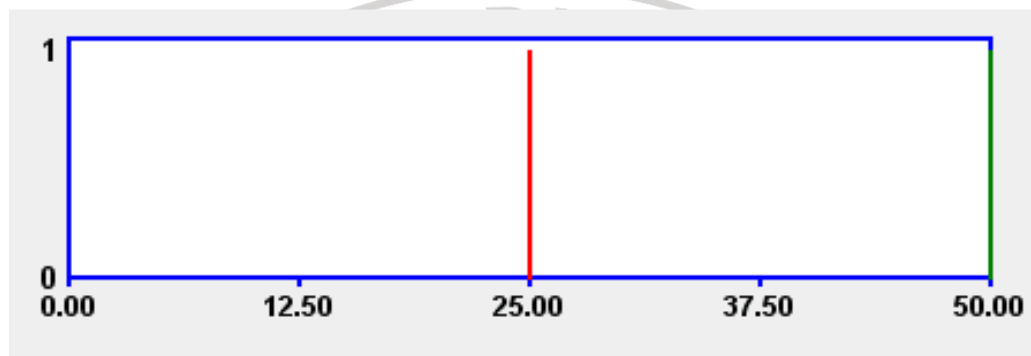


Gambar 3.8 MF Jarak robot – halangan

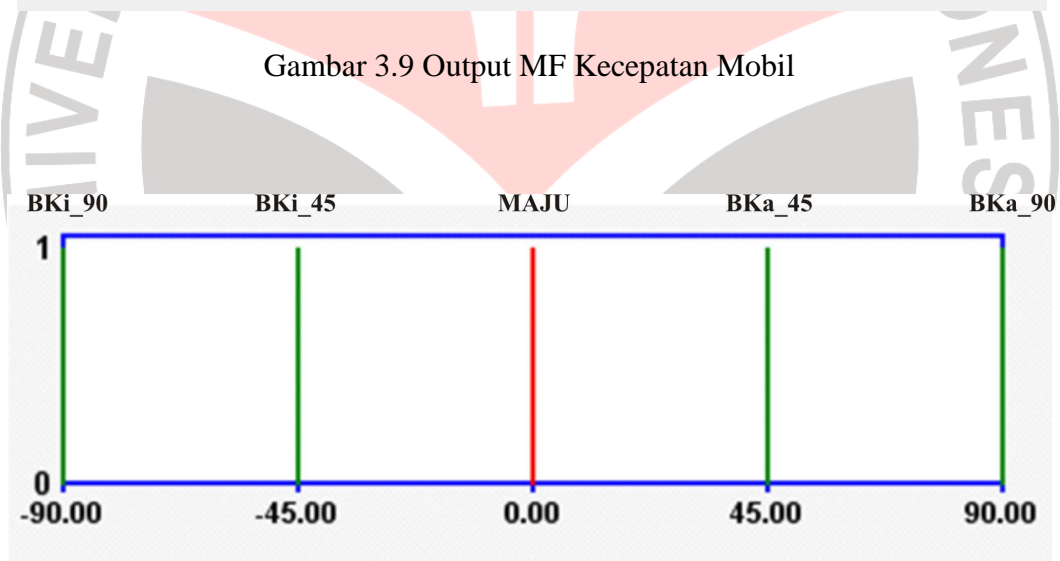
Crisp keluaran untuk sistem logika fuzzy pada penelitian ini yaitu posisi sudut kemudi dan kecepatan robot avoider. Output MF untuk posisi sudut kemudi mempunyai enam label yaitu Maju, Mundur, Belok Kanan 45° (BKa45), Belok Kanan 90° (BKa90), Belok Kiri 45° (BKl45), Belok Kiri 90° (BKl90). Sedang MF untuk kecepatan robot avoider hanya memiliki dua label, yaitu cepat dan lambat.

Dalam *evaluasi rule* terdapat aturan *linguistic* untuk menentukan aksi kontrol yang harus dilakukan dalam merespon nilai masukan yang diberikan. Kemudian dilakukan evaluasi hubungan atau derajat antecedent setiap aturan.

Langkah selanjutnya adalah pencarian derajat kebenaran untuk setiap rule. Saat antecedent dihubungkan dengan operator “AND”, rule strength mengasumsikan nilai terkuat dari antecedent rule. Nilai maksimum ini yang akan menjadi nilai kebenaran bagi rule tersebut. Metode ini dinamakan “MIN-MAX”, karena mengambil nilai maksimum untuk keluaran fuzzy.



Gambar 3.9 Output MF Kecepatan Mobil



Gambar 3.10 Output MF Posisi Sudut Kemudi

Dari tabel 3.1 dibawah ini, didapatkan didapatkan aturan-aturan linguistik yang mana aturan-aturan tersebut yang akan digunakan sebagai sistem dalam robot itu sendiri.

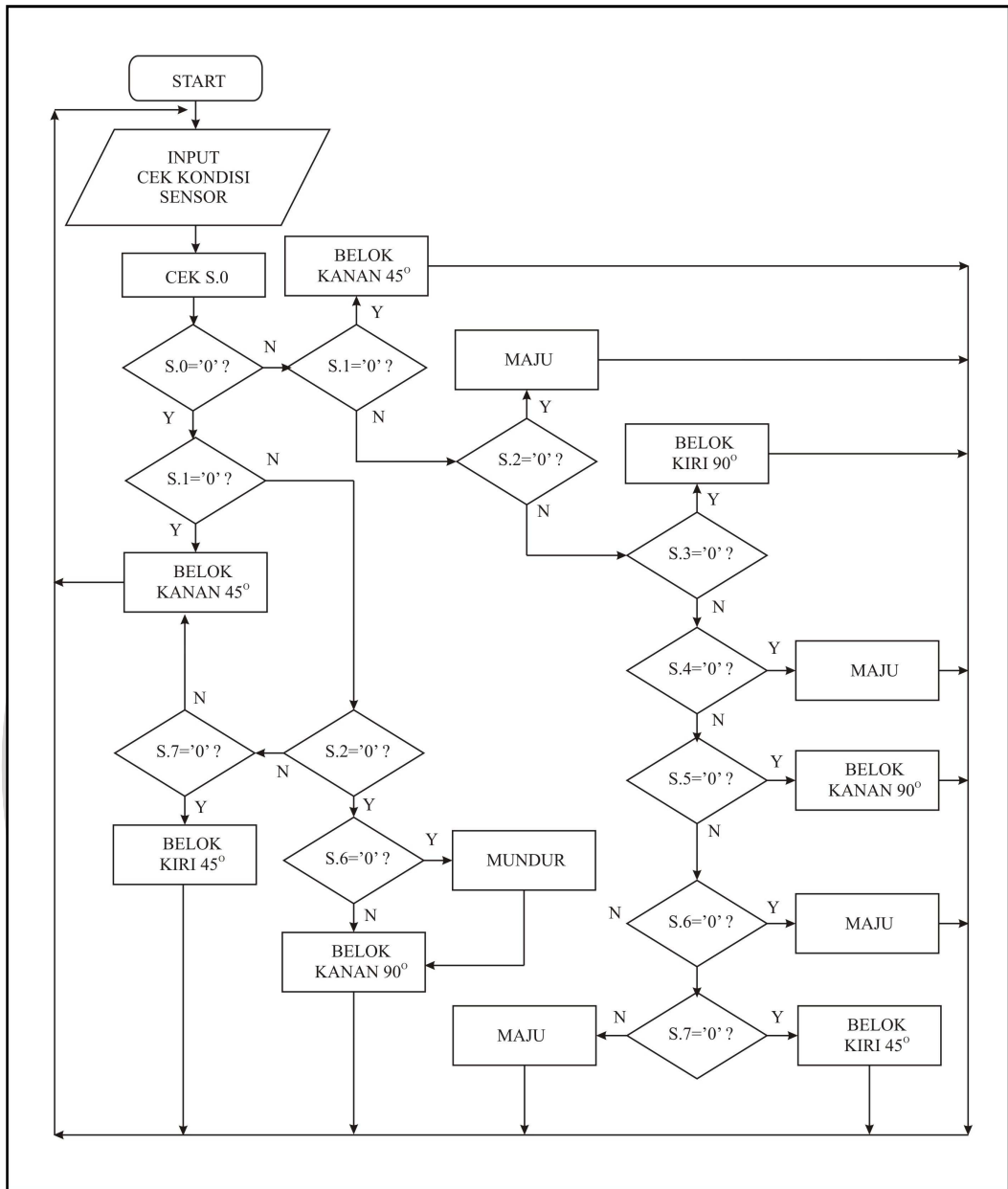
Tabel 3.1 Evaluasi Rule Fuzzy untuk Kendali Robot Avoider

CRISP INPUT								CRISP OUTPUT	
P2.0 (front)	P2.1 (more left)	P2.2 (left)	P2.3 (back left)	P2.4 (back)	P2.5 (back right)	P2.6 (right)	P2.7 (more right)	KMDI	KEC.
DEKAT	JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	BKa45	L
JAUH	DEKAT	JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	BKa45	L
JAUH	JAUH	DEKAT	JAUH	JAUH	JAUH	JAUH	JAUH	MA	L
JAUH	JAUH	JAUH	DEKAT	JAUH	JAUH	JAUH	JAUH	BKi90	C
JAUH	JAUH	JAUH	JAUH	DEKAT	JAUH	JAUH	JAUH	MA	L
JAUH	JAUH	JAUH	JAUH	JAUH	DEKAT	JAUH	JAUH	BKa90	C
JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	DEKAT	JAUH	MA	L
JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	DEKAT	BKi45	L
DEKAT	DEKAT	JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	BKa45	L
DEKAT	JAUH	DEKAT	JAUH	JAUH	JAUH	JAUH	JAUH	BKa90	C
DEKAT	JAUH	DEKAT	JAUH	JAUH	JAUH	DEKAT	JAUH	MU	C
DEKAT	JAUH	JAUH	JAUH	JAUH	JAUH	JAUH	DEKAT	BKi45	L
DEKAT	JAUH	JAUH	JAUH	JAUH	JAUH	DEKAT	JAUH	BKi90	C

3.2.2 Perangkat Lunak Pengendali

Perangkat lunak ditujukan untuk mengendalikan sistem, yaitu berupa program yang nantinya disimpan di dalam chip mikrokontroler, sehingga mikrokontroler dapat melaksanakan perintah secara otomatis sesuai dengan urutan programnya. Program yang dibuat dengan menggunakan bahasa assembler, di mana set instruksinya sesuai dengan set instruksi MCS-51.

Gambar dibawah ini, memperlihatkan algoritma program yang dibuat. Sebenarnya kondisi sensor jika dikenai halangan akan bernilai high (logika “1”), tetapi sebelum diproses dimikrokontroler, terlebih dulu masuk melalui *not gate* 74HC04, sehingga yang akan diproses dalam mikrokontroler adalah sensor yang bernilai low (logika “0”). Maka dari itu, jika kondisi sensor berlogika “0” berarti robot berhadapan atau mendeteksi halangan disekitarnya, maka robot avoider akan bergerak baik belok kanan, belok kiri ataupun mundur.



Gambar 3.11 Algoritma Sistem Robot AVOIDER

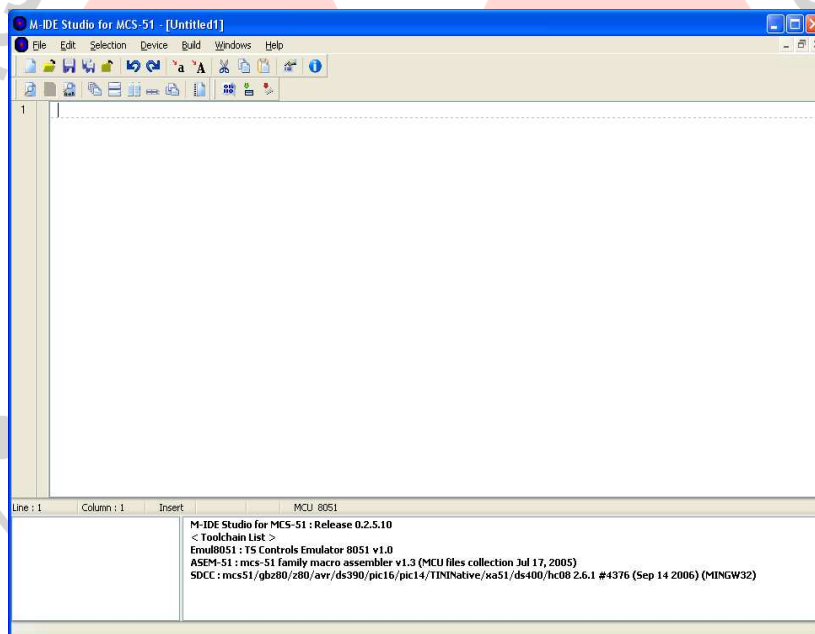
3.3 Teknis Kompilasi Program dan Penyimpanan Program ke Mikrokontroler

3.3.1 Teknis Kompilasi Program

Program dibuat menggunakan Mide-51 dengan bahasa pemrograman assembly. Mide-51 menyediakan fasilitas editor untuk pembuatan program dan terdapat dua bahasa pemrograman, yakni bahasa assembly dan bahasa c.

Teknis pembuatan dan kompilasi program dengan Mide-51 sebagai berikut :

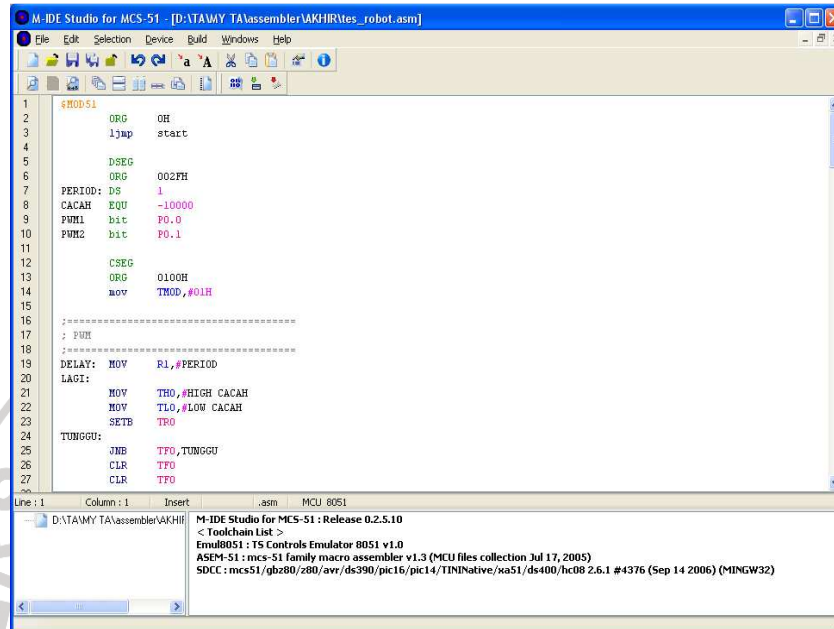
1. Untuk memulai klik Start>Mide-51 maka yang tampil dilayar monitor seperti pada gambar berikut :



Gambar 3.12 Tampilan Utama Mide-51

2. Kemudian untuk memulai penulisan program, klik File>New maka akan muncul dokumen kosong yang siap untuk ditulis. Penulisan program bisa menggunakan bahasa assembly ataupun dengan bahasa c. Pada tugas akhir

ini, program yang dibuat menggunakan bahasa assembly. Seperti pada gambar dibawah ini :



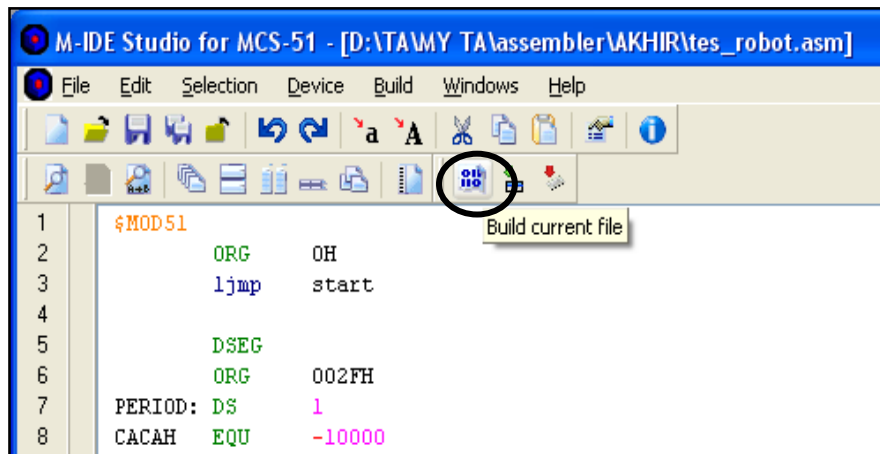
```
1 $M051
2   ORG   0H
3   ljmp  start
4
5   DSEG
6   ORG   002FH
7   PERIOD: DS 1
8   CACAH EQU -10000
9   PMM1  bit  P0.0
10  PMM2  bit  P0.1
11
12  CSEG
13  ORG   0100H
14  mov  TH0,#01H
15
16  ;=====
17  ; PUL
18  ;=====
19  DELAY: MOV  R1,#PERIOD
20
21  LAGI:
22        MOV  TH0,#HIGH CACAH
23        MOV  TLO,#LOW CACAH
24        SETB TH0
25
26  TUNGGU: JNB  TFO,TUNGGU
27         CLR  TFO
28         CLR  TFO
29
```

Line : 1 Column : 1 Insert .asm MCU: 8051

M-IDE Studio for MCS-51 : Release 0.2.5.10
< Toolchain List >
Emul8051 : TS Controls Emulator 8051 v1.0
ASEM-51 : mcs-51 family macro assembler v1.3 (MCU files collection Jul 17, 2005)
SDCC : mcs51/gbz80/z80/avr/ds390/pic16/pic14/TINNative/xa51/ds400/hc08 2.6.1 #4376 (Sep 14 2006) (MINGW32)

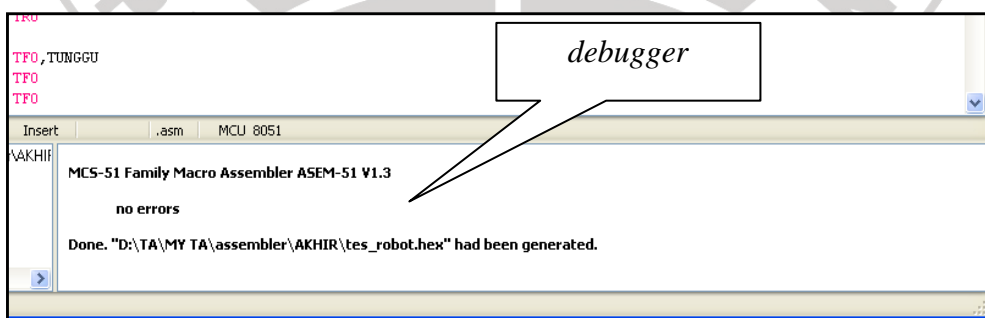
Gambar 3.13 pembuatan program dengan bahasa assembly

3. Setelah pembuatan program selesai, kemudian file disimpan terlebih dahulu dengan cara klik File>Save As, maka akan muncul kotak dialog Save As. File yang akan disimpan harus menggunakan ekstensi .asm untuk bahasa assembly dan ekstensi .c untuk bahasa c, contoh : tes_robot.asm, robot.c.
4. Setelah file disimpan, maka langkah selanjutnya yaitu mengkompilasi file dengan ekstensi .asm atau ekstensi .c ke ekstensi .hex. karena yang digunakan untuk penyimpanan program ke mikrokontroler adalah file yang berekstensi .hex. oleh karena itu, Mide-51 mempunyai fasilitas untuk mengkompilasi file yang berekstensi .asm atau .c menjadi ekstensi .hex dengan cara klik tombol *Built Current File* (lingkaran hitam pada gambar) atau klik Build>Build atau tekan F9 pada keyboard.



Gambar 3.14 Tombol untuk Mengkompilasi Program

- Setelah melakukan kompilasi, Mide-51 akan memberikan informasi jika terdapat kesalahan ataupun tidak terdapat kesalahan pada program yang dibuat, informasi tersebut dapat dilihat pada kotak *debugger*. *Debugger* akan memberikan informasi sesuai dengan baris mana yang terjadi error (kesalahan) dan jika program tersebut sudah benar, maka pada kotak debugger akan muncul tulisan “no errors” dan file tersebut akan berubah menjadi file yang ber-ekstensi .hex yang tersimpan satu folder dengan file yang berekstensi .asm atau .c.



Gambar 3.15 Kotak Debugger pada Mide-51

3.3.2 Teknis Penyimpanan Program ke Mikrokontroler

Program yang telah dibuat disimpan pada mikrokontroler dengan menggunakan perangkat lunak *Atmel 89 Series Flash Programmer versi 3.1* buatan Muhammad asim khan. Perangkat lunak ini dipakai untuk mengisikan program ke mikrokontroler keluarga MCS-51.

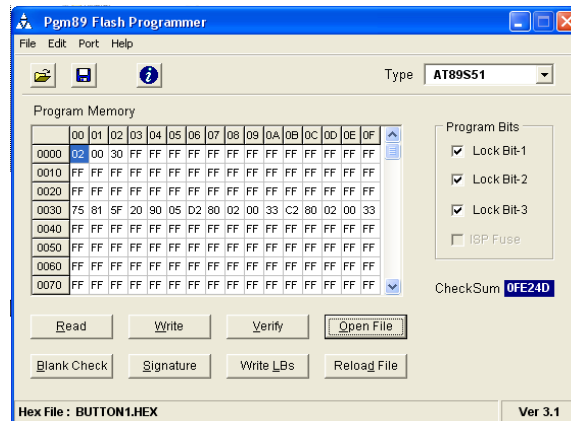
Berikut ini langkah – langkah penyimpanan program pada mikrokontroler dengan menggunakan *Atmel 89 Series Flash Programmer* :

1. Jalankan software *Atmel 89 Series Flash Programmer* dengan mengklik dua kali *pgm89v31.exe* (untuk *Windows 98*) atau dengan mengklik *pgm89xp.bat* (untuk *Windows XP*). Maka akan muncul tampilan awal sebagai berikut :



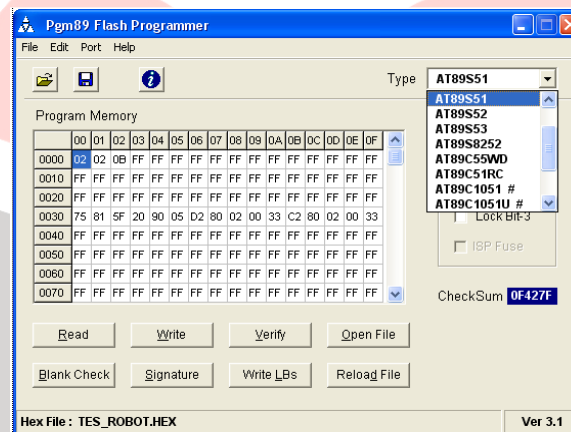
Gambar 3.16 Tampilan awal *Atmel 89 Series Flash Programmer*

Setelah tampilan awal muncul kemudian akan muncul tampilan utama *Atmel 89 Series Flash Programmer*.



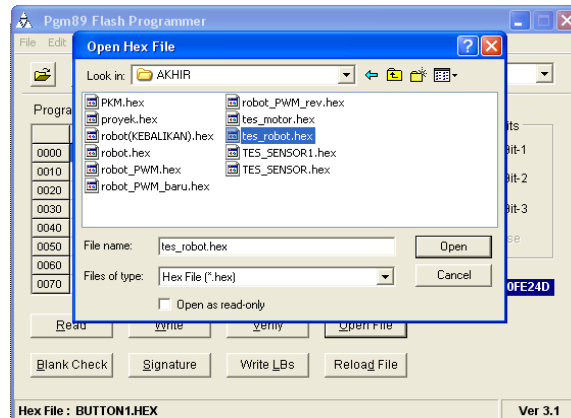
Gambar 3.17 Tampilan utama *Atmel 89 Series Flash Programmer*

- Pilih type IC mikrokontroler yang digunakan melalui combo box **Type**.

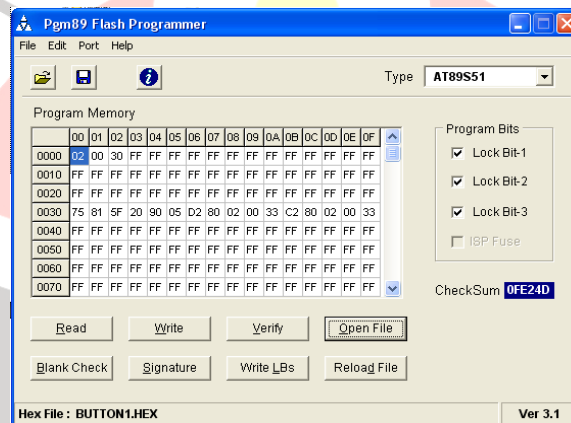


Gambar 3.18 Combo box **Type** : untuk memilih tipe mikrokontroler

- Buka file HEX yang akan dituliskan pada IC dengan menggunakan menu (**File>Open Hex File**), atau tool button **Open File** atau tombol **Open File**. Isi dari file yang dibuka akan ditampilkan pada buffer.



Gambar 3.19 Kotak dialog open hex file



Gambar 3.20 Buffer yang berisi program

4. Programkan kode program yang ditampilkan ke dalam IC dengan menekan tombol **Write**. Perhatikan, program mulai menuliskan kode program kedalam IC yang ditunjukkan oleh menyalnya LED merah pada *downloader*. Program akan sekaligus memverifikasi kode program yang telah dituliskan dengan program aslinya (yang ada di buffer).
5. Untuk lebih meyakinkan kebenaran pengisian program, tekan tombol **Verifikasi** lagi.