BAB IV

PENGUJIAN ALAT DAN ANALISA

Dari perancangan yang dilakukan pada bab III sebelumnya maka didapatkan hasil perancangan sebagai berikut :

4.1. Pengujian Alat

Berikut ini adalah urut-urutan untuk menjalankan emulator AT89S51 :

 Setelah alat dihubungkan ke power suplai dan dihubungkan ke port serial komputer maka aplikasi Delphi 7 dijalankan (dieksekusi) sehingga akan muncul tampilan berikut ini :



Gambar 4. 1. Tampilan awal emulator mikrokontroler AT89S51

- ាក Settings DOM1. ¥ Port 9600 • Baud rate 8 Ŧ Data bits 1 Stop bits None Parity None Flow control ΟΚ Cancel Gambar 4. 2. Tampilan SetCom
- 3. Setelah di OK maka dapat dipilih mode kerja dari *emulator*, apakah akan men-*download file* hex ke RAM atau melihat isi RAM. Jika ditekan tombol men-*download file* hex ke RAM maka akan muncul tampilan sebagai berikut :



Gambar 4. 3. Tampilan indikator mikrokontroler sudah terhubung ke computer

2. Kemudian untuk mengaktifkan komunikasi serial tekan tombol SetCom dengan tampilan sebagai berikut :

Jika pada kolom RX tidak diterima data V berarti port serial komputer belum terhubung ke mikrokontroler.

4. Setelah data V diterima maka tekan tombol open file sehingga akan muncul tampilan seperti Gambar 4.4. berikut :

Open	are as the second	and a start of the	an ison an abar	i Zali Vali 2. aniety	?X
Look in:) Delphí Aan		<u>-</u>]		-
My Recent Documents Desktop	L ALEX1 L BOSCO L COBA L EMULATOR L EMULATOR		Z		
My Documents					
My Computer	File name		ž	;	
Places	Files of Lype. File File Files	Hex (*.hex)			Cancel

Gambar 4. 4. Tampilan open file

5. Setelah nama filenya dipilih maka aplikasi akan meminta pemasukan 4 bit

alamat awal seperti Gambar 4.5 berikut

KA F		
Masukk	an alamat awai HU	M dalam 4 hex char
0000		

Gambar 4. 5. Tampilan pemasukan alamat awal

6. Setelah memasukkan alamat awal maka proses selanjutnya adalah menulisakan data ke RAM seperti pada pembahasan berikut

4.2. Mode penyimpanan data ke RAM

Tampilan Delphi saat penyimpanan data ke RAM ditunjukkan pada gambar 4.6 berikut :

7' Emulator EPROM Dow						
Form Utama – Pengisian Memori – Melihat isi Memori –						
Download file Hexa ke RAM						
Tekan BK Unluk kembali						
Rx 0 Open files Send						
Data yang terkirim						
:100000080402010080402010102040810204080F2: 00100080402010080402010102040810204080D2:1000 200080402010080402010102040810204080D2:1000 00FF7F3F1F0F07030100000103070F1F3F52:10004(07FFFFF7F3F1F0F07030100000103070F22:100050) 1F3F7FFF80402010080402010102040886:10006000 020408080402010080402010102040882:1000700010 0408000182442810081422418000092:100080001824 281008142241800FFE7DBBD7EFF77:100090007EE DBE7FFFE7DBBD7EFF7EBDDBE7FF6D:0100A0 0520D:00000001FF						



Contoh aplikasi yang digunakan pada pengujian emulator EPROM ini adalah aplikasi lampu led. File hexa yang akan disimpan ke RAM adalah sebagai berikut :

- 1. :100000080402010080402010102040810204080F2
- 2. :1000100080402010080402010102040810204080E2
- 3. :1000200080402010080402010102040810204080D2
- 4. :10003000FF7F3F1F0F07030100000103070F1F3F52
- 5. :100040007**FFFF7F3F1F0F07030100000103070F**22
- 6. :100050001F3F7FF804020100804020101020408B6
- 7. :100060001**0204080804020100804020101020408**92
- 8. :100070001020408000182442810081422418000092
- 9. :1000800018244281008142241800FFE7DBBD7EFF77
- 10. :100090007EBDDBE7FFFFF7DBBD7EFF7EBDDBE7FF6D
- 11. :0100A000**52**0D
- 12. :0000001FF

Dari data di atas dapat dilihat bahwa setiap baris data hexa selalu diawali dengan data ':' dan diakhiri dengan *checksum*. Semua data yang dicetak tebal adalah data. *Checksum* merupakan kebalikan dari hasil penjumlahan data setelah data ':' sebagai contoh pada baris 1 dapat data checksumnya adalah F2. Data F2 tersebut didapat dengan perhitungan sebagai berikut :

```
10+00+00+00+80+40+20+10+08+04+02+01+01+02+04+08+10+20+40+80=20 E
```

Data 20E dibalik (*inverts*) menjadi FFFFDF1, kemudian diambil 2 *byte* terkecil ditambah 1 sehingga menjadi F1 +1 = F2.

4.3. Mode Pemakaian Memori (RAM)

Tampilan Delphi saat pemakaian data RAM ditunjukkan pada gambar 4.7 berikut :





Jika data yang disimpan pada RAM seperti pada tampilan gambar 4.7 maka saat pemakaian maka led akan menyala sesuai dengan data yang dibaca dari RAM sebagai contoh misalnya diterima data seperti pada baris 1 yaitu : 80402010080402010102040810204080 maka setiap pembacaan 1 byte (2 digit) akan terjadi perubahan nyala pada 8 buah LED.

data 80 ===================================	data 01 = 0000000
data 40 =○●○○○○○	data $02 = \bigcirc $
data 20 =0000000	data 04 = 000000000
data 10 =00000000	data 08 = 00000000
data 08 =0000●000	data 10 = ○○○●○○○○
data 04 =000000000	data 20 = $\bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc$
data 02 =0000000	data 40 = ○●○○○○○
data 01 =0000000	data 80 = •0000000

Gambar 4. 8. Tampilan LED berdasarkan data yang diterima

