

BAB IV

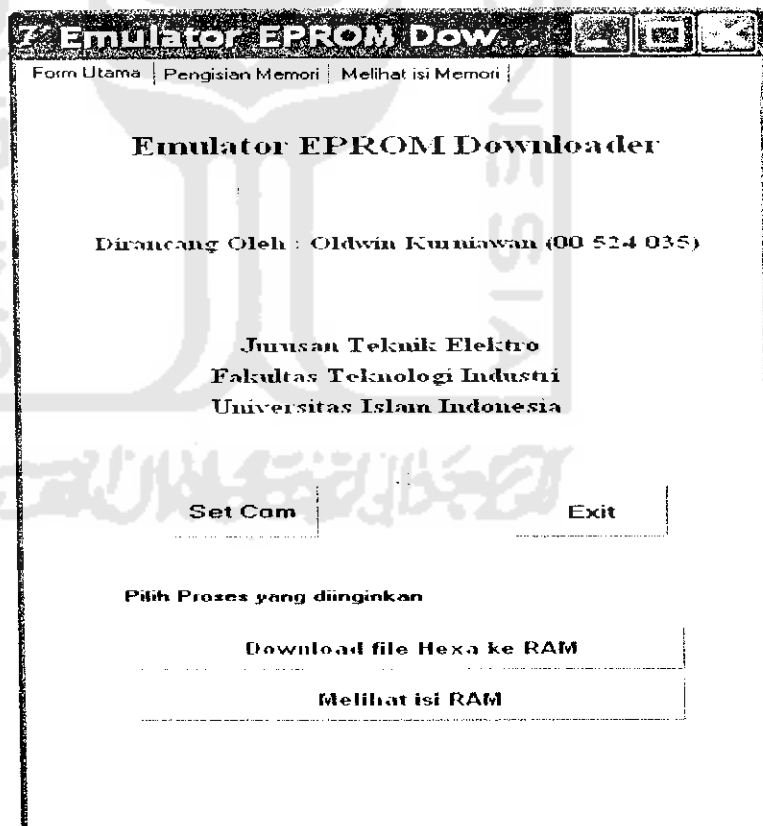
PENGUJIAN ALAT DAN ANALISA

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

4.1. Pengujian Alat

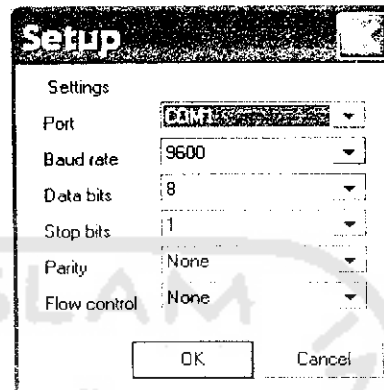
Berikut ini adalah urutan untuk menjalankan emulator AT89S51 :

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



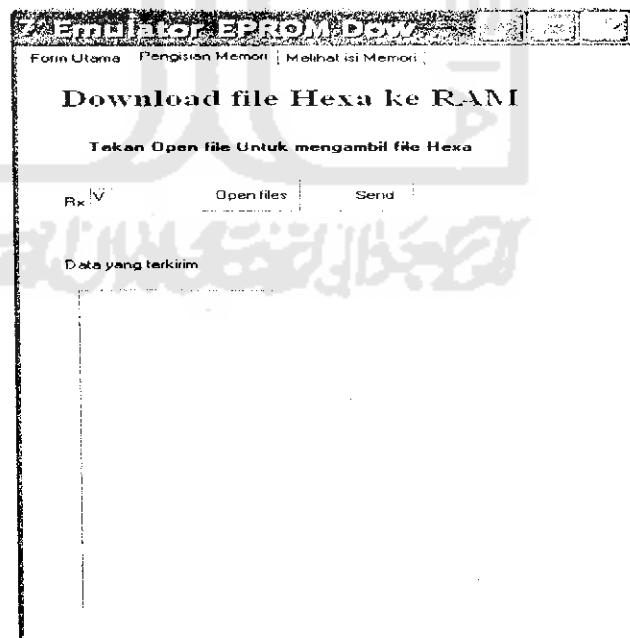
Gambar 4. 1. Tampilan awal *emulator* mikrokontroler AT89S51

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



Gambar 4. 2. Tampilan SetCom

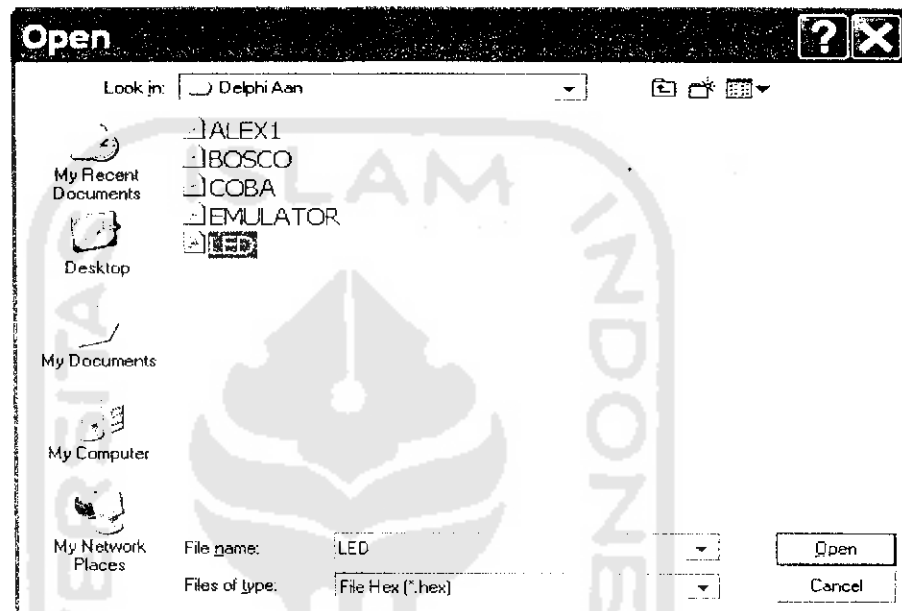
3. Setelah di OK maka dapat dipilih mode kerja dari *emulator*, apakah akan men-*download file hex* ke RAM atau melihat isi RAM. Jika ditekan tombol men-*download file hex* ke RAM maka akan muncul tampilan sebagai berikut :



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

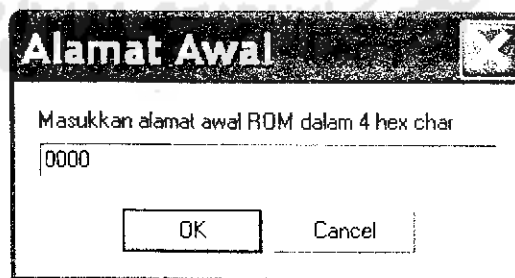
Jika pada kolom RX tidak diterima data V berarti port serial komputer belum terhubung ke mikrokontroler.

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



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

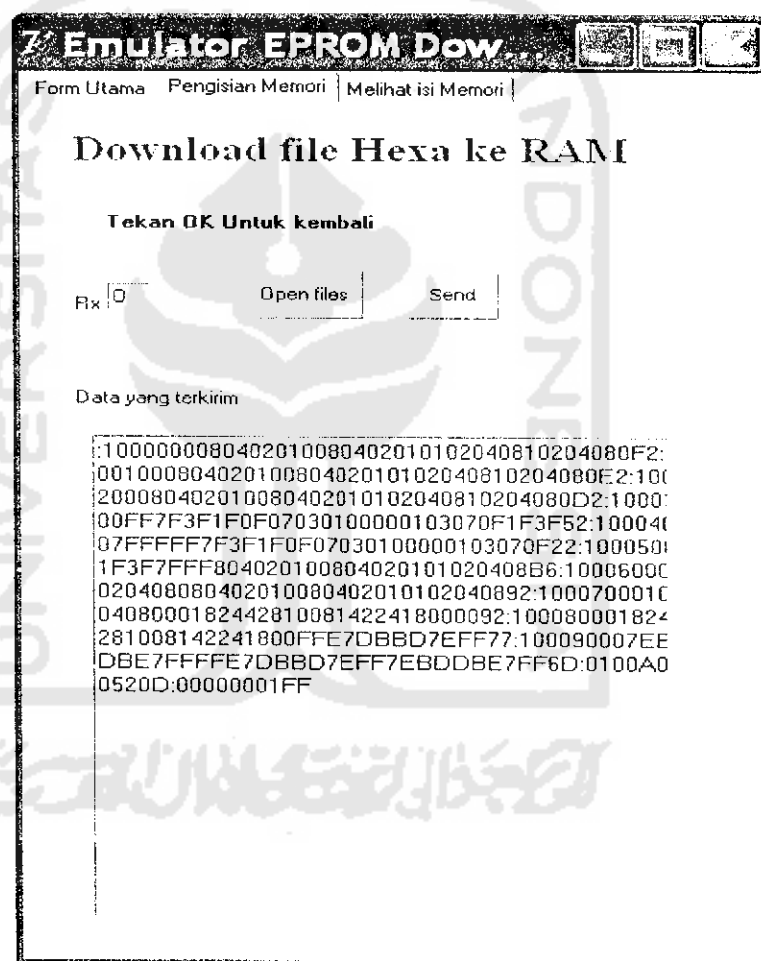


Gambar 4. 5. Tampilan pemasukan alamat awal

- Setelah memasukkan alamat awal maka proses selanjutnya adalah menuliskan 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 :



Gambar 4. 6. Tampilan Delphi saat penyimpanan data ke RAM

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

1. :10000000**80402010080402010102040810204080F2**
2. :10001000**80402010080402010102040810204080E2**
3. :10002000**80402010080402010102040810204080D2**
4. :10003000**FF7F3F1F0F07030100000103070F1F3F52**
5. :10004000**7FFFFFFF7F3F1F0F07030100000103070F22**
6. :10005000**1F3F7FFF804020100804020101020408B6**
7. :10006000**1020408080402010080402010102040892**
8. :10007000**1020408000182442810081422418000092**
9. :10008000**18244281008142241800FFE7DBBD7EFF77**
10. :10009000**7EBDDBE7FFFFFFE7DBBD7EFF7EBDDBE7FF6D**
11. :0100A000520D
12. :00000001FF

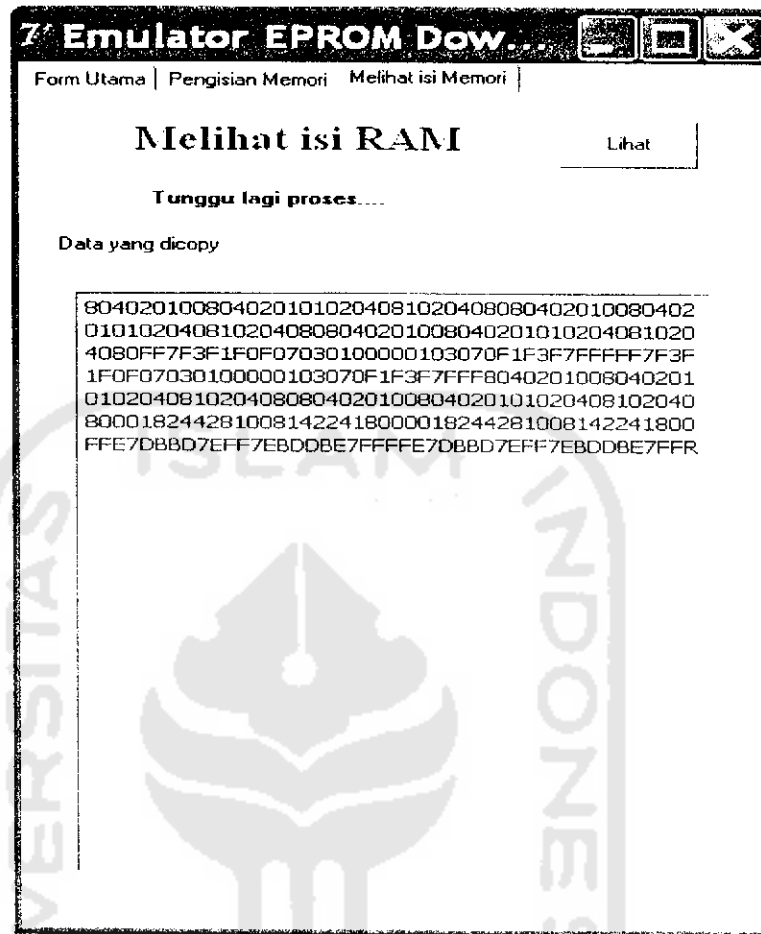
Dari data di atas dapat dilihat bahwa setiap baris data hexa selalu diawali dengan data ':' dan diakhiri dengan *checksum*. Semua data yang dicetak tebal adalah data. *Checksum* merupakan kebalikan dari hasil penjumlahan data setelah data ':' sebagai contoh pada baris 1 dapat data checksumnya adalah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

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

Data 20E dibalik (*inverts*) menjadi FFFFD1, 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 :



Gambar 4. 7. Tampilan Delphi saat pemakaian data RAM

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 = ○○○○○○○○●
data 40 = ○●○○○○○○○	data 02 = ○○○○○○●○
data 20 = ○○●○○○○○	data 04 = ○○○○○●○○
data 10 = ○○○●○○○○	data 08 = ○○○○●○○○
data 08 = ○○○○●○○○	data 10 = ○○○●○○○○
data 04 = ○○○○○●○○	data 20 = ○○●○○○○○
data 02 = ○○○○○○●○	data 40 = ○●○○○○○○
data 01 = ○○○○○○○○●	data 80 = ●○○○○○○○

Gambar 4. 8. Tampilan LED berdasarkan data yang diterima

