LAB SHEET 1

PENGENALAN Z80 SIMULATOR IDE OSHONSOFT

A. TUJUAN

1. Dapat mengoperasikan program simulator Z80 IDE Oshonsoft dengan baik.
2. Dapat melihat/mengecek isi memori pada lintas data untuk setiap program.

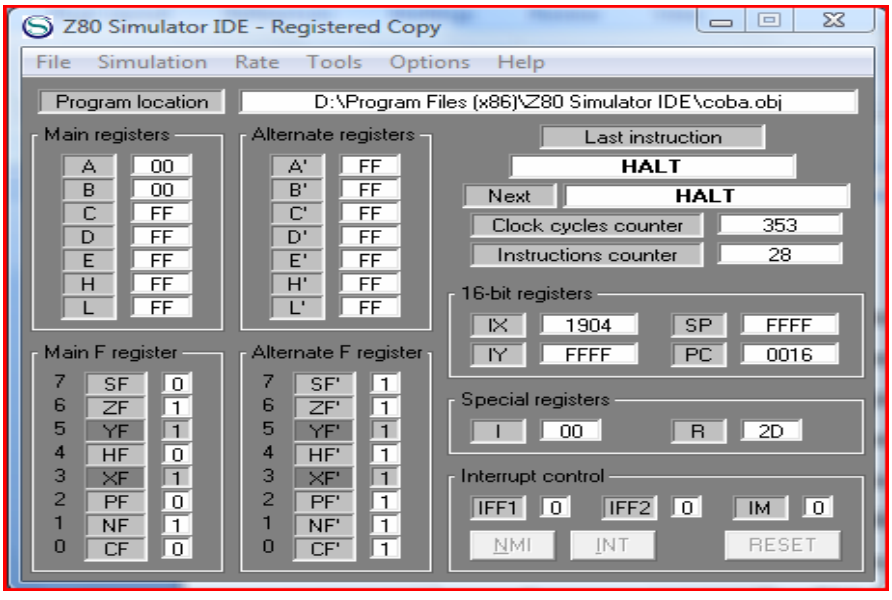
B. BAHAN DAN ALAT

1. Lembar tugas
2. Sofware Z80 Simulator IDE Oshonsoft

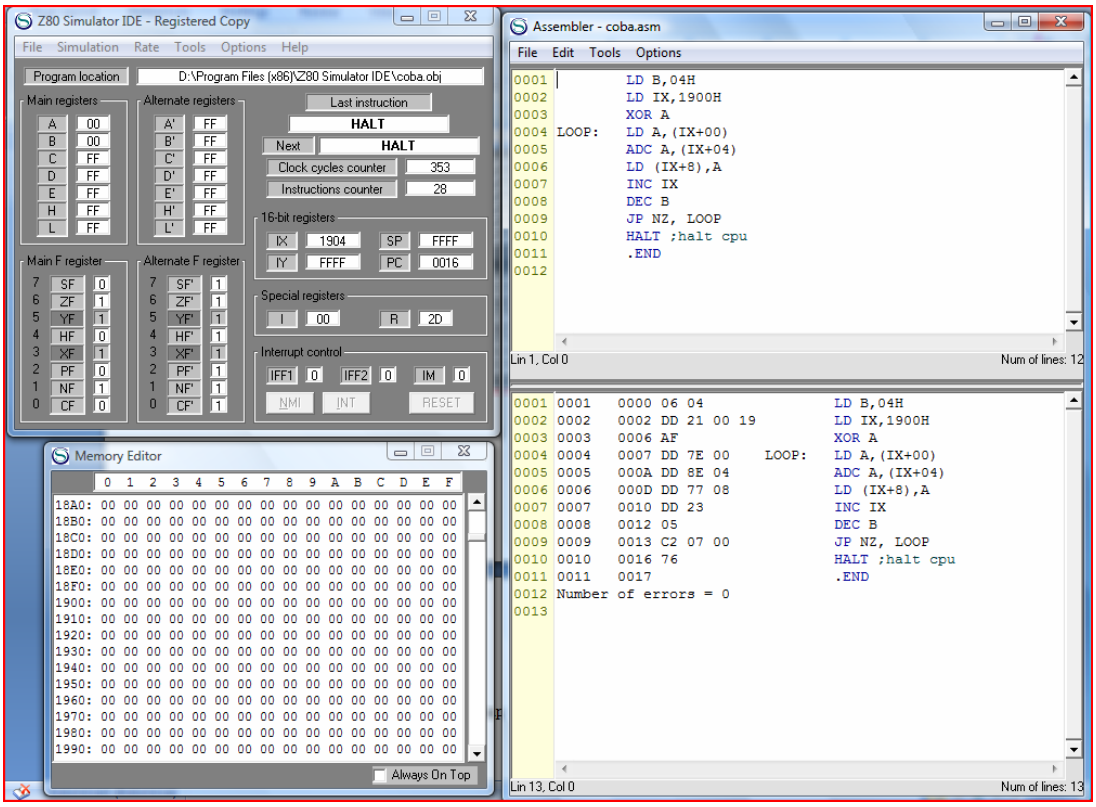
C. TEORI DASAR

Z80 Simulator IDE adalah applikasi simulator mikroprosesor Z80 yang memudahkan mahasiswa mempelajari mikroprosesor Z80. Simulator ini berbasis grafis yang terintegrasi dengan kompiler BASIC dan Assembler serta dilengkapi fasilitas debugger dan disassembler untuk mikroprosesor Z80. Aplikasi simulator ini mampu menunjukkan isi dari register internal termasuk status flagnya, mnemonic yang telah dieksekusi dan yang akan dieksekusi, nilai clock, instruction counter dan interupt interface.

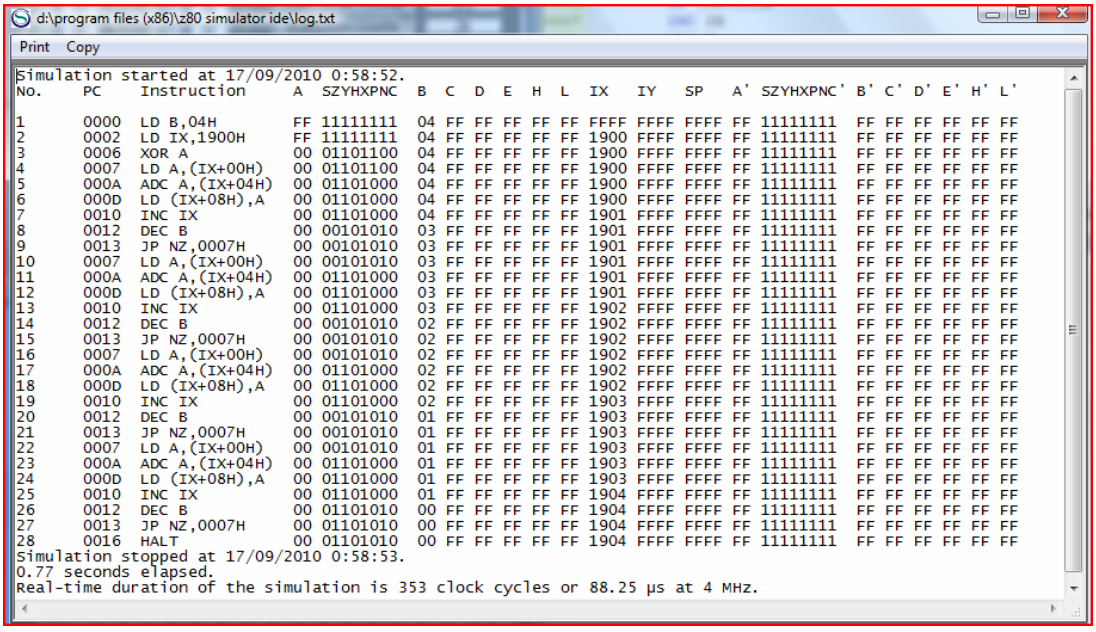
Secara umum tampilan Z80 Simulator IDE dapat dilihat dalam gambar berikut :



Sedangkan tampilan lengkap dengan memory editor dan assembler sebagai berikut :



Hasil eksekusi juga bisa dilihat dalam resume dalam file berekstensi .txt yang akan menjelaskan urutan proses dan perubahan nilai register yang terlibat dalam suatu program aplikasi Z80 yang telah dieksekusi.



Penjelasan masing-masing fungsi tombol menu Z80 Simulator IDE dapat dijelaskan sebagai berikut :

- Menu File
  - Clear Memory ; memerintahkan simulator untuk mengembalikan isi memori 64K dari alamat 0000H-FFFFH ke nilai awal (00H).
  - Load Program ; memuat program ke memori CPU. Program file harus dalam format HEX atau binary image (OBJ extension) yang termuat dalam memori dimulai dari alamat 0000H sampai alamat maksimal 64K.
  - Save Memory ; perintah ini untuk menyimpan isi dari memori ke sebuah file.
- Menu Simulation
  - Start ; membuat Z80 Simulator IDE dalam mode simulasi dan memulai mengeksekusi perintah yang dimulai dari alamat 0000H atau alamat awal program lain yang bisa disesuaikan dengan cara mengubah isi perintah Change Starting Address dari menu Option.
  - Step ; Perintah ini dapat digunakan jika rate simulasi dipilih dengan mode Step By Step. Instruksi berikutnya akan dieksekusi tiap penekanan tombol F2 pada keyboard.
  - Stop ; mengakhiri mode simulasi dan menyajikan informasi tentang jumlah total instruksi yang telah dieksekusi dan lama simulasi.
- Menu Rate
  - Memungkinkan pengguna untuk mengubah laju simulasi. Ada beberapa pilihan yaitu : Step By Step, Slow, Normal, Fast, Extremely Fast, dan Ultimate.
- Menu Tools
  - Memory Editor ; untuk mengakses interface grafis ke 64K memori. Untuk mengganti isi memori dilakukan dengan mengklik lokasi memori yang diinginkan kemudian memasukkan nilai bilangan hexadesimal dan diakhiri dengan menekan tombol ENTER pada keyboard, kursor akan otomatis pindah ke lokasi memori satu alamat berikutnya.

- Disassembler ; digunakan untuk memunculkan perintah mnemonic Z80 dari suatu file berektensi HEX atau OBJ. Hasil list perintah berupa file berekstensi LST.
- Peripheral Devices ; digunakan untuk memonitor perintah IN dan OUT. Jumlah I/O yang bisa diatur sampai 4 buah dan satu terminal output untuk melihat karakter ASCII yang dikirimkan ke salah satu port.
- I/O Port Editor ; untuk mengatur isi dari port I/O. Nilai dari port I/O dapat diganti dengan mengklik port yang diinginkan dan setelah nilai dimasukkan diakhiri dengan menekan tombol ENTER keyboard.
- External Modules ; digunakan untuk memantapkan interface otomasi sampai lima modul eksternal client/server,
- Assembler ; file program assembler Z80 dapat langsung diketik, dikompile dan dimuat ke memori simulator dalam satu lingkungan tampilan grafis. Ekstensi file berupa .ASM, ketika proses assembly berhasil dilakukan akan dibangkitkan dua file yaitu file OBJ yang dapat dimuat ke memori CPU dan file LST hasil proses debugger, Sebuah file HEX juga bisa dibangkitkan jika kita memilih Generate HEX File Also dalam tag Option.

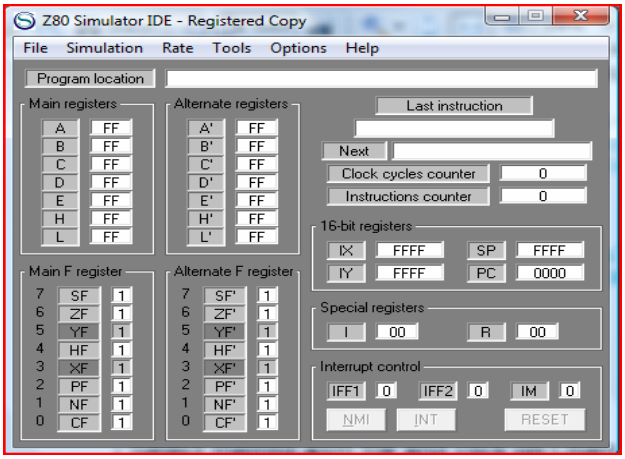
- Menu Option
  - Enable Logging ; memungkinkan simulator untuk menampilkan isi dari file LOG.TXT yang menunjukkan perubahan isi register dan status flag Z80 setelah program dieksekusi.
  - HALT Stops Simulation ; memungkinkan menghentikan simulasi secara otomatis jika menemui perintah HALT.
  - FF Power On Defaults ; mengganti nilai awal register Z80 dari nilai 00H ke FFH.
  - Refresh Memory Editor ; memperbaharui isi memori setiap kali perintah simulasi diaktifkan untuk semua nilai laju simulasi.
  - Refresh Breakpoint Manager ; jika pilihan ini diaktifkan dan Breakpoint Manager dimulai, akan memperbaharui breakpoint tiap instruksi simulasi dimulai.

- Save Position ; menetapkan posisi masing-masing jendela tampilan simulator.
- Auto Start Option ; mengatur tampilan dan bagian menu apa saja yang akan ditampilkan saat kita menjalankan pertama kali Z80 Simulator IDE ini juga file yang akan dihasilkan.
- Change Clock Frequency ; mengganti nilai parameter frekuensi untuk menghitung durasi real-time simulasi, nilainya dalam MHz.
- Change Starting Address ; mengganti alamat awal program yang akan disimulasikan, nilai default adalah 0000H.

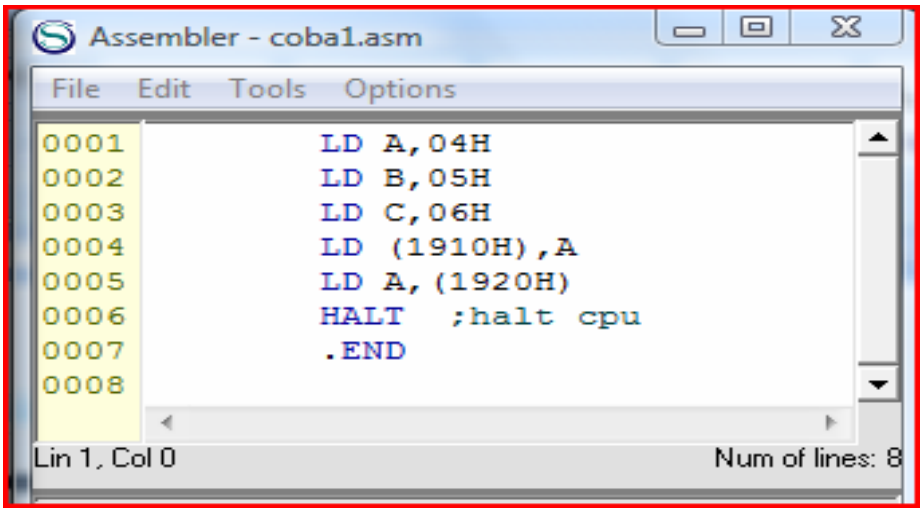
D. LANGKAH KERJA

Sebagai latihan lakikan beberapa langkah kegiatan sebagai berikut :

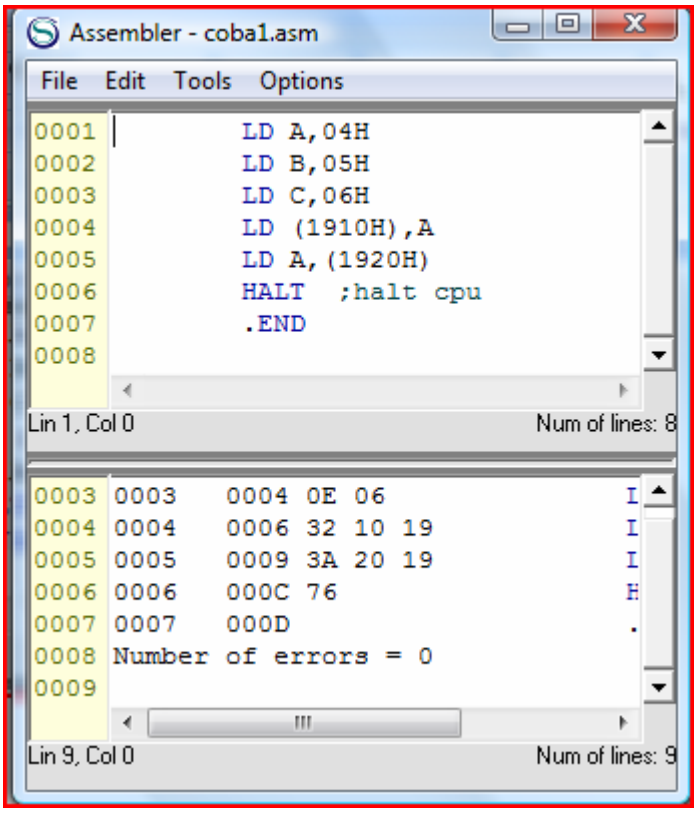
1. Jalankan Z80 Simulator IDE



2. Klik Tools / Assembler dan ketik program coba sebagai berikut, dengan ketentuan setiap mengetik perintah harus didahului dengan menekan tombol Tab pada keyboard untuk menghindari kesalahan tempat penulisan Label :

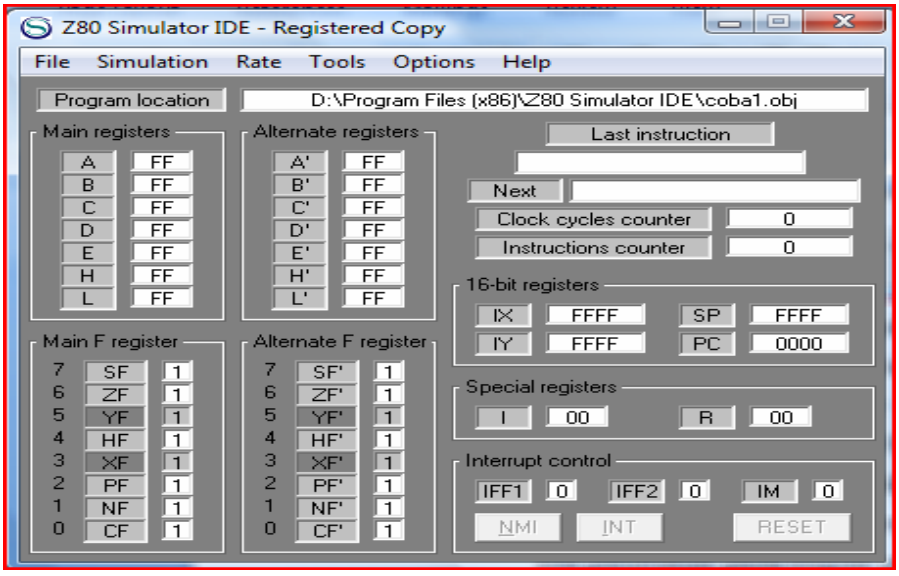


3. Pada jendela Assembler, klik Tools / Assemble untuk mengkompile program dan mengetahui apakah ada kesalahan yang terjadi.

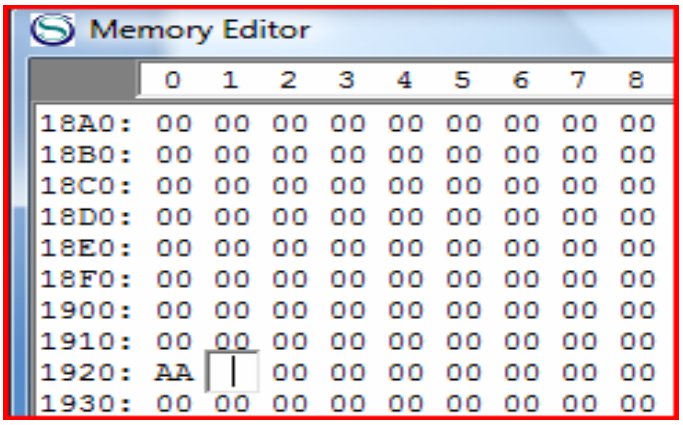


Dapat dilihat dalam tampilan jendela assembler, jendela atas menampilkan mnemonik Z80 yang kita ketik dan jendela bawah menampilkan hasil kompile yang menunjukkan opcode dari perintah yang ada di atas (bisa dibandingkan dengan tabel mnemonik Z80) dan jumlah kesalahan yang ditemukan.

4. Pada jendela Assembler klik Tools / Assemble & Load untuk memuat program ke simulator, hasilnya dilihat pada jendela Z80 Simulator IDE nampak ada perubahan pada bagian program location



5. Pada jendela Z80 Simulator IDE klik Tools / Memory Editor, scroll ke bawah sampai tampil lokasi memori 1920H. Klik lokasi tersebut, masukkan data AA dan diakhiri menekan ENTER.



6. Cek pilihan Rate adalah Fast dan Enable Logging pada menu Option telah dipilih.

7. Jalankan simulasi dengan mengklik Simulation / Start atau menekan F1

8. Amati perubahan di alamat 1910H dari bernilai 00H menjadi 04H dan akan tampil jendela baru log.txt yang menampilkan resume semua langkah dan hasil simulasi berupa perubahan nilai register dan status flag.

```
18E0: 00 00 00
18F0: 00 00 00
1900: 00 00 00
1910: 04 00 00
1920: AA 00 00
1930: 00 00 00
```

d:\program files (x86)\z80 simulator ide\log.txt

Print Copy

```
Simulation started at 17/09/2010 3:26:38.
NO.     PC    Instruction     A  SZYHXPNC  B  C  D  E  H  L  IX   IY   SP   A' SZYHXPNC' B' C' D' E' H' L'

1       0000  LD A,04H        04 11111111  FF FF FF FF FF FF FFFF FFFF FFFF FF 11111111  FF FF FF FF FF FF
2       0002  LD B,05H        04 11111111  05 FF FF FF FF FF FFFF FFFF FFFF FF 11111111  FF FF FF FF FF FF
3       0004  LD C,06H        04 11111111  05 06 FF FF FF FF FFFF FFFF FFFF FF 11111111  FF FF FF FF FF FF
4       0006  LD (1910H),A    04 11111111  05 06 FF FF FF FF FFFF FFFF FFFF FF 11111111  FF FF FF FF FF FF
5       0009  LD A,(1920H)    AA 11111111  05 06 FF FF FF FF FFFF FFFF FFFF FF 11111111  FF FF FF FF FF FF
6       000C  HALT            AA 11111111  05 06 FF FF FF FF FFFF FFFF FFFF FF 11111111  FF FF FF FF FF FF
Simulation stopped at 17/09/2010 3:26:38.
0.15 seconds elapsed.
Real-time duration of the simulation is 51 clock cycles or 12.75 µs at 4 MHz.
```