Modul Praktikum
Sistem Mikropressor dan Mikrokontroler

Robotic and Embedded System Laboratory
Jurusan Teknik Komputer
Universitas Andalas
2020

MODUL 2

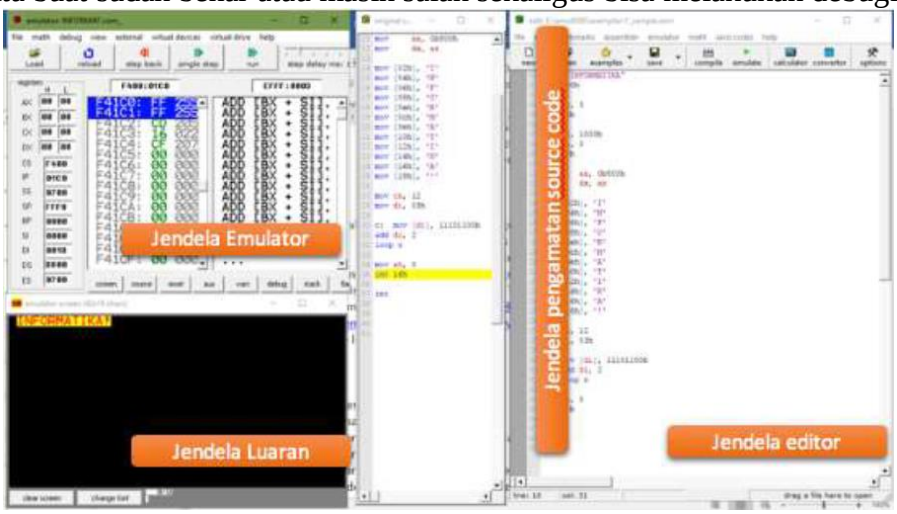
ASSEMBLY II

1.1. TUJUAN

1. Memahami dan mengetahui intruksiaritmatika dalam bahasa assembly
2. mampu mengenal salah satu toolspemrograman bahasa assembly
3. mampu menggunakan editor emu8086 untuk menulis kode bahasa assembly
4. mampu meng-compiledan menjalan program assembly sederhana dengan emu8086.
5. Mengenal stack dan function dalam assembly emu8086

1.2. DASAR TEORI

Emu8086 merupakan salah satu tools yang dapat digunakan untuk pemrograman assembly. Emu8086 merupakan aplikasi editor sekaligus emulator (debug, simulasi) bagi pemrograman bahasa assembler atau mikrokontroler. Denganmenggunakan aplikasi Emu8086, kita dapat mensimulasikan apakah program yang kita buat sudah benar atau masih salah sekaligus bisa melakukan debugingnya.



A. INSTRUKSI ARITMATIKA

1. OPERASI PERNAMBAHAN

A. ADD

Untuk menambah dalam bahasa assembler digunakan perintah ADD dan ADC serta INC. Perintah ADD digunakan dengan syntax : ADD Tujuan,Asal. Perintah ADD ini akan menambahkan nilai pada Tujuan dan Asal. Hasil yang didapat akan ditaruh pada Tujuan, dalam bahasa pascal sama dengan instruksi Tujuan:=Tujuan + Asal.

b. ADC

Perintah ADC digunakan dengan cara yang sama pada perintah ADD, yaitu : ADC Tujuan,Asal. Perbedaannya pada perintah ADC ini Tujuan tempat menampung hasil pertambahan Tujuan dan Asal ditambah lagi dengan carry flag (Tujuan:=Tujuan+Asal+Carry).

c. INC

Perintah INC(Increment) digunakan khusus untuk pertambahan dengan 1. Perintah INC hanya menggunakan 1 byte memory, sedangkan perintah ADD dan ADC menggunakan 3 byte. Oleh sebab itu bila anda ingin melakukan operasi pertambahan

dengan 1 gunakanlah perintah INC. Syntax pemakainya adalah : INC Tujuan. Nilai pada tujuan akan ditambah dengan 1, seperti perintah Tujuan:=Tujuan+1 dalam Turbo Pascal. Tujuan disini dapat berupa suatu register maupun memory.

2. Pengurangan

a. SUB

Untuk Operasi pengurangan dapat digunakan perintah SUB dengan syntax: SUB Tujuan,Asal. Perintah SUB akan mengurangkan nilai pada Tujuan dengan Asal. Hasil yang didapat akan ditaruh pada Tujuan, dalam bahasa pascal sama dengan instruksi Tujuan:=Tujuan-Asal.

b. SBB

Seperti pada operasi penambahan, maka pada operasi pengurangan dengan bilangan yang besar(lebih dari 16 bit), bisa anda gunakan perintah SUB disertai dengan SBB(Substract With Carry). Perintah SBB digunakan dengan syntax: SBB Tujuan,Asal. Perintah SBB akan mengurangkan nilai Tujuan dengan Asal dengan cara yang sama seperti perintah SUB, kemudian hasil yang didapat dikurangi lagi dengan Carry Flag(Tujuan:=Tujuan-Asal-CF).

C. DEC

Perintah DEC(Decrement) digunakan khusus untuk pengurangan dengan 1. Perintah DEC hanya menggunakan 1 byte memory, sedangkan perintah SUB dan SBB menggunakan 3 byte. Oleh sebab itu bila anda ingin melakukan operasi pengurangan dengan 1 gunakanlah perintah DEC. Syntax pemakaian perintah dec ini adalah: DEC Tujuan. Nilai pada tujuan akan dikurangi 1, seperti perintah Tujuan:=Tujuan-1 dalam Turbo Pascal. Tujuan disini dapat berupa suatu register maupun memory.

3. OPERASI PERKALIAN

Untuk perkalian bisa digunakan perintah MUL dengan syntax: MUL Sumber Sumber disini dapat berupa suatu register 8 bit(Mis:BL,BH,..), register 16 bit(Mis: BX,DX,..) atau suatu varibel. Ada 2 kemungkinan yang akan terjadi pada perintah MUL ini sesuai dengan jenis perkalian 8 bit atau 16 bit. Bila Sumber merupakan 8 bit seperti MUL BH maka komputer akan mengambil nilai yang terdapat pada BH dan nilai pada AL untuk dikalikan. Hasil yang didapat akan selalu disimpan pada register AX. Bila sumber merupakan 16 bit seperti MUL BX maka komputer akan mengambil nilai yang terdapat pada BX dan nilai pada AX untuk dikalikan. Hasil yang didapat akan disimpan pada register DX dan AX(DX:AX), jadi register DX menyimpan Word tingginya dan AX menyimpan Word rendahnya.

4. PEMBAGIAN

a. DIV

Operasi pada pembagian pada dasarnya sama dengan perkalian. Untuk operasi pembagian digunakan perintah DIV dengan syntax: DIV Sumber. Bila sumber merupakan operand 8 bit seperti DIV BH, maka komputer akan mengambil nilai pada register AX dan membaginya dengan nilai BH. Hasil pembagian 8 bit ini akan disimpan pada register AL dan sisa dari pembagian akan disimpan pada register AH.

Bila sumber merupakan operand 16 bit seperti DIV BX, maka komputer akan mengambil nilai yang terdapat pada register DX:AX dan membaginya dengan nilai BX. Hasil pembagian 16 bit ini akan disimpan pada register AX dan sisa dari pembagian akan disimpan pada register DX.

B. **STACK**

Stack merupakan bagian memori yang digunakan untuk menyimpan nilai dari suatu register secara sementara. Operasi stack dinamakan juga LIFO (Last In First Out). Bila kita terjemahkan secara bebas, stack artinya adalah 'tumpukan'. Stack adalah bagian memory yang digunakan untuk menyimpan nilai darisuatu register untuk sementara. Operasi- operasi pada assembler yang langsung menggunakan stack misalnya pada perintah PUSH, POP, PUSF dan POPF. Stack digunakan dengan instruksi CALL untuk menyimpan alamat yang dikembalikan pada prosedur, instruksi RET mengambil nilai ini dari stack dan mengembalikannya ke offset. Stack menggunakan algoritma LIFO (Last In First Out) artinya jika kita push nilai satu per satu kedalam stack : 1,2,3,4,5 nilai pertama yang dapat kita pop adalah 5, lalu 4,3,2, dan terakhir 1

1.3. PERCOBAAN

A. ALAT DAN BAHAN

1. Laptop/PC
2. Aplikasi Emu8086

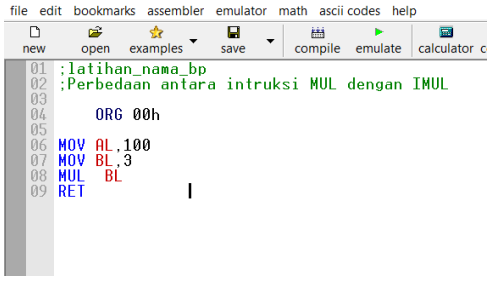
B. MEMULAI PERCOBAAN

1. Hidukan laptop
2. Buka aplikasi Emu8086

C. PERCOBAAN 1: PENGGUNAAN INTRUKSI ARITMATIKA

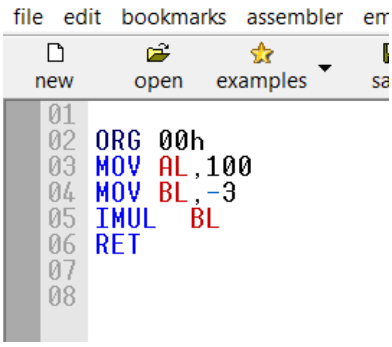
- Perbedaan antara intruksi MUL dengan IMUL pada intruksi perkalian

1) Klik new dan pilih template .com
2) Tuliskan program dibawah ini :



3) Save program dan lakukan compile program dengan klik compile
4) Emulate program dengan klik emulate
5) Perhatikan *step by step* proses eksekusi dari setiap baris program

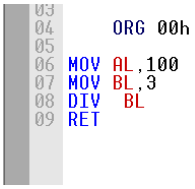
6) Hasil program di screenshoot dan buatlah Analisa program
7) Klik new file dan pilih Klik new dan pilih template .com
8) Tuliskan program dibawah ini :



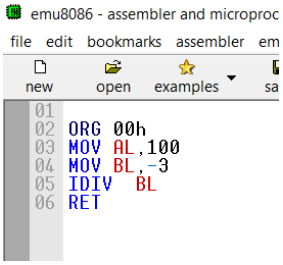
9) Ulangi Langkah 3-5
10) Hasil program di screenshoot dan buatlah Analisa program

- Perbedaan antara intruksi DIV dan IDIV

11) Klik new dan pilih template .com
12) Tuliskan program dibawah ini :



13) Save program dan lakukan compile program dengan klik compile
14) Emulate program dengan klik emulate
15) Perhatikan *step by step* proses eksekusi dari setiap baris program
16) Hasil program di screenshoot dan buatlah Analisa program
17) Klik new file dan pilih Klik new dan pilih template .com
18) Tuliskan program dibawah ini :



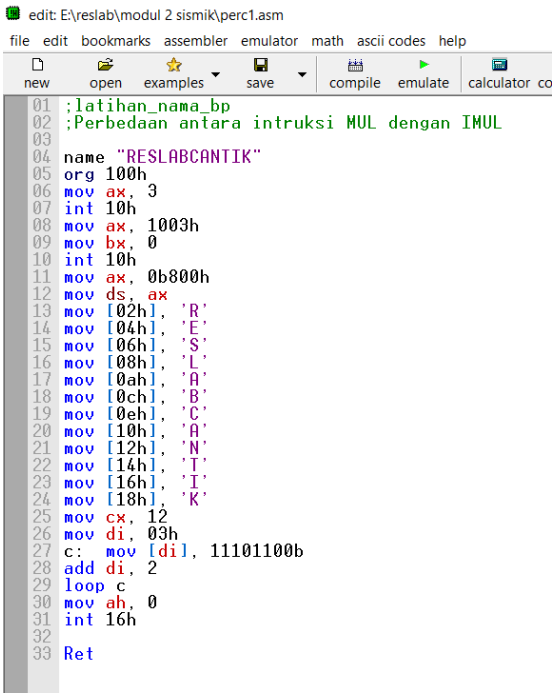
19) Ulangi Langkah 3-5
20) Hasil program di screenshoot dan buatlah Analisa program

Analisa :

1) Perbedaan Mul dengan Imul?
2) Perbedaan Div dengan Idiv?
3) Apakah yang di magsud dengan ORG dan Bagiamana penggunaannya dalam bahasa asembly ?
4) Apa Hubungan pembelajaran intruksi aritmatika dengan bahasa assembly ?

D. PERCOBAAN 2 : MEMBUAT TULISAN DENGAN INTRUKSI ARITMATIKA

1) Klik new dan pilih template .com
2) Tuliskan program dibawah ini :



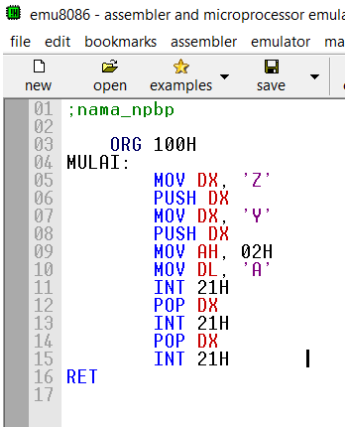
3) Save program dan lakukan compile program dengan klik compile
4) Emulate program dengan klik emulate
5) Perhatikan *step by step* proses eksekusi dari setiap baris program
6) Hasil program di screenshoot dan buatlah Analisa program
7) Modifikasilah program tersebut di atas sehingga mampu memunculkan tulisan nama praktikan!
8) Hasil program modifikasi di screenshoot dan buatlah Analisa program

Analisa :

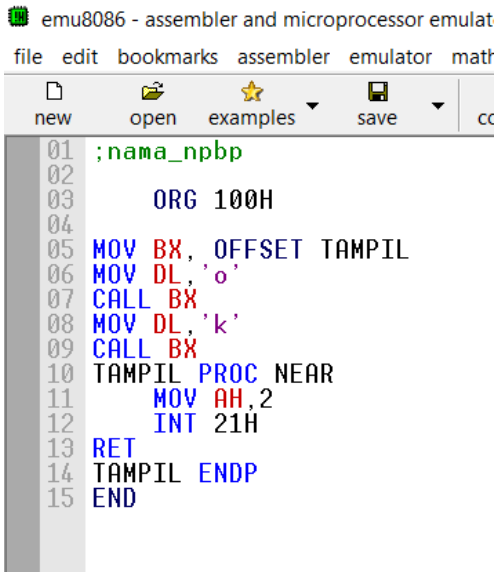
1) jelaskan analisanya berdasarkan baris program-alamat register-output program !
2) Jelaskan bagaimana cara praktikan memodifikasi program tersebut agar memunculkan nama anda sendiri !

E. PERCOBAAN 3 : STACK

9) Klik new dan pilih template .com
10) Tuliskan program dibawah ini :



11) Save program dan lakukan compile program dengan klik compile
12) Emulate program dengan klik emulate
13) Perhatikan *step by step* proses eksekusi dari setiap baris program
14) Hasil program di screenshoot dan buatlah Analisa program
21) Klik new file dan pilih Klik new dan pilih template .com
22) Tuliskan program dibawah ini :



23) Ulangi Langkah 3-5

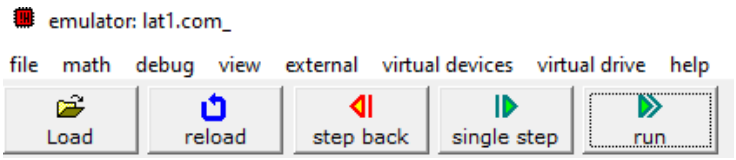
15) Hasil program di screenshoot dan buatlah Analisa program

Analisa :

1) Jelaskan ke2 program tsb dan apa perbedaannya!
2) Jelaskan setiap proses eksekusi program!

Catatan :

- Jurnal praktikum berisi screenshot hasil praktikum (hasil setiap baris), dan penjelasan dari pertanyaan Analisa yang terdapat pada modul praktikum. *setiap gambar diberi keterangan (spt. nomor dan nama)
- Pada Program Emu8086 menggunakan format hexadecimal
- Pada saat melakukan simulasi dengan emulator gunakan tools dibawah ini untuk melihat detail setiap proses yang dilakukan



- **DB** - stays for Define Byte.
- **DW** - stays for Define Word.