LAPORAN PRAKTIKUM ARSITEKTUR DAN ORGANISASI KOMPUTER

ASSEMBLY PROGRAMMING USING MARS SIMULATOR

Agus Pranata Marpaung
13323033
DIII Teknologi Komputer

INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI

Judul Praktikum

Minggu/Sesi	:	XII/1,2
Kode Mata Kuliah	:	1031103/1041103
Nama Mata Kuliah	:	ARSITEKTUR DAN ORGANISASI KOMPUTER
Setoran	:	Laporan Materi Assembly Programming using MARS Simulator dikirimkan dalam bentuk PDF pada ecourse
Batas Waktu Setoran	:	*28 November 2023 jam 8:00*
Tujuan	:	1. Mahasiswa mampu mengenal Assembly Programming 2. Mahasiswa mampu menggunakan MARS Simulator 3. Mahasiswa mampu menerapkan Assembly Programming pada MARS Simulator

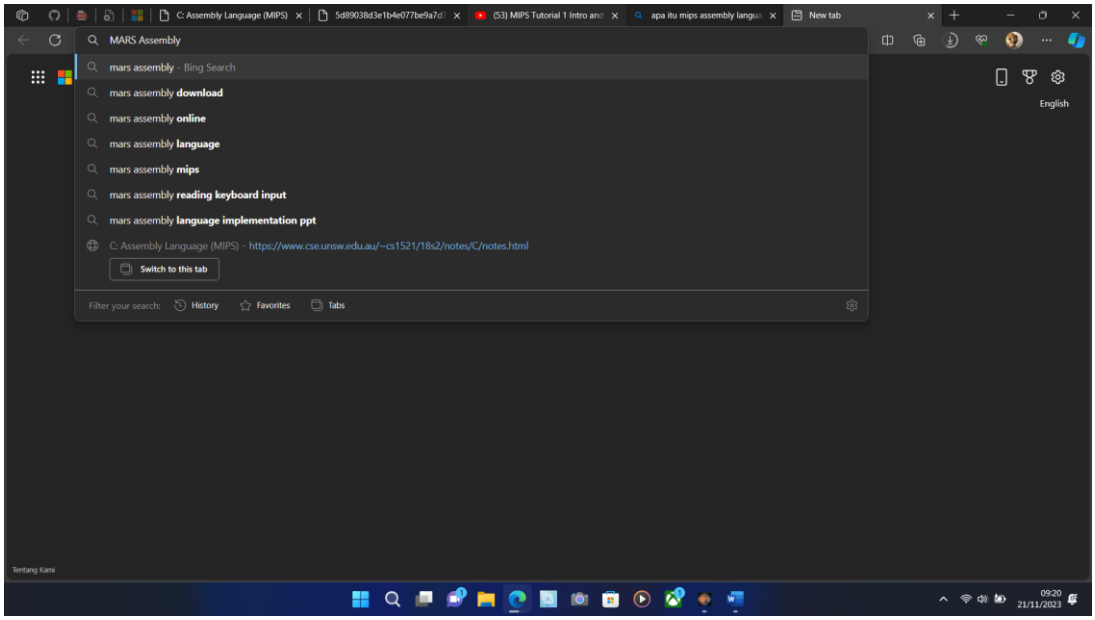
Petunjuk

1. Laporan praktikum dikerjakan secara individu (tidak berkelompok)
2. Setiap individu diperbolehkan memberikan pertanyaan dan diskusi melalui WAG pada sesi kedua di hari praktikum
3. Tidak ada toleransi keterlambatan, jika terlambat maka akan terjadi pengurangan nilai.
4. Dalam pengerjaan laporan praktikum, dilarang keras melakukan plagiasi (mencontek).

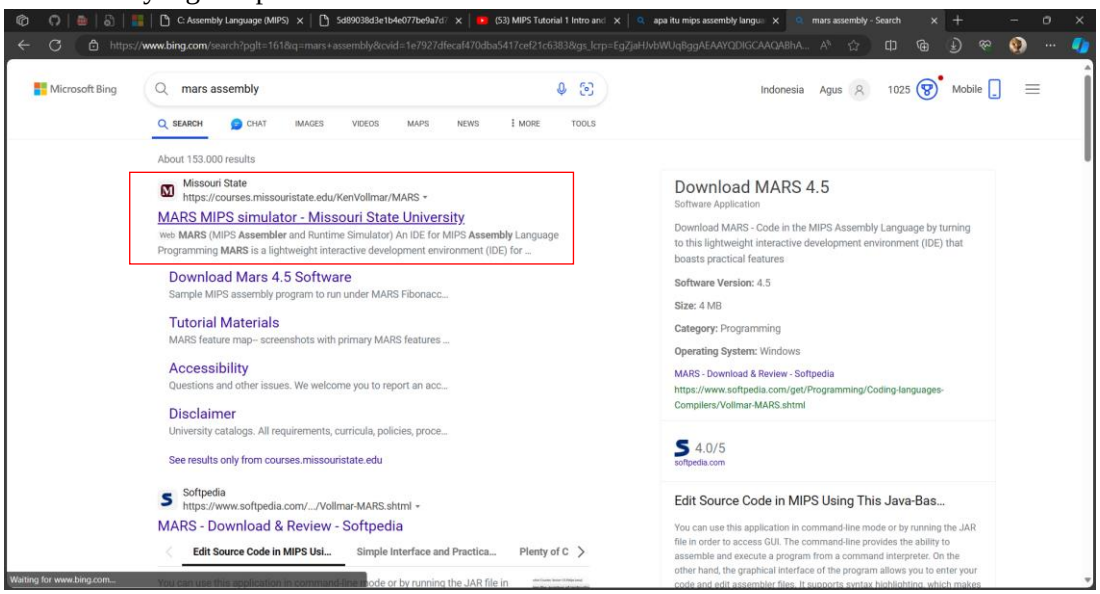
ARSITEKTUR DAN ORGANISASI KOMPUTER

1. MIPS Tutorial 1 Intro and Mars

- Buka Browser dan ketik MARS Assembly di kotak pencarian browser kemudian tekan enter.

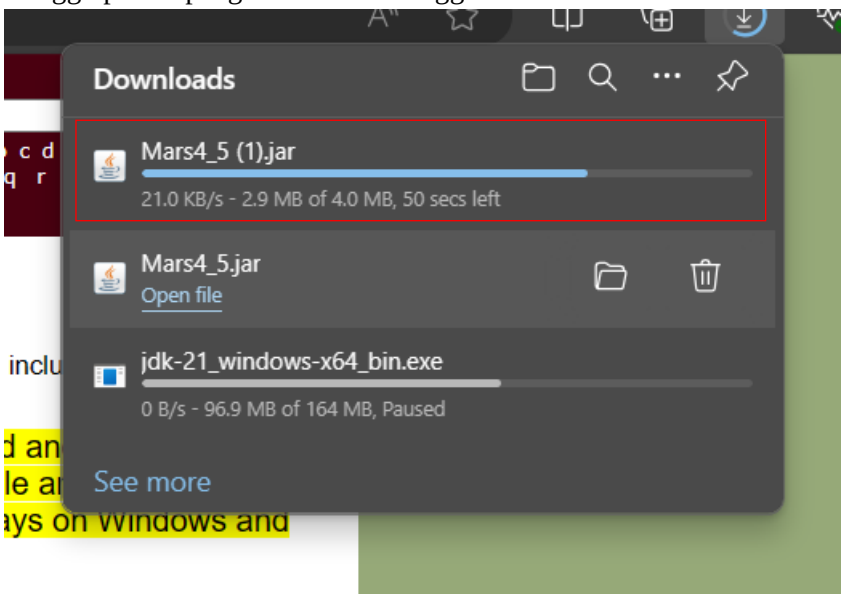


- Klik tautan yang ada pada kotak merah.

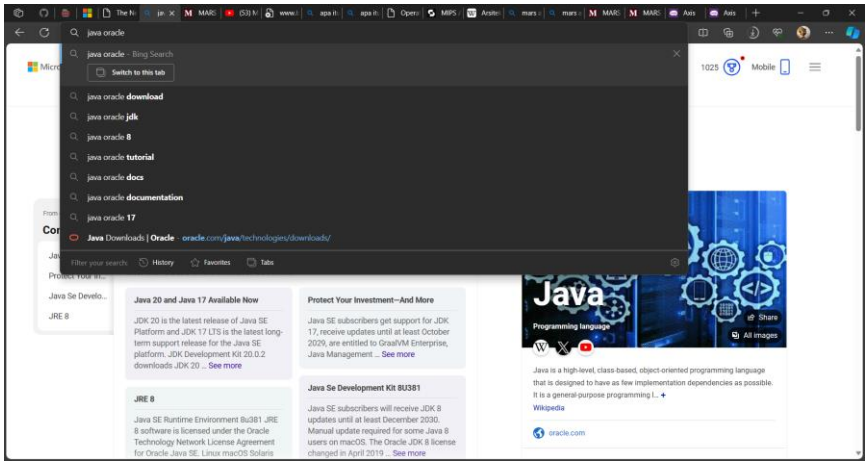


- Klik tombol Download MARS

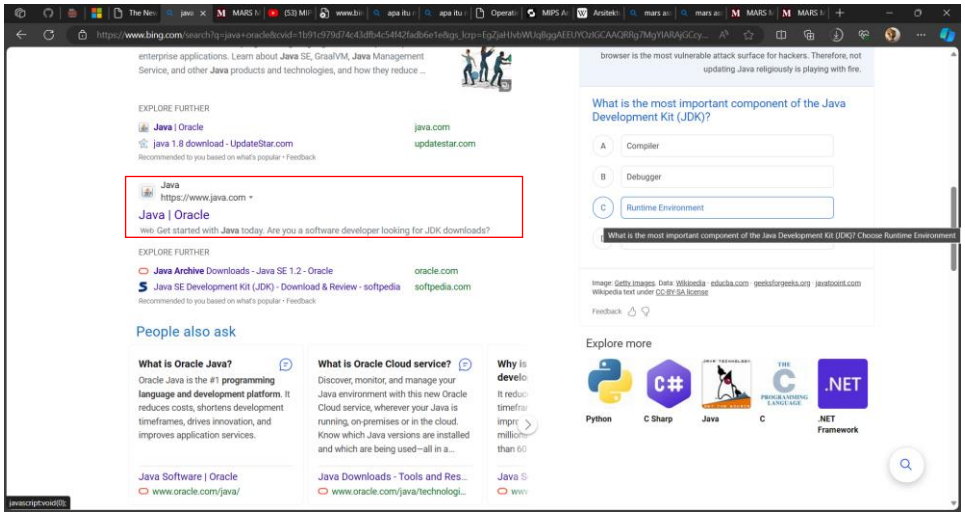
- Tunggu proses pengunduhan file hingga selesai.



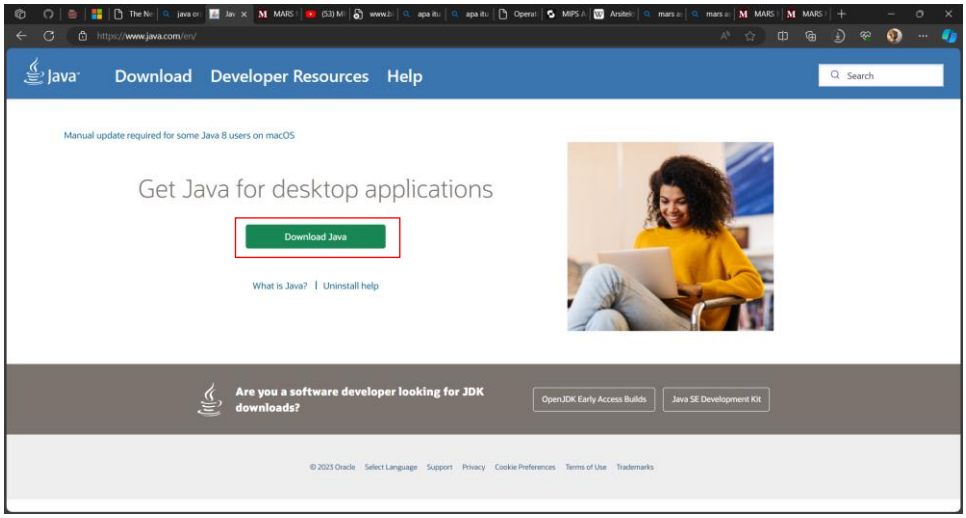
- Sebelum membuka file MARS tersebut, pastikan Java sudah diinstal pada perangkat yang anda gunakan. Untuk menginstall Java, ketik Java Oracle di kotak pencarian browser dan tekan enter.



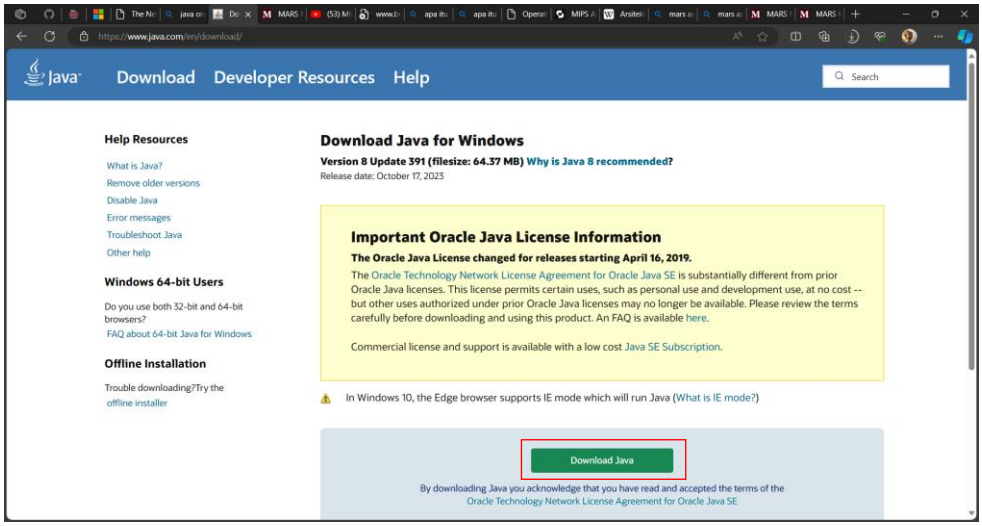
- Scroll ke bawah dan klik tautan yang ada pada kotak merah berikut.



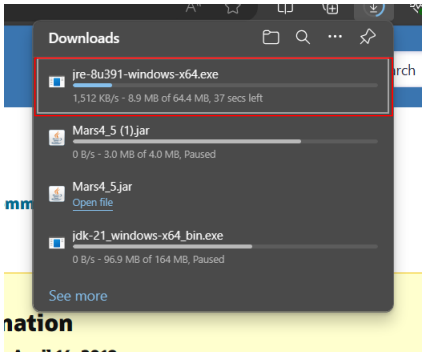
- Klik tombol Download Java pada halaman Java



- Klik tombol Download Java

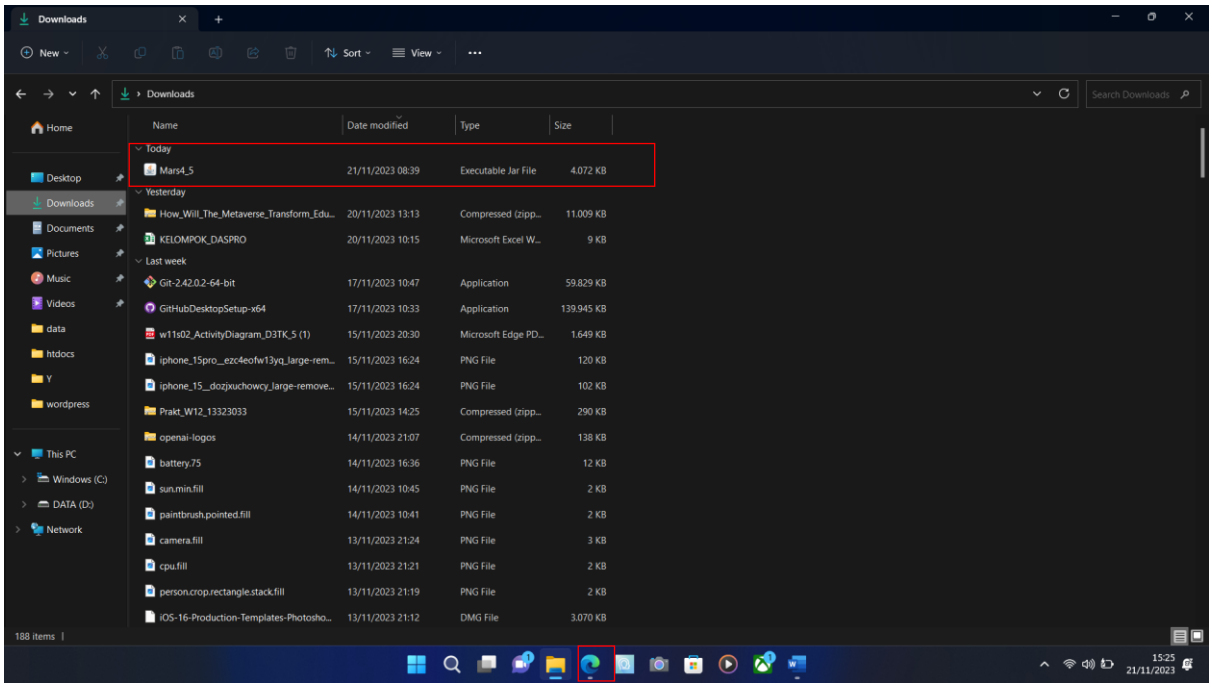


- Tunggu proses pengunduhan Java hingga selesai.

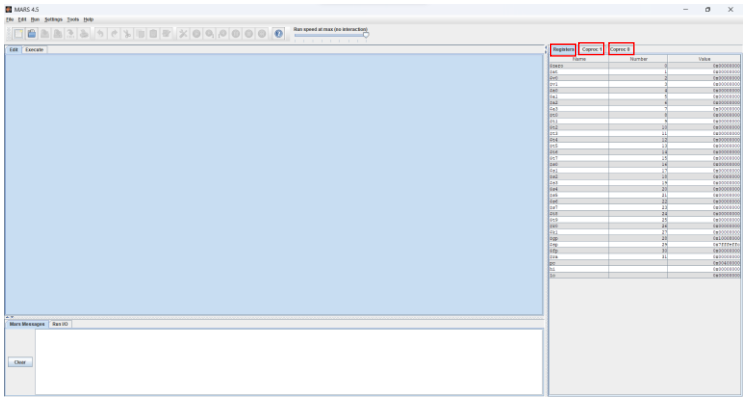


2. MIPS Tutorial 2 Registers

- Buka file explorer kemudian klik bagian Downloads dan klik dua kali file berbentuk jar yang Bernama Mars4_5

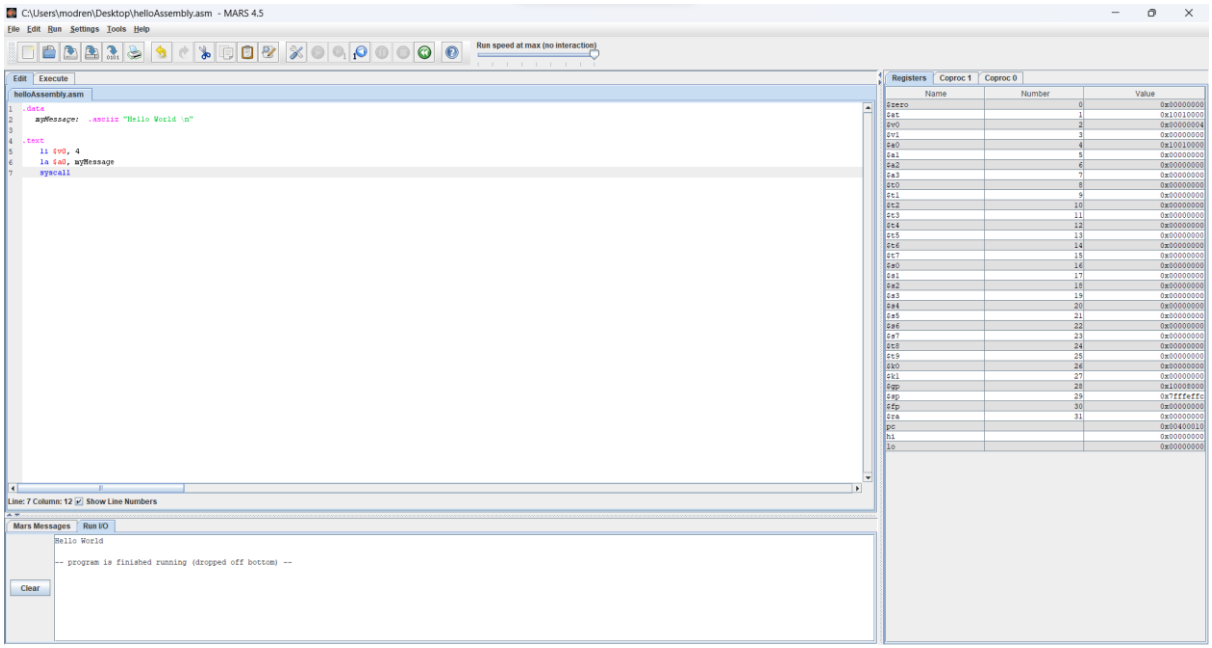


- Berikut tampilan dari MARS



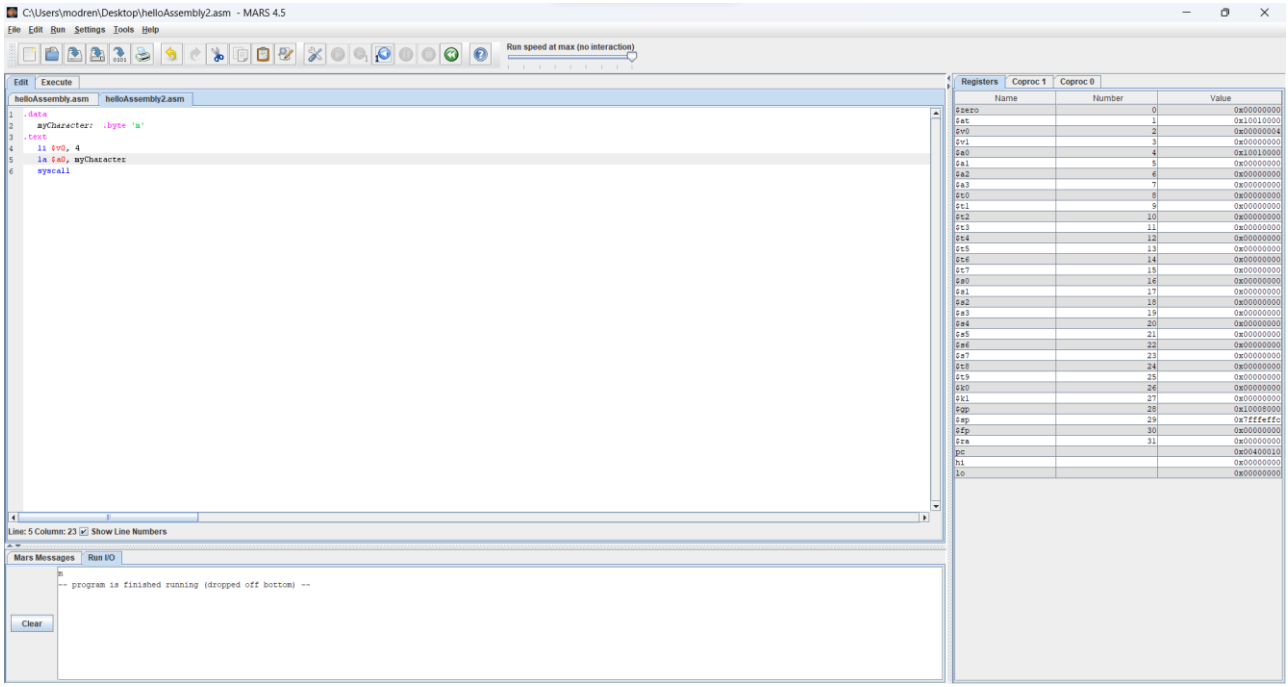
- **Registers** : Register berada pada CPU dan ini merupakan CPU Register dan register merupakan memori tercepat yang ada pada komputer.
- **Coproc 1** : Coprocessor 1(floating point unit) registers.
- **Coproc 0** : Memilih Coprocessor 0 (exceptions and interrupts) registers.

3. MIPS Tutorial 3 Hello Assembly



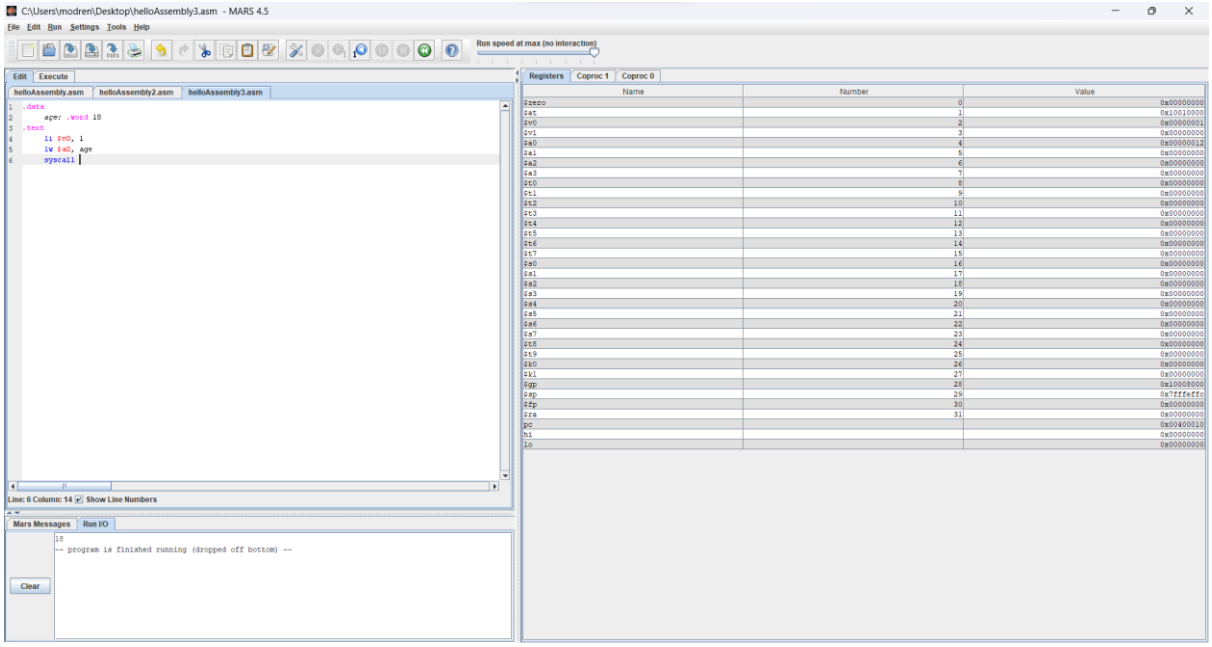
- **.data** : untuk mendeklarasi variabel
- **myMessage** : data untuk menampilkan pesan
- **.asciiz** : tipe data ascii
- **“Hello World \n”** : data
- **.text** : Sebuah variabel data untuk teks
- **li** : load immediate
- **$v0, 4** : mengatur nilai register dengan nilai 4
- **la** : load address
- **$a0, myMessage** : melakukan sistem panggilan

4. MIPS Tutorial 4 Printing a Character



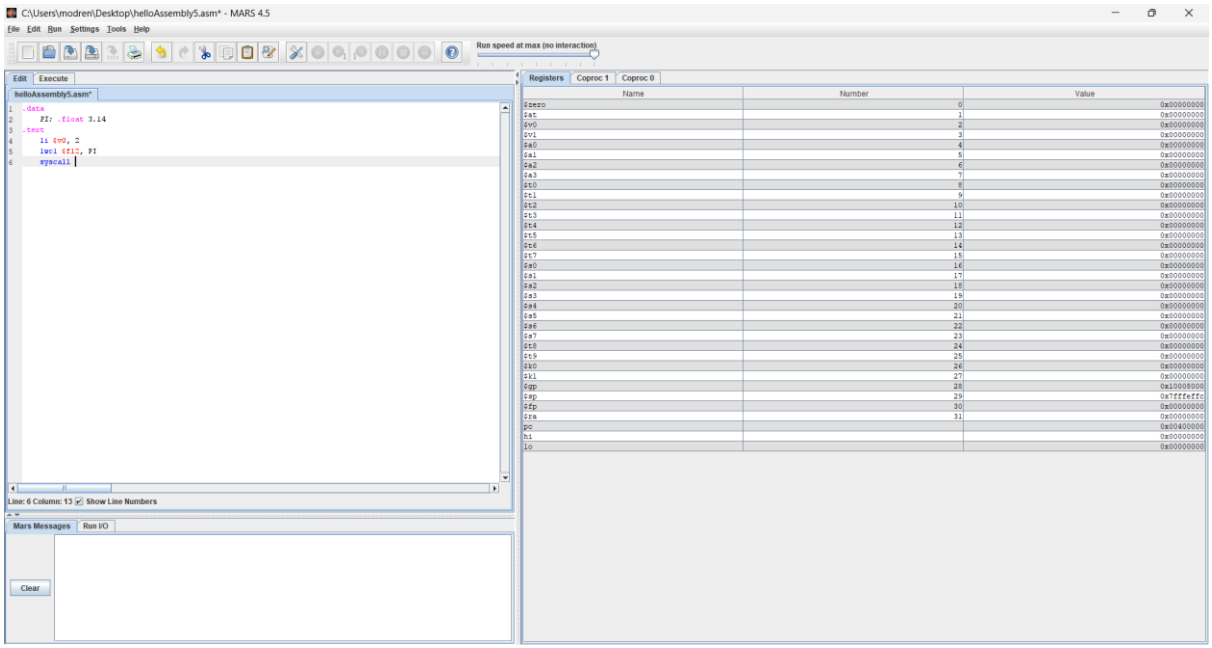
- **.data** : untuk mendeklarasi variabel
- **myCharacter** : menampilkan karakter
- **.byte** : tipe data byte
- **'m'** : sebuah karakter
- **.text** : sebuah variabel data untuk teks
- **li** : load immediate
- **$v0, 4** : mengatur nilai register dengan nilai 4
- **la** : load address
- **$a0, myCharacter** : memanggil print karakter
- **syscall** : melakukan sistem panggilan

5. MIPS Tutorial 5 Printing an Integer



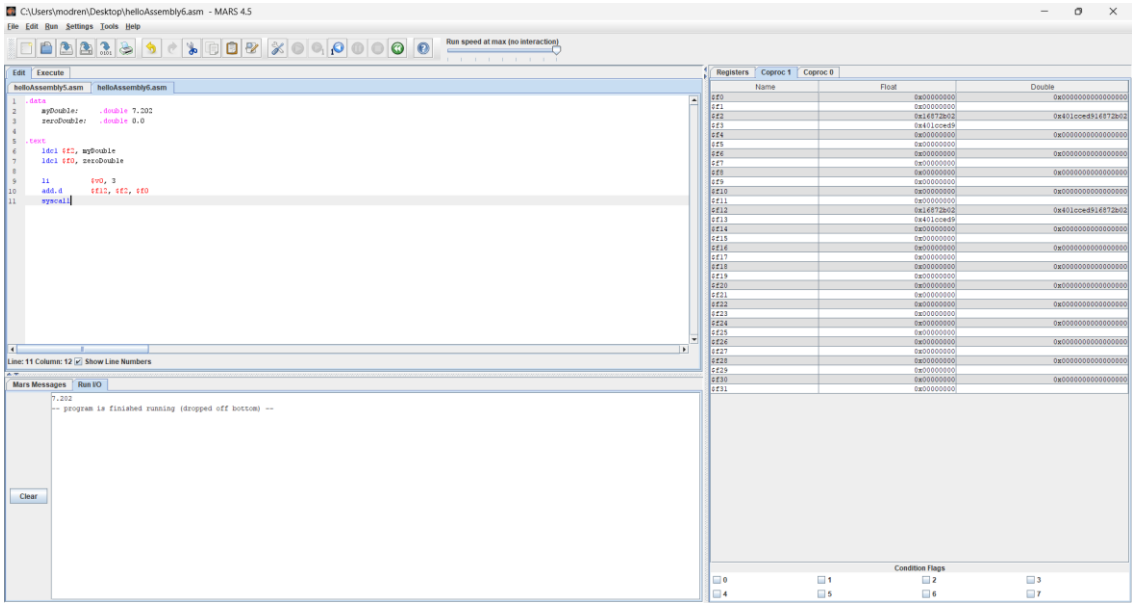
- **.data** : untuk mendeklarasi variabel
- **age** : data untuk umur
- **.word** : tipe data untuk word dengan menggunakan 32-bit words
- **10** : data angka pada umur
- **.text** : sebuah variabel data untuk teks
- **li** : load immediate
- **$v0, 1** : mengatur nilai register dengan nilai 1
- **la** : load address
- **$a0, age** : memanggil print umur
- **syscall** : melakukan sistem panggilan

6. MIPS Tutorial 6 Printing a Float



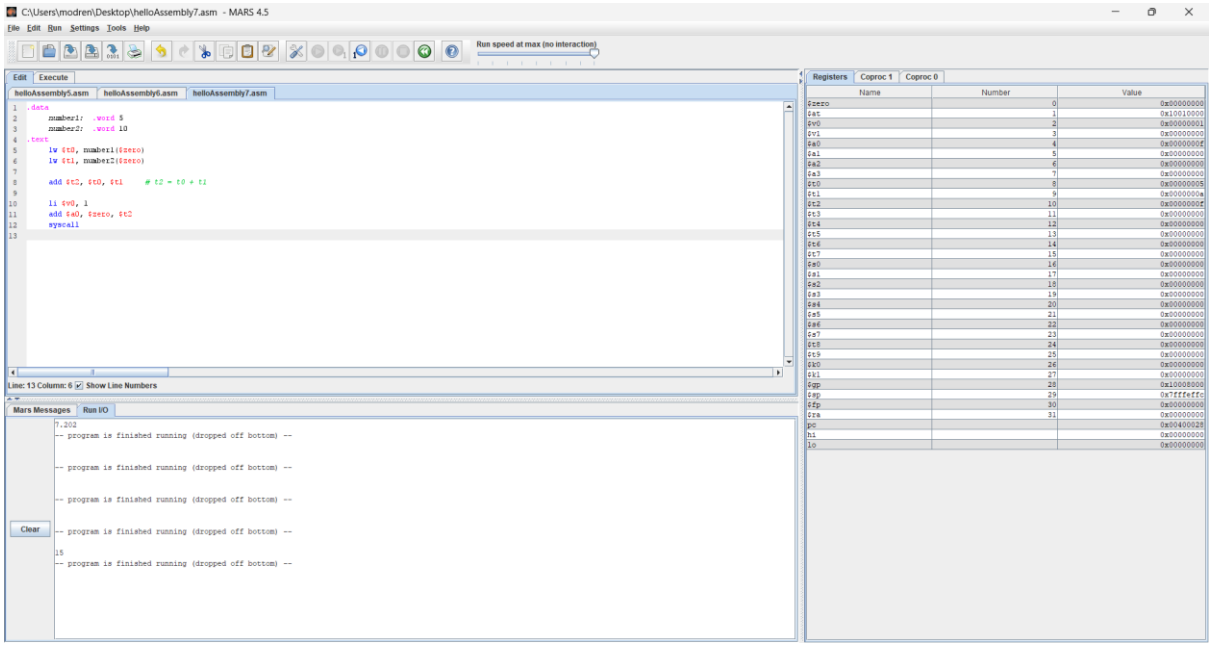
- **.data** : untuk mendeklarasi variabel
- **PI** : data untuk PI (π)
- **.float** : tipe data untuk float yang dapat berpindah-pindah
- **3,14** : data yang dimiliki oleh PI
- **.text** : sebuah variabel data untuk teks
- **li** : load immediate
- **$v0, 2** : mengatur nilai register dengan nilai 2
- **lwc1** : load word into Coprocessor 1
- **$f12** : Suatu data yang ada di Coprocessor 1
- **PI** : data yang akan digunakan pada Coprocessor 1.
- **syscall** : melakukan sistem panggilan

7. MIPS Tutorial 7 Printing a Double



- **.data** : untuk deklarasi variabel.
- **myDouble** : data untuk menunjukkan nilai *double*.
- **zeroDouble** : data untuk menunjukkan nilai 0.
- **.double** : tipe data untuk myDouble dan zeroDouble.
- **.text** : sebuah variabel data untuk teks.
- **ldc1** : load double word Coprocessor 1.
- **$f2** : suatu register memori.
- **myDouble** : Nilai yang akan dimuat ke dalam register.
- **$f0** : suatu register memori.
- **zeroDouble** : Nilai yang akan dimuat ke dalam register.
- **li** : load immediate.
- **$v0, 3** : mengatur nilai register dengan nilai 3.
- **add.d** : menjumlahkan nilai register $f2, $f0 dan hasilnya disimpan dalam register $f12.
- **syscall** : melakukan sistem panggilan.

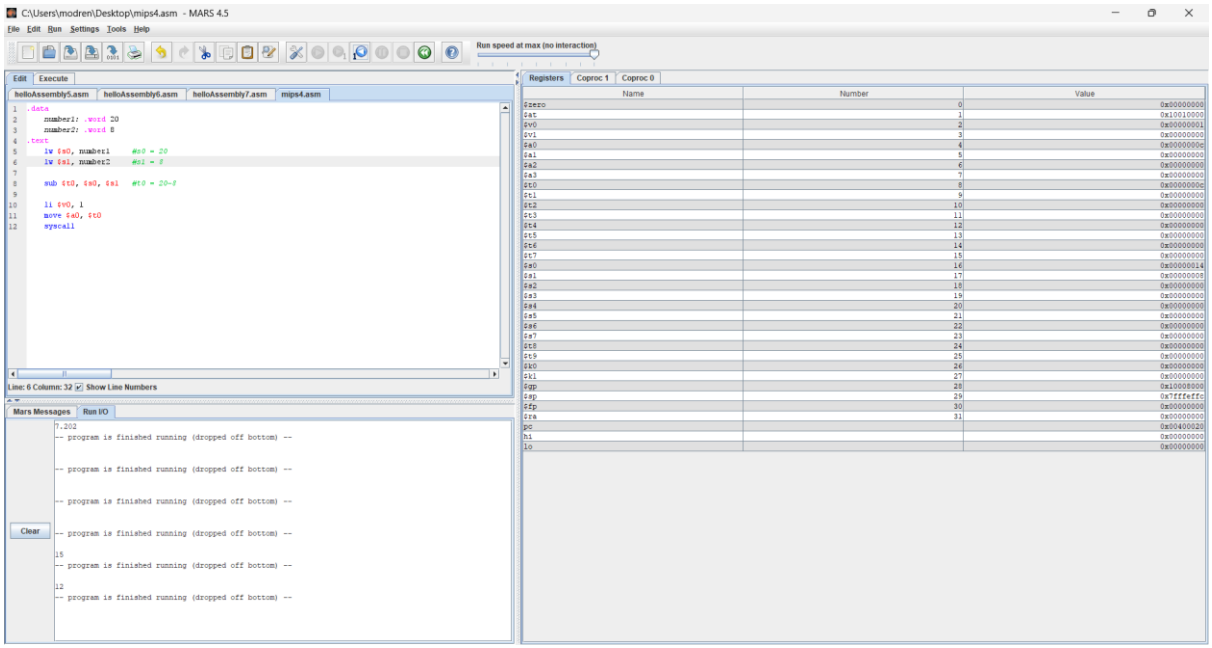
8. MIPS Tutorial 8 Adding Integers



- **.data** : untuk deklarasi variabel.
- **number1: .word5** : mengalokasikan 4 byte di memori dan menyimpan nilai 5 di lokasi tersebut
- **number2: .word10** : mengalokasikan 4 byte di memori dan menyimpan nilai 10 di lokasi tersebut.
- **.text** : sebuah variabel data untuk teks
- **lw $t0, number1($zero)** : memuat nilai dari number1 ke dalam register $t0.
- **lw $t1, number2($zero)** : Memuat nilai dari number2 ke dalam register $t1.
- **add $t2, $t0, $t1** : Menambahkan nilai di register $t0 dengan nilai di register $t1 dan hasilnya disimpan di register $t2.
- **li $v0, 1** : Mengatur sistem call.
- **add $a0, $zero, $t2** : Menyalin nilai dari register $t2 ke register argumen $a0.

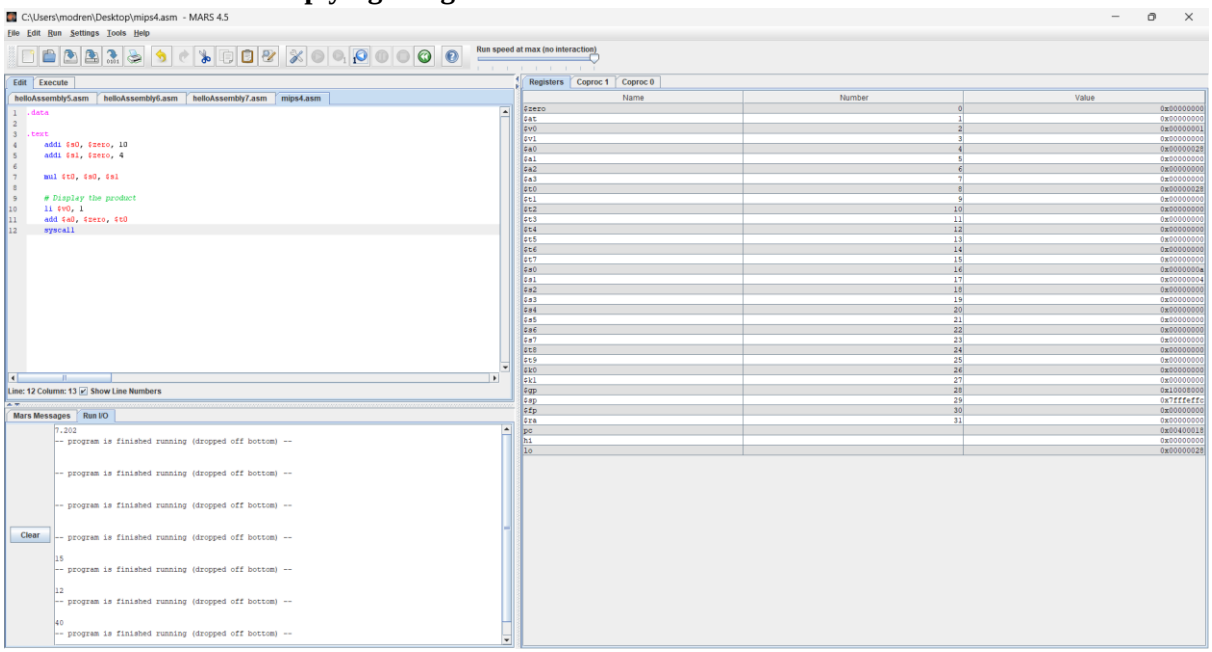
- **syscall** : Menjalankan sistem call untuk mencetak nilai yang ada di register $a0.

9. MIPS Tutorial 9 Subtracing Integers



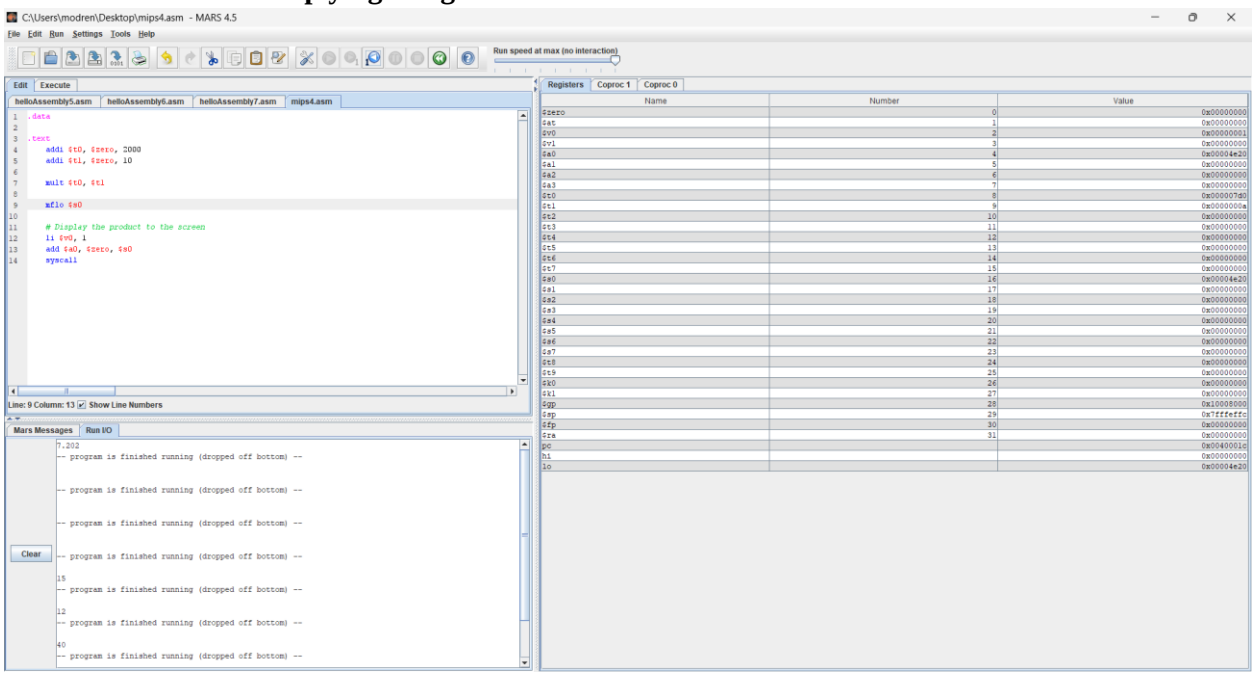
- **.data** : untuk deklarasi variabel number1 dan number2.
- **number1: .word20** : mengalokasikan 4 byte di memori dan menyimpan nilai 20 di lokasi tersebut.
- **number2: .word8** : mengalokasikan 4 byte di memori dan menyimpan nilai 8 di lokasi tersebut.
- **.text** : sebuah variabel data untuk teks.
- **lw $t0, number1($zero)** : memuat nilai dari number1 ke dalam register $t0.
- **lw $t1, number2($zero)** : Memuat nilai dari number2 ke dalam register $t1.
- **sub $t0, $s0, $s1** : untuk mengurangkan nilai yang ada di register $s0 dengan nilai di register $s1, dan hasilnya disimpan di register $t0.
- **li $v0, 1** : untuk mengatur register sistem $v0.
- **move $a0, $t0** : untuk memindahkan nilai dari register $t0 ke register argumen $a0.
- **syscall** : Menjalankan sistem call untuk mencetak nilai yang ada.

10.MIPS Tutorial 10 Multiplying Integers Mul



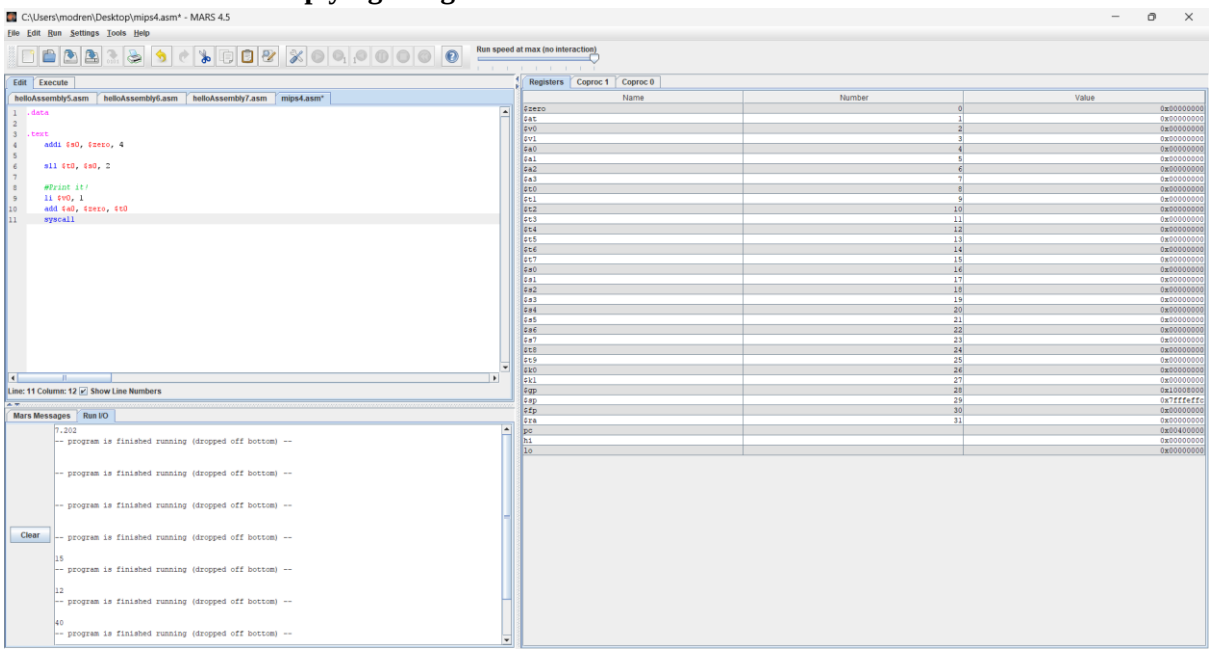
- **.data** : untuk deklarasi variabel number1 dan number2.
- **addi $s0, $zero, 10** : untuk menambahkan nilai ke register dengan $s0 diisi dengan nilai 10.
- **addi $s1, $zero, 4** : untuk menambahkan nilai ke register dengan $s1 diisi dengan nilai 4.
- **.text** : sebuah variabel data untuk teks.
- **mul $t0, $s0, $s1** : mengalikan isi dari dua register dan hasilnya disimpan ke register lain yaitu $t0 diisi dengan hasil perkalian dari $s0 dan $s1
- **li $v0, 1** : mengatur register $v0 ke nilai 1.
- **add $a0, $zero, $t0** : mengatur register $a0 dengan nilai dari $t0.
- **syscall** : Menjalankan sistem call untuk mencetak nilai yang ada.

11.MIPS Tutorial 11 Multiplying Integers Mult



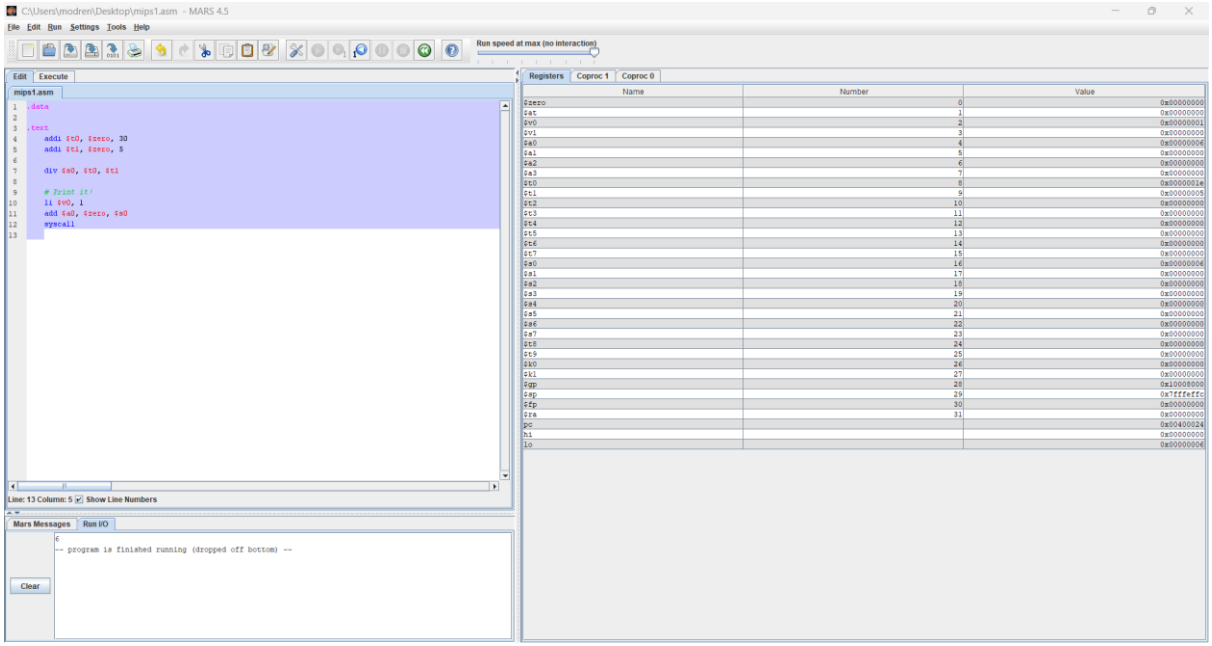
- **.data** : untuk deklarasi variabel number1 dan number2.
- **.text** : mendefinisikan segmen teks, yang berisi kode program.
- **addi $t0, $zero, 2000** : Menambahkan immediate value (2000) ke register $zero dan menyimpan hasilnya di register $t0.
- **addi $t1, $zero, 10** : Menambahkan immediate value (10) ke register $zero dan menyimpan hasilnya di register $t1.
- **mult $t0, $t1** : Mengalikan nilai yang ada di register $t0 dan $t1.
- **mflo $s0** : Mengambil nilai dari register hasil perkalian yang lebih rendah dan menyimpannya di register $s0.
- **li $v0, 1** : Menginisialisasi register sistem $v0 dengan nilai 1.
- **add $a0, $zero, $s0** : Menyalin nilai dari register $s0 ke register argumen sistem $a0.
- **syscall** : Menjalankan sistem call ntuk mencetak nilai yang ada.

12.MIPS Tutorial 12 Multiplying Integers Sll



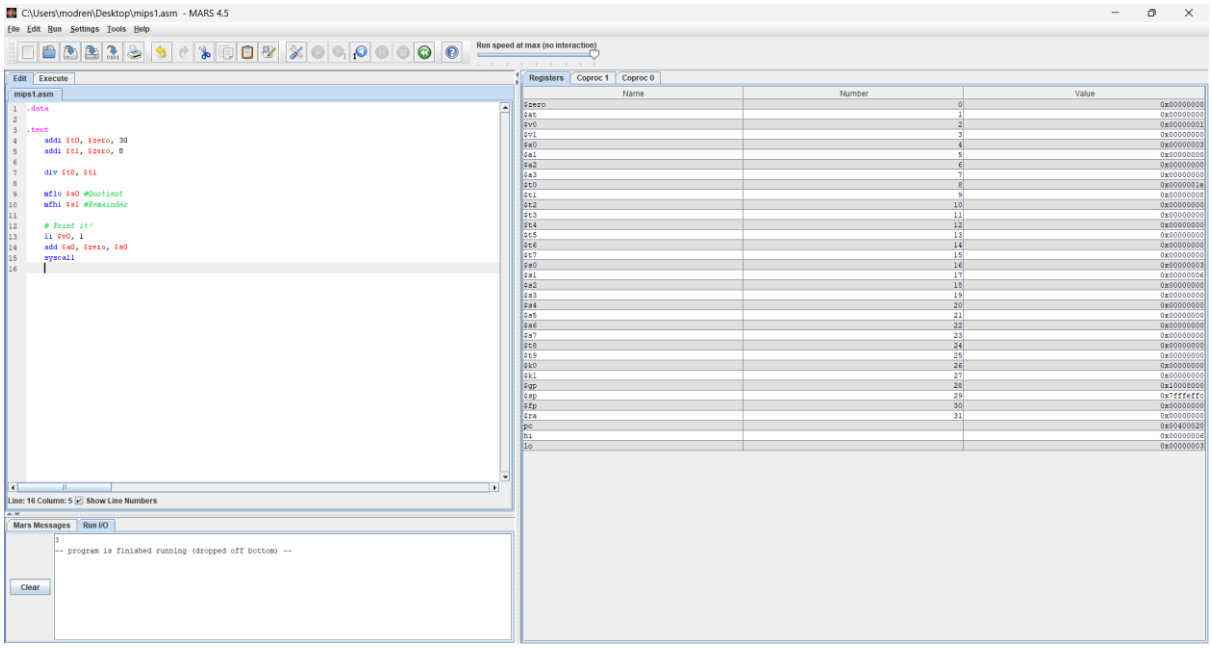
- **.data** : untuk mendeklarasikan dan menginisialisasi data statis.
- **.text** : berisi kode program utama.
- **addi $s0, $zero, 4** : menambahkan nilai 4 ke register $s0.
- **sll $t0, $s0, 2** : melakukan shift left logical (sll) pada nilai yang ada di register $s0 sebanyak 2 bit dan hasilnya disimpan di register $t0.
- **li $v0, 1** : memuat nilai 1 ke register $v0.
- **add $a0, $zero, $t0** : menambahkan nilai yang ada di register $t0 ke register $a0.
- **syscall** : Menjalankan sistem call ntuk mencetak nilai yang ada.

13.MIPS Tutorial 13 Dividing Integers part 1



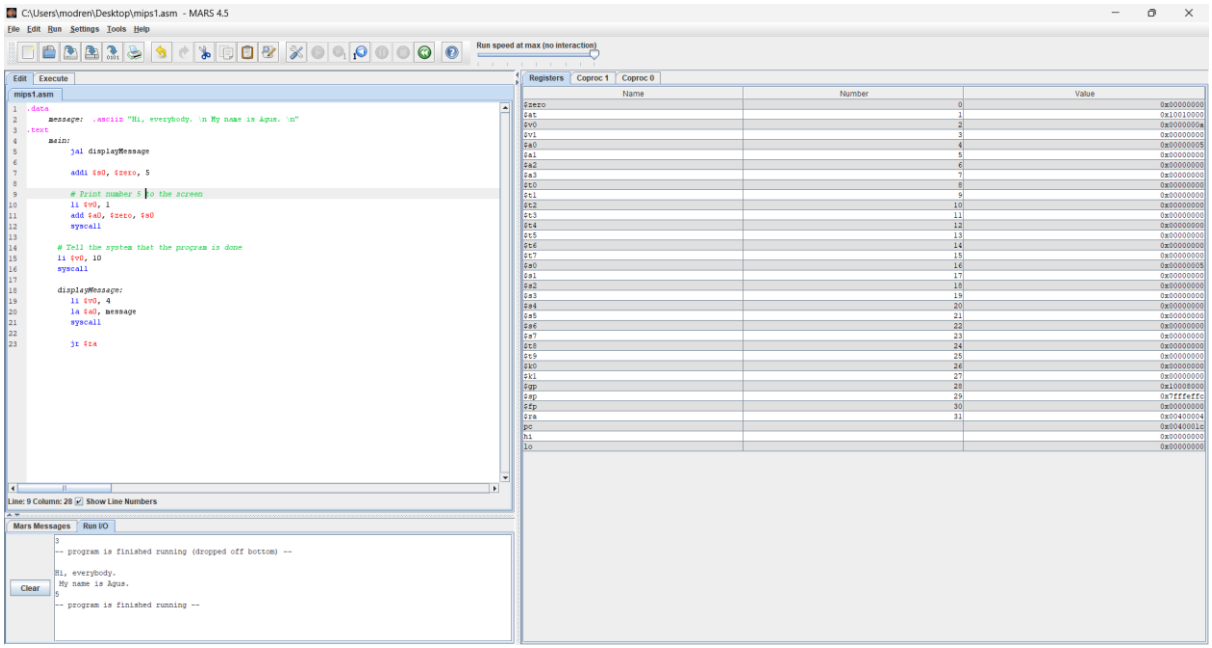
- **.data** : untuk mendefinisikan variabel-variabel yang akan digunakan dalam program.
- **.text** : berisi kode program utama atau instruksi-instruksi yang akan dieksekusi oleh CPU.
- **addi $t0, $zero, 30** : menambahkan nilai 30 ke register $t0.
- **addi $t1, $zero, 5** : menambahkan nilai 5 ke register $t1.
- **div $s0, $t0, $t1** : melakukan operasi pembagian antara nilai yang ada di register $t0 dan $t1, dan hasilnya disimpan di register $s0.
- **li $v0, 1** : memuat nilai 1 ke register $v0.
- **add $a0, $zero, $s0** : menyalin nilai yang ada di register $s0 ke register $a0.
- **syscall** : Menjalankan sistem call ntuk mencetak nilai yang ada.

14.MIPS Tutorial 14 Dividing Integers part 2



- **.data** : untuk mendefinisikan variabel-variabel yang akan digunakan dalam program.
- **.text** : berisi kode program utama atau instruksi-instruksi yang akan dieksekusi oleh CPU.
- **addi $t0, $zero, 30** : menambahkan nilai 30 ke register $t0.
- **addi $t1, $zero, 5** : menambahkan nilai 5 ke register $t1.
- **div $s0, $t0, $t1** : melakukan operasi pembagian antara nilai yang ada di register $t0 dan $t1, dan hasilnya disimpan di register $s0.
- **li $v0, 1** : memuat nilai 1 ke register $v0.
- **add $a0, $zero, $s0** : menyalin nilai yang ada di register $s0 ke register $a0.
- **syscall** : Menjalankan sistem call ntuk mencetak nilai yang ada.

15.MIPS Tutorial 15 Introduction to Functions



- **.data** : mendeklarasikan segmen data.
- **.text** : berisi kode program utama atau instruksi-instruksi yang akan dieksekusi oleh CPU.
- **main** : label untuk fungsi utama program.
- **jal displayMessage** : melakukan "jump and link" ke fungsi displayMessage.
- **addi $s0, $zero, 5** : Menambahkan nilai 5 ke register $s0.
- **li $v0, 1** : Memuat nilai 1 ke register $v0.
- **add $a0, $zero, $s0** : Menyalin nilai dari register $s0 ke register $a0.
- **syscall** : Instruksi ini mengeksekusi layanan sistem berdasarkan nilai yang ada
- **li $v0, 4** : Memuat nilai 4 ke register $v0.
- **la $a0, message** : Memuat alamat string message ke register $a0.
- **syscall** : mengeksekusi layanan sistem berdasarkan nilai yang ada
- **jr $ra** : melakukan "jump register" ke alamat yang ada di register $ra.