



PETUNJUK PRAKTIKUM

DASAR SISTEM KOMPUTER

Penyusun:

Taufiq Ismail, S.T., M.Cs.
Ahmad Azhari, S.Kom. M.Eng.
Ali Tarmuji, S.T., M.Cs.

2019

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga dapat diselesaikannya Petunjuk Praktikum ini. Sholawat dan salam semoga terlimpah kepada baginda Nabi Muhammad saw beserta keluarga, sahabat dan para pengikutnya.

Petunjuk Praktikum Dasar Sistem Komputer ini disusun dengan harapan dapat memberikan kemudahan bagi mahasiswa untuk memahami dasar-dasar sistem komputer terutama dalam pemrograman Bahasa rakitan (*assembly*). Berbagai contoh aplikasi pemrograman *assembly* dan teknologi pengembangannya diberikan yang meliputi debug, aritmatika, word process, dengan lingkungan pemrograman bervariasi, yaitu fasilitas DOS, TASM, emu8086, dan Microprocessor Simulator.

Ucapkan terima kasih yang sebesar-besarnya diberikan kepada pihak semua pihak yang telah membantu dalam proses penyusunan petunjuk praktikum ini.

Akhirnya, tiada gading yang tak retak, kritik dan saran membangun kami harapkan untuk lebih sempurnanya petunjuk praktikum ini. Semoga ilmu yang diperoleh dari membaca dan mempraktekkan petunjuk praktikum ini dapat bermanfaat dan barokah.

Yogyakarta, Agustus 2019

Penyusun

DAFTAR PENYUSUN

Taufik Ismail, S.T., M.Cs.
Ahmad Azhari, S.Kom., M.Eng.
Ali Tarmuji, S.T., M.Cs.

HALAMAN REVISI

Yang bertanda tangan di bawah ini:

Nama : Taufiq Ismail, S.T., M.Cs.

NIY : 60010314

Jabatan : Koordinator Mata Kuliah Dasar Sistem Komputer (DSK)

Dengan ini menyatakan pelaksanaan Revisi Petunjuk Praktikum Dasar Sistem Komputer untuk Program Studi Teknik Informatika telah dilaksanakan dengan penjelasan sebagai berikut:

No	Keterangan Detail Revisi (Per Pertemuan)	Tanggal Revisi	Nomor Modul
1.	Menerapkan materi praktikum ke template baru sesuai ketentuan: a. Menambahkan tujuan instruksional dan ketercapaian b. Menghilangkan pre-test c. Menghilangkan daftar pustaka per pertemuan dan menggabungkannya di akhir petunjuk praktikum	21 Agustus 2019	PP/018/I/R2
2.	Menambahkan pertemuan praktikum, dari 8 menjadi 10	21 Agustus 2019	PP/018/I/R2
	Menambahkan pertemuan khusus untuk uji kompetensi (2 kali uji kompetensi)	21 Agustus 2019	PP/018/I/R2
3.	Menggeser materi praktikum 1 s.d. 8 menjadi praktikum 2 s.d. 9	21 Agustus 2019	PP/018/I/R2
4.	Menambahkan pertemuan praktikum 1 materi perangkat keras komputer (PC)	21 Agustus 2019	PP/018/I/R2
5.	Menambahkan deskripsi tugas untuk pertemuan praktikum 2	21 Agustus 2019	PP/018/I/R2
6.	Menambahkan pertemuan praktikum 10 materi pemrograman assembly menggunakan Microprocessor Simulator	21 Agustus 2019	PP/018/I/R2

Yogyakarta, Agustus 2019

Koordinator Penyusun,

Taufiq Ismail, S.T., M.Cs.

NIY. 60010314

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Lisna Zahrotun, S.T., M.Cs.

NIK/NIY : 60150773

Jabatan : Kepala Laboratorium Praktikum Teknik Informatika

Menerangkan dengan sesungguhnya bahwa Petunjuk Praktikum ini telah direview dan akan digunakan untuk pelaksanaan praktikum di Semester Gasal Tahun Akademik 2019/2020 di Laboratorium Praktikum Teknik Informatika, Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Yogyakarta, 23 Agustus 2019

Mengetahui,
Ketua Kelompok Keilmuan Rekayasa Perangkat Lunak dan Data (RELATA)



Drs., Tedy Setiadi, M.T.
NIY. 60030475

Kepala Kepala Laboratorium Praktikum
Teknik Informatika



Lisna Zahrotun, S.T., M.Cs.
NIY. 60150773

VISI DAN MISI PRODI TEKNIK INFORMATIKA

VISI

Menjadi Program Studi Informatika yang diakui secara internasional dan unggul dalam bidang Informatika serta berbasis nilai-nilai Islam.

MISI

1. Menjalankan pendidikan sesuai dengan kompetensi bidang Informatika yang diakui nasional dan internasional
2. Meningkatkan penelitian dosen dan mahasiswa dalam bidang Informatika yang kreatif, inovatif dan tepat guna.
3. Meningkatkan kuantitas dan kualitas publikasi ilmiah tingkat nasional dan internasional
4. Melaksanakan dan meningkatkan kegiatan pengabdian masyarakat oleh dosen dan mahasiswa dalam bidang Informatika.
5. Menyelenggarakan aktivitas yang mendukung pengembangan program studi dengan melibatkan dosen dan mahasiswa.
6. Menyelenggarakan kerja sama dengan lembaga tingkat nasional dan internasional.
7. Menciptakan kehidupan Islami di lingkungan program studi.

TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA

DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum dan menandatangani presensi kehadiran praktikum.
2. Dosen membuat modul praktikum, soal seleksi asisten, pre-test, post-test, dan responsi dengan berkoordinasi dengan asisten dan pengampu mata praktikum.
3. Dosen berkoordinasi dengan koordinator asisten praktikum untuk evaluasi praktikum setiap minggu.
4. Dosen menandatangani surat kontrak asisten praktikum dan koordinator asisten praktikum.
5. Dosen yang tidak hadir pada slot praktikum tertentu tanpa pemberitahuan selama 2 minggu berturut-turut mendapat teguran dari Kepala Laboratorium, apabila masih berlanjut 2 minggu berikutnya maka Kepala Laboratorium berhak mengganti koordinator praktikum pada slot tersebut.

PRAKTIKAN

1. Praktikan harus hadir 15 menit sebelum kegiatan praktikum dimulai, dan dispensasi terlambat 15 menit dengan alasan yang jelas (kecuali asisten menentukan lain dan patokan jam adalah jam yang ada di Laboratorium, terlambat lebih dari 15 menit tidak boleh masuk praktikum & dianggap INHAL).
2. Praktikan yang tidak mengikuti praktikum dengan alasan apapun, wajib mengikuti INHAL, maksimal 4 kali praktikum dan jika lebih dari 4 kali maka praktikum dianggap GAGAL.
3. Praktikan harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Praktikan tidak boleh makan dan minum selama kegiatan praktikum berlangsung, harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di dalam laboratorium (tidak boleh membuang sampah sembarangan baik kertas, potongan kertas, bungkus permen baik di lantai karpet maupun di dalam ruang CPU).
5. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Asisten atau Laboran.
6. Praktikan harus meletakkan sepatu dan tas pada rak/loker yang telah disediakan.
7. Selama praktikum dilarang NGENET/NGE-GAME, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
8. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengetahuan laboran
9. Praktikan harus memiliki FILE Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.
10. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.
11. Praktikan dilarang mengubah setting software/hardware komputer baik menambah atau mengurangi tanpa permintaan asisten atau laboran dan melakukan sesuatu yang dapat merugikan laboratorium atau praktikum lain.

12. Asisten, Koordinator Praktikum, Kepala laboratorium dan Laboran mempunyai hak untuk menegur, memperingatkan bahkan meminta praktikan keluar ruang praktikum apabila dirasa anda mengganggu praktikan lain atau tidak melaksanakan kegiatan praktikum sebagaimana mestinya dan atau tidak mematuhi aturan lab yang berlaku.
13. Pelanggaran terhadap salah satu atau lebih dari aturan diatas maka Nilai praktikum pada pertemuan tersebut dianggap 0 (NOL) dengan status INHAL.

ASISTEN PRAKTIKUM

1. Asisten harus hadir 15 Menit sebelum praktikum dimulai (konfirmasi ke koordinator bila mengalami keterlambatan atau berhalangan hadir).
2. Asisten yang tidak bisa hadir WAJIB mencari pengganti, dan melaporkan kepada Koordinator Asisten.
3. Asisten harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Asisten harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di laboratorium, menegur atau mengingatkan jika ada praktikan yang tidak dapat menjaga kebersihan, ketertiban atau kesopanan.
5. Asisten harus dapat merapikan dan mengamankan presensi praktikum, Kartu Nilai serta tertib dalam memasukan/Input nilai secara Online/Offline.
6. Asisten harus dapat bertindak secara profesional sebagai seorang asisten praktikum dan dapat menjadi teladan bagi praktikan.
7. Asisten harus dapat memberikan penjelasan/pemahaman yang dibutuhkan oleh praktikan berkenaan dengan materi praktikum yang diasistensi sehingga praktikan dapat melaksanakan dan mengerjakan tugas praktikum dengan baik dan jelas.
8. Asisten tidak diperkenankan mengobrol sendiri apalagi sampai membuat gaduh.
9. Asisten dimohon mengkoordinasikan untuk meminta praktikan agar mematikan komputer untuk jadwal terakhir dan sudah dilakukan penilaian terhadap hasil kerja praktikan.
10. Asisten wajib untuk mematikan LCD Projector dan komputer asisten/praktikan apabila tidak digunakan.
11. Asisten tidak diperkenankan menggunakan akses internet selain untuk kegiatan praktikum, seperti Youtube/Game/Medsos/Streaming Film di komputer praktikan.

LAIN-LAIN

1. Pada Saat Responsi Harus menggunakan Baju Kemeja untuk Laki-laki dan Perempuan untuk Praktikan dan Asisten.
2. Ketidakhadiran praktikum dengan alasan apapun dianggap INHAL.
3. Izin praktikum mengikuti aturan izin SIMERU/KULIAH.
4. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.
5. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat ijin dari Kepala Laboratorium Prodi Teknik Informatika.

Yogyakarta, 23 Agustus 2019

Kepala Laboratorium Praktikum Teknik
Informatika

A handwritten signature in black ink, appearing to read 'Lisna Zahrotun', is written over a light blue rectangular background.

Lisna Zahrotun, S.T., M.Cs.

NIY. 60150773

DAFTAR ISI

KATA PENGANTAR.....	1
DAFTAR PENYUSUN.....	2
HALAMAN REVISI.....	3
HALAMAN PERNYATAAN.....	4
VISI DAN MISI PRODI TEKNIK INFORMATIKA	5
TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA.....	6
DAFTAR ISI.....	9
DAFTAR GAMBAR.....	10
DAFTAR TABEL.....	11
PRAKTIKUM 1: PERANGKAT KERAS KOMPUTER.....	12
PRAKTIKUM 2: RANDOM ACCESS MEMORY.....	15
PRAKTIKUM 3: PEMROGRAMAN ASSEMBLY DENGAN DEBUG	18
PEKAN UJI KOMPETENSI I.....	24
PRAKTIKUM 4. PEMROGRAMAN COMPILER ASSEMBLY DENGAN TASM DAN TLINK.....	25
PRAKTIKUM 5. PEMROGRAMAN ASSEMBLY DENGAN WORD PROCESS	31
PRAKTIKUM 6. ARITMATIKA.....	36
PEKAN UJI KOMPETENSI II.....	41
PRAKTIKUM 7. MOUSE.....	42
PRAKTIKUM 8. PEMROGRAMAN ASSEMBLY MENGGUNAKAN EMU8086.....	47
PRAKTIKUM 9. PENELUSURAN PROGRAM ASSEMBLY MENGGUNAKAN EMU8086.....	50
PRAKTIKUM 10. PEMROGRAMAN ASSEMBLY MENGGUNAKAN MICROPROCESSOR SIMULATOR	52
DAFTAR PUSTAKA.....	55

DAFTAR GAMBAR

Gambar 1.1. Komponen sistem komputer	13
Gambar 1.2. Penampakan isi komputer (PC)	13
Gambar 2.1. Random Access Memory (RAM)	15
Gambar 2.2. Tampilan hasil dari Task Manager	16
Gambar 3.1. Macam-macam Register	18
Gambar 3.2. Tampilan perintah-perintah Debug	20
Gambar 3.3. Tampilan trace instruksi	21
Gambar 3.4. Tampilan penyimpanan program ke dalam disk	21
Gambar 4.1. Tampilan program assembly latihan1	27
Gambar 8.1. Tampilan emu8086	46
Gambar 9.1. Fasilitas debu di emu8-86	49
Gambar 10.1. Tampilan antarmuka Microprocessor Simulator	51
Gambar 10.2. Proses simulasi untuk program lift.asm	52

DAFTAR TABEL

Tabel 5.1. Karakter control yang sering digunakan.....	30
Tabel 5.2. Karakter control pada ASCII.....	30
Tabel 7.1. Tabel bitmap kursor	42

PRAKTIKUM 1: PERANGKAT KERAS KOMPUTER

Pertemuan ke	: 1
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

1.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu mengetahui teknologi penyusun (komponen) perangkat keras komputer
2. mampu menjelaskan fungsi perangkat input dan output serta peripheral lainnya

Indikator ketercapaian diukur dengan:

1. mampu memahami bagian komponen perangkat keras komputer
2. mampu menjelaskan kembali bagian-bagian komputer dan fungsinya masing-masing

1.2. TEORI PENDUKUNG

Komputer merupakan alat yang digunakan untuk mengolah data berdasarkan prosedur yang telah dirumuskan. Komputer memiliki kemampuan untuk menyimpan, menerima, dan mengolah data. Kata komputer semula dipergunakan untuk menggambarkan orang yang perkerjaannya melakukan perhitungan aritmatika, dengan atau tanpa alat bantu, tetapi arti kata ini kemudian dipindahkan kepada mesin itu sendiri.

Komputer terdiri dari perangkat keras dan perangkat lunak. Kolaborasi antara komputer dengan manusia sebagai pengendali (pembuat program dan pengoperasiannya) Sistem Komputer adalah elemen-elemen yang terkait untuk menjalankan suatu aktifitas dengan menggunakan komputer. Elemen dari sistem komputer terdiri dari manusianya (*brainware*), perangkat lunak (*software*), set instruksi (*instruction set*), dan perangkat keras (*hardware*).

Hardware (perangkat keras) merupakan perangkat komputer yang berwujud fisik pada komputer yang digunakan oleh sistem untuk menjalankan perintah yang telah diprogramkan atau dalam arti singkatnya sebuah komponen pada komputer yang bisa disentuh, dilihat dan diraba.

Software (Perangkat lunak) merupakan program yang berjalan di perangkat keras yang berfungsi untuk mengendalikan komponen perangkat keras untuk melakukan tugas-tugas tertentu.

Brainware (perangkat otak) ini merupakan orang yang membuat program komputer atau menggunakan komputer tersebut.

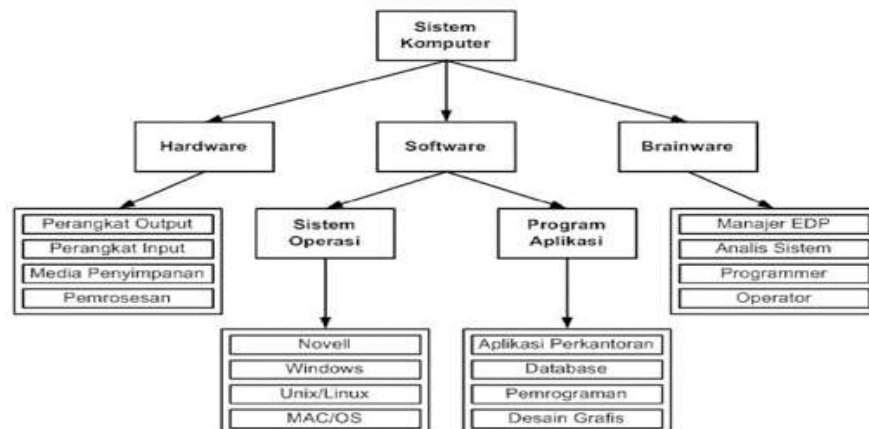
Secara umum perangkat keras komputer dibagi menjadi beberapa kelompok yaitu: peralatan input, peralatan output, perangkat pemroses, dan perangkat penyimpanan.

Peralatan input adalah perangkat keras komputer yang berfungsi sebagai alat untuk memasukkan data atau perintah ke dalam komputer. Peralatan input antara lain keyboard, mouse, joystick, scanner, dll.

Peralatan output adalah perangkat keras komputer yang berfungsi untuk menampilkan keluaran sebagai hasil pengolahan data. Keluaran dapat berupa hard-copy (ke kertas), soft-copy (ke monitor), ataupun berupa suara. Peralatan output antara lain monitor, printer, plotter, speaker, dll.

Perangkat pemroses berupa CPU (Central Processing Unit) dan perangkat pendukungnya.

Perangkat penyimpanan, merupakan perangkat yang digunakan untuk menyimpan data dan informasi yang digunakan oleh komputer maupun data program aplikasi pengguna. Perangkat penyimpanan ini antara lain RAM, Harddisk, CDRom, dll.



Gambar 1.1 Komponen sistem komputer

1.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Kamera

1.4. LANGKAH PRAKTIKUM

Praktikum pertama ini adalah mengamati komponen-komponen penyusun perangkat keras sebuah komputer. Pengamatan bisa dilakukan secara kelompok atau klasikal dengan satu objek yang dipandu oleh yang berkompeten.

1. Siapkan sebuah komputer (PC) khusus atau yang tidak digunakan untuk operasional.
2. Hidupkan komputer lakukan pengecekan BIOS (untuk mendata informasi perangkat keras yang ditampilkan: RAM, Prosesor, dll)
3. Ambil gambar dari tampilan informasi perangkat keras tersebut dengan kamera
4. Matikan komputer



Gambar 1.2. Penampakan isi komputer (PC)

5. Lakukan pembukaan/pembongkaran tutup casing PC tersebut, sehingga nampak bagian-bagian perangkat keras komputer
6. Amati tiap-tiap komponen yang nampak
7. Ambillah gambar dari penampakan isi komputer tersebut dengan kamera sehingga jelas nampak bagian-bagian demi bagian komponen tersebut.
8. Lakukan pelepasan beberapa komponen yang memang bisa dilepas-pasang, untuk dicatat informasi yang ada.
9. Tiap praktikan wajib mengambil gambar secara langsung, dan tidak boleh copas.

1.5. TUGAS

Berdasarkan hasil dokumentasi kamera lakukanlah beberapa hal:

1. Gambar hasil pemotretan dicetak dan diberi nomor di tiap komponen lalu dijelaskan tiap nomor sesuai nama masing-masing.
2. Buatlah tabel daftar komponen yang terdapat dalam perangkat komputer yang telah diamati dan dikelompokkan berdasarkan perangkat input

PRAKTIKUM 2: RANDOM ACCESS MEMORY

Pertemuan ke	: 2
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

2.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu menjelaskan tentang RAM
2. mampu menjelaskan kegunaan RAM
3. mampu menjelaskan penerapan RAM pada program komputer

Indikator ketercapaian diukur dengan:

1. kemampuan menjelaskan tentang RAM
2. kemampuan menjelaskan kegunaan RAM
3. kemampuan menjelaskan penerapan RAM pada komputer

2.2. TEORI PENDUKUNG

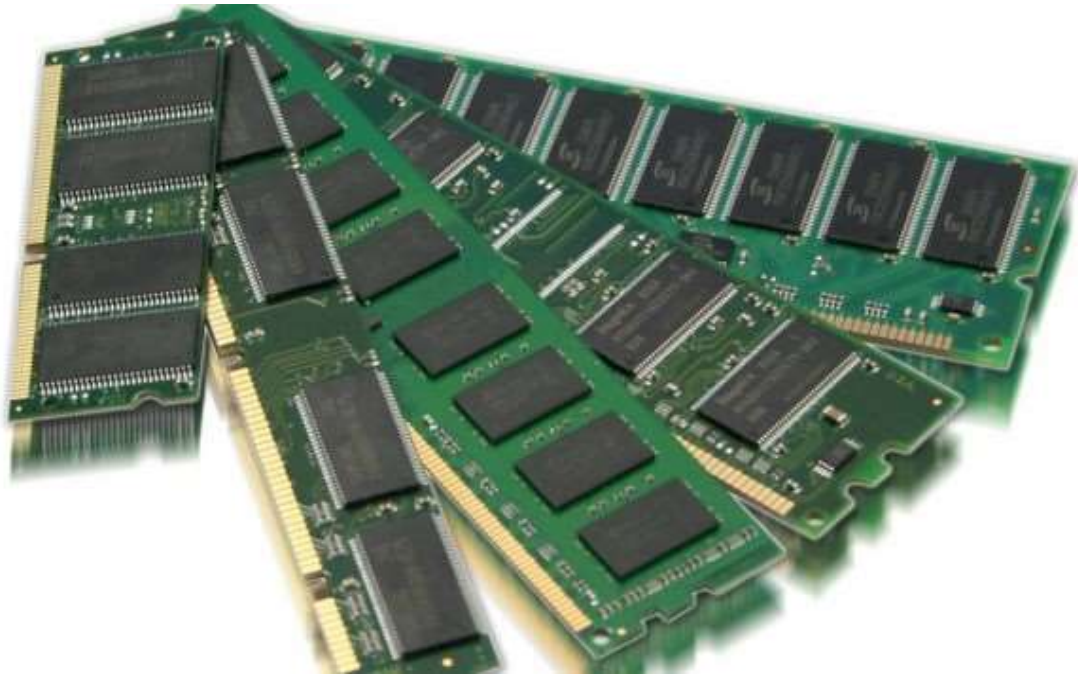
RAM (*Random Access Memory*) adalah jenis penyimpanan data komputer. Perangkat RAM memungkinkan untuk mengakses data dalam urutan acak, yang membuatnya sangat cepat untuk menemukan sepotong informasi tertentu. Jenis penyimpanan tertentu lainnya bukan akses acak. Misalnya, hard disk drive dan CD akan membaca dan menulis data dalam urutan yang telah ditentukan. Rancangan mekanis dari perangkat ini menentukan bahwa akses data secara berurutan. Ini berarti bahwa waktu yang diperlukan untuk menemukan bagian informasi tertentu dapat sangat bervariasi tergantung dimana letaknya pada disk.

RAM (*Random Access Memory*) digunakan dalam sistem komputer sebagai memori utama. RAM dianggap memori *volatile*, yang berarti bahwa informasi yang disimpan hilang ketika tidak ada daya. Jadi, RAM digunakan oleh central processing unit (CPU) ketika komputer berjalan untuk menyimpan informasi yang perlu digunakan dengan sangat cepat, tetapi tidak menyimpan informasi apa pun secara permanen.

Perangkat RAM saat ini menggunakan sirkuit terpadu untuk menyimpan informasi. Ini adalah bentuk penyimpanan yang relatif mahal dan biaya per unit penyimpanan jauh lebih tinggi daripada perangkat seperti hard drive. Namun, waktu untuk mengakses data jauh lebih cepat untuk RAM yang

kecepatannya melebihi biaya. Oleh karena itu, Komputer menggunakan sejumlah RAM untuk akses cepat, penyimpanan sementara informasi dan jumlah penyimpanan massal permanen yang tidak acak, seperti hard disk drive. Sebagai contoh, sistem komputer yang khas mungkin memiliki dua hingga delapan GB (gigabyte) RAM, sementara kapasitas penyimpanan hard disk drive bisa beberapa ratus GB atau bahkan satu TB (terabyte).

RAM memungkinkan data untuk dibaca atau ditulis dalam jumlah waktu yang hampir sama terlepas dari lokasi fisik data di dalam memori. Beberapa jenis RAM, yang memiliki Kecepatan, konsumsi daya dan teknologi yang berbeda. Gambar 1 menunjukkan contoh dari RAM pada komputer.



Gambar 2. 1. Random Access Memory (RAM)

2.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Windows OS.
3. Task Manager.

2.4. LANGKAH PRAKTIKUM

1. Buka Microsoft Word.
2. Tuliskan beberapa kata atau kalimat pada Microsoft Word.
3. Buka Task Manager.
4. Pada Tab "Processes" dapat dilihat bahwa pada kolom "Memory" menunjukkan jumlah memori yang digunakan oleh Sistem Komputer untuk menyimpan informasi sementara berupa sejumlah kata pada aplikasi Microsoft Word.

The screenshot shows the Windows Task Manager Performance tab. At the top, it displays overall system usage: CPU at 41%, Memory at 65%, Disk at 5%, and Network at 0%. Below this, a table lists various processes with their respective resource usage. The 'Snipping Tool' process is highlighted in blue.

Name	Status	41% CPU	65% Memory	5% Disk	0% Network
IrfanView 64-bit		0%	2,8 MB	0 MB/s	0 Mbps
Microsoft Excel		0%	46,7 MB	0 MB/s	0 Mbps
Microsoft Project		0%	49,9 MB	0 MB/s	0 Mbps
Microsoft Word (2)		0,1%	245,4 MB	0 MB/s	0 Mbps
Snipping Tool		0,2%	10,1 MB	0 MB/s	0 Mbps
Task Manager		0,4%	23,1 MB	0,1 MB/s	0 Mbps
Background processes (69)					
64-bit Synaptics Pointing Enhanc...		0%	0,7 MB	0 MB/s	0 Mbps
adb (32 bit)		0%	1,6 MB	0 MB/s	0 Mbps
Antimalware Service Executable		0,1%	146,1 MB	0 MB/s	0 Mbps
Application Frame Host		0%	3,4 MB	0 MB/s	0 Mbps
Casting protocol connection list...		0%	2,1 MB	0 MB/s	0 Mbps
COM Surrogate		0%	1,6 MB	0 MB/s	0 Mbps
COM Surrogate		0%	1,7 MB	0 MB/s	0 Mbps
COM Surrogate		0%	1,6 MB	0 MB/s	0 Mbps

Gambar 2.2 Tampilan hasil dari Task Manager

5. Dengan menyimpan data secara sementara pada RAM maka program dapat berjalan dengan lebih cepat dan responsive. Mengingat sifat penyimpanan pada RAM adalah sementara, jangan lupa untuk menyimpan data dalam format yang permanen karena kejadian yang tidak diinginkan misalnya listrik mati atau hal-hal lain bisa menyebabkan data hilang dan tidak bisa diakses kembali.

2.5. TUGAS

Berdasarkan pengamatan *Task Manager* lakukan analisis terhadap penggunaan memory, dibuat dalam bentuk tabel penggunaan dan grafik perbandingannya.

PRAKTIKUM 3: PEMROGRAMAN ASSEMBLY DENGAN DEBUG

Pertemuan ke	: 3
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

3.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu menjelaskan perintah dalam debug
2. mampu membuat program assembly
3. mampu menerapkan debug pada program assembly

Indikator ketercapaian diukur dengan:

1. menulis perintah debug
2. membuat baris perintah dengan program assembly sederhana

3.2. TEORI PENDUKUNG

Dalam mempelajari bahasa assembly tidak lepas dari register, karena dalam pemrograman bahasa assembly selalu terhubung dengan register yang digunakan dalam pemrograman untuk melakukan operasi-operasi dan instruksi-instruksi yang diperintahkan pada bahasa assembly.

Register adalah sebagian kecil memory CPU yang dipakai untuk tempat penampungan data. Data yang disimpan dalam register akan diproses dalam berbagai operasi. Besarnya data disesuaikan dengan daya tampung register. Secara umum register dapat dibagi dalam lima golongan yaitu:

1. General Purpose Register.

Register yg digunakan untuk keperluan umum pemrograman, keperluan umum yang dimaksud yaitu melakukan perhitungan aritmetika dan perpindahan data. Register ini terdiri atas:

- a. Register AX
Register AX merupakan register 16 bit sebagai register aritmatika atau Accumulator Register. Register AX selalu dipakai dalam operasi penjumlahan, pengurangan, perkalian dan pembagian. Register AX terdiri dari dua buah register 8 bit, yaitu AH dan AL.
- b. Register BX
Register BX merupakan register 16 bit sebagai register base addressing mode, yaitu berfungsi mengambil atau menulis data langsung dari/ke memory disebut Base Address Register. Register BX terdiri dari dua buah register 8 bit yaitu BH dan BL.
- c. Register CX

Register CX merupakan register 16 bit sebagai suatu counter untuk meletakkan jumlah lompatan pada loop-loop yang anda lakukan, disebut Count Register. Register CX terdiri dari dua buah register 8 bit yaitu CH dan CL.

d. Register DX

Disebut juga Data Register. Register ini terdiri dari dua buah register 8 bit yaitu DH dan DL. Register DX merupakan register 16 bit yang mempunyai tugas:

- Membantu AX dalam operasi perkalian pembagian
- DX merupakan register offset dari DS
- Menunjukkan nomor port pada operasi port.

2. Segment Register

Segment Register membentuk alamat memori bagi suatu data, terdapat pada 2 operasi (Real Mode dan Protected Real), berikut register-register pada segment register:

- a. Register CS : Code Segment Register
- b. Register DS : Data Segment Register
- c. Register SS : Stack Segment Register
- d. Register ES : Extra Segment Register

3. Point Register

Pointer Register adalah register yg digunakan untuk menunjukkan alamat sebuah data di lokasi memori. Penujukan pada saat operasi perpindahan data, operasi stack (PUSH dan POP), akan dieksekusi pada saat proram dijalankan. Pointer register terdiri atas:

- a. Instruction Point (IP) Register
Instruction Pointer berfungsi sebagai tempat menyimpan alamat dari kode yang akan dieksekusi selanjutnya oleh mikroprosesor. Dalam kerjanya IP bekerja dengan code segmen (CS) untuk menghasilkan alamat relative dari suatu instruksi.
- b. Stack Point (SP) Register
- c. Base Point (BP) Register

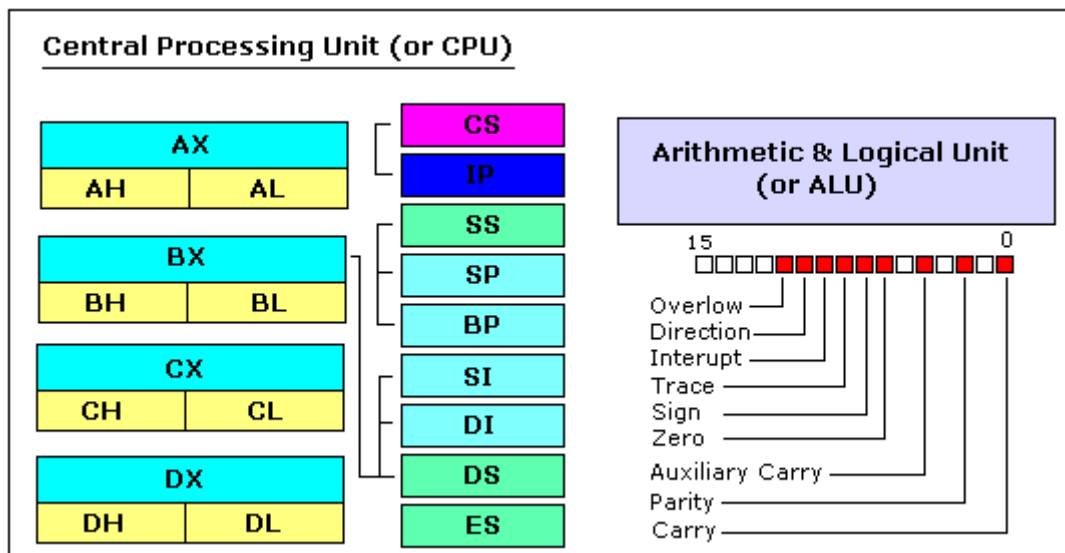
4. Index Register

Index Register digunakan untuk menunjukkan alamat sebuah data di lokasi memori pada operasi-operasi string. Register ini juga digunakan untuk menunjukkan data berindek (data tipe larik) di dalam memori, Register ini terdiri atas :

- a. Source Index (SI) Register
- b. Destination Index (DI) Register

5. Flag Register

Flag Register berfungsi untuk menunjukan status (Keadaan) sesaat dari mikroprocecor. Register ini terdiri atas : overflow, direction, overflow, zero, carry, parity, auxiliary, sign, trap dan Interrupt.



Gambar 3.1. Macam-macam Register

Perintah-Perintah Debug:

- A : Merakit instruksi simbolik (kode mesin)
- D : Menampilkan isi suatu daerah memori
- E : Memasukan data ke memori yang dimulai pada lokasi tertentu
- G : Run executable program ke memori
- N : Memberikan Nama Program
- P : Eksekusi sekumpulan instruksi yang terkait
- Q : Quit
- R : Menampilkan isi satu atau lebih register
- T : Trace isi sebuah Instruksi
- U : Unassembled kode mesin ke kode simbolik
- W : Menulis program ke disk

Perintah Dasar Assembly:

1. MOV.

Perintah MOV adalah perintah untuk mengisi, memindahkan, memperbarui isi suatu register, variable ataupun lokasi memory. Adapun tata penulisan perintah MOV adalah :

```
MOV [operand A], [Operand B]
```

Contoh :

```
MOV AH,02
```

Operand A adalah Register AH Operand B adalah bilangan 02

Hal yang dilakukan oleh komputer untuk perintah di atas adalah memasukan 02 ke register AH.

2. INT.

Perintah INT memanggil subroutine (sub program kecil) yang telah disediakan oleh memory komputer. INT ada dua 2 jenis yaitu:

- a. Interrupt 00h – 1Fh(0 - 31) merupakan interrupt BIOS dan standar di semua komputer baik yang menggunakan sistem operasi DOS maupun bukan yang menggunakan sistem operasi DOS. Lokasi Interrupt Vector Table-nya ada di alamat absolute 0000h-007Fh.

- b. Interrupt 20h–FFh(32-255) merupakan interrupt DOS. Interrupt ini hanya ada pada komputer yang menggunakan sistem operasi DOS dan interrupt handler-nya diproses ke memori oleh DOS pada saat DOS digunakan. Lokasi Interrupt Vector Table-nya ada di alamat absolute 07h-3FFh.

3.3. ALAT DAN BAHAN

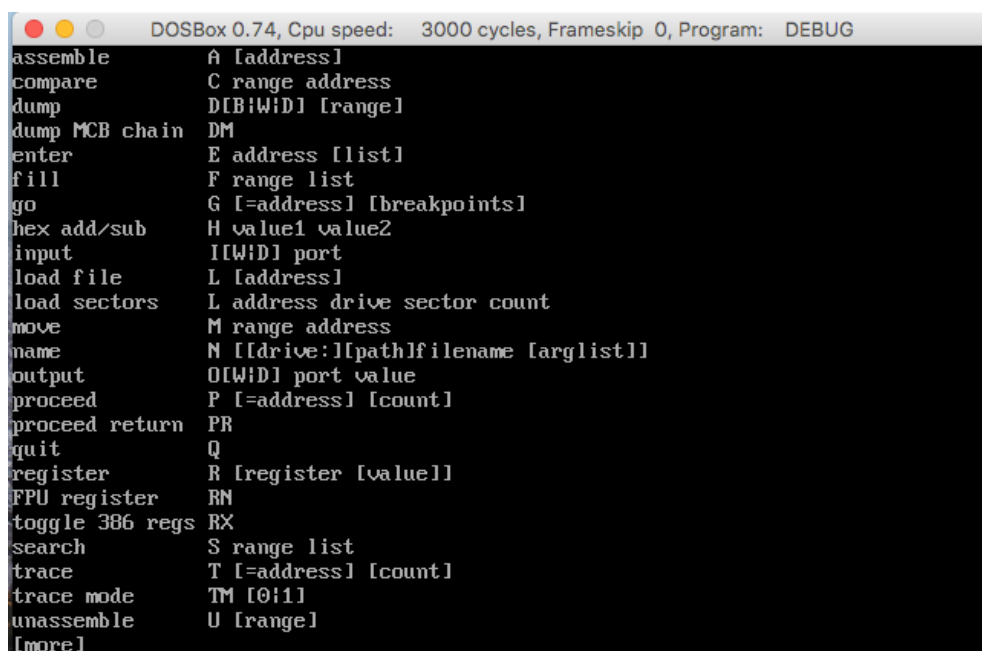
Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Windows OS.
3. Command Prompt.

3.4. LANGKAH PRAKTIKUM

Latihan 1.

1. Buka Command Prompt.
2. Ketik "DEBUG" pada halaman Command Prompt kemudian tekan enter.
3. Untuk mengetahui list perintah pada DEBUG, Ketikkan "?" lalu tekan enter.



```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
assemble      A [address]
compare       C range address
dump          D[BiWiD] [range]
dump MCB chain DM
enter         E address [list]
fill         F range list
go           G [=address] [breakpoints]
hex add/sub   H value1 value2
input        I[WiD] port
load file     L [address]
load sectors  L address drive sector count
move         M range address
name         N [[drive:][path]filename [arglist]]
output       O[WiD] port value
proceed      P [=address] [count]
proceed return PR
quit         Q
register      R [register [value]]
FPU register  RN
toggle 386 regs RX
search       S range list
trace        T [=address] [count]
trace mode   TM [0:1]
unassemble   U [range]
[more]

```

Gambar 3.2. Tampilan perintah-perintah Debug

4. Ketikkan "A 100" lalu tekan enter.
5. Selanjutnya ketikkan program di bawah ini:

```

Mov ax,05
Mov ds,ax
Mov si,012
Mov di,021
Mov ax,si
Mov di,ax
Int 20

```

6. Untuk melihat perubahan Register terhadap instruksi yang diberikan program di atas tekan enter, ketikkan "t" dan kemudian tekan enter.

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Z:\>C:
C:\>DEBUG.COM
-A 100
06B0:0100 mov ax, 05
06B0:0103 mov ds, ax
06B0:0105 mov si, 012
06B0:0108 mov di, 021
06B0:010B mov ax,si
06B0:010D mov di,ax
06B0:010F int 20
06B0:0111
-t
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0103 NU UP EI PL NZ NA PO NC
06B0:0103 BED8          MDU      DS,AX
-t
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=0105 NU UP EI PL NZ NA PO NC
06B0:0105 BE1200       MDU      SI,0012
-t
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0000
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=0108 NU UP EI PL NZ NA PO NC
06B0:0108 BF2100       MDU      DI,0021
-t
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0021
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=010B NU UP EI PL NZ NA PO NC
06B0:010B 89F0          MDU      AX,SI
-t
AX=0012 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0021
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=010D NU UP EI PL NZ NA PO NC
06B0:010D 89C7          MDU      DI,AX
-t
AX=0012 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0012
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=010F NU UP EI PL NZ NA PO NC
06B0:010F CD20          INT      20
-

```

Gambar 3.3. Tampilan trace instruksi

7. Untuk menyimpan program ke disk dengan cara mengetikkan huruf "N [nama_program].com" kemudian Ketikkan "W" lalu tekan Enter.

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
06B0:0103 BED8          MDU      DS,AX
-t
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=0105 NU UP EI PL NZ NA PO NC
06B0:0105 BE1200       MDU      SI,0012
-t
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0000
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=0108 NU UP EI PL NZ NA PO NC
06B0:0108 BF2100       MDU      DI,0021
-t
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0021
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=010B NU UP EI PL NZ NA PO NC
06B0:010B 89F0          MDU      AX,SI
-t
AX=0012 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0021
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=010D NU UP EI PL NZ NA PO NC
06B0:010D 89C7          MDU      DI,AX
-t
AX=0012 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0012 DI=0012
DS=0005 ES=06B0 SS=06B0 CS=06B0 IP=010F NU UP EI PL NZ NA PO NC
06B0:010F CD20          INT      20
-n coba.com
-w
Writing 0000 bytes
-

```

Gambar 3.4. Tampilan penyimpanan program ke dalam disk

Latihan 2.

1. Latihan menulis, menghitung panjang program, memberi nama, menyimpan program, memberi nama dan menjalankan program.
2. Ketikkan "A 100" lalu tekan enter.
3. Selanjutnya ketikkan program di bawah ini:

```
Mov ax,02
Mov dl,41
Int 20
Int 21
```

4. Untuk menentukan Panjang program tekan enter, kemudian ketikkan "RCX" lalu isi angka "8" karena panjangnya program 8 bit, diperoleh dari angka alamat A0108 - A0100 = 8.
5. Untuk menyimpan program, ketikkan huruf "N [nama_program].com" kemudian Ketikkan "W" lalu tekan Enter.
6. Untuk menjalankan program ketikkan huruf "G" lalu tekan enter, Maka akan terlihat hasil berupa huruf "A".
7. Ketikkan "Q" untuk keluar dari DEBUG lalu tekan enter.
8. Untuk mengetahui Panjang program, ketikkan "dir nama_program.com" lalu tekan enter.

Latihan 3.

1. Ketikkan "A 100" lalu tekan enter.
2. Selanjutnya ketikkan program di bawah ini:

```
Mov ah,02
Mov dl,31
Int 20
Int 21
```

```
N nama_program.com
W
G
```

3. Hasilnya berupa angka "1"

3.5. TUGAS

1. Analisis Program pada Latihan 1, 2, dan 3 dengan menggunakan trace. Jelaskan perubahan yang terjadi!
2. Ubahlah Latihan 1 dengan menukar data register DI ke Register SI, kemudian hapus data pada Register AX.

PEKAN UJI KOMPETENSI I

PRAKTIKUM 4. PEMROGRAMAN COMPILER ASSEMBLY DENGAN TASM DAN TLINK

Pertemuan ke	: 4
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

4.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu menjelaskan tentang konsep tanda *directive*.
2. mampu menjelaskan tentang konsep assembly compiler Tasm dan Tlink.
3. mampu menguasai assembly compiler Tasm dan Tlink.
4. mampu menguasai penulisan program assembly dengan text editor.
5. mampu meng-compile program asm menjadi com dan exe.

Indikator ketercapaian diukur dengan:

1. kemampuan membuat program assembly dengan TASM dan TLINK
2. menghasilkan program sederhana bebas error

4.2. TEORI PENDUKUNG

Program assembly adalah kumpulan dari baris-baris text program. Baris-baris program ini dapat ditulis dengan text editor secara umum seperti text editor dari window maupun text editor khusus mengedit program seperti note++.

Struktur Program Assembly

Struktur program adalah aturan penulisan program agar nantinya dapat diterima dan dapat di-compile. Struktur program assembly ini akan compile dengan TASM dan TLINK.

```

-----
.MODEL SMALL
.CODE
ORG 100H
Label1 : JMP Label2
+-----+
| TEMPAT DATA PROGRAM |
+-----+

```

```

Label2 : +-----+
| TEMPAT PROGRAM |
+-----+
INT 20H
END Label1
-----

```

Tanda Directive

1. .MODEL. Model-model pada program dan codenya:

- a. TINY, program hanya akan menggunakan 1 segment seperti program COM, model ini disediakan untuk program COM.
- b. SMALL, data dan code yang digunakan oleh program kurang dari ukuran 1 segment atau 64 KB.
- c. MEDIUM, data yang digunakan oleh program kurang dari 64 KB tetapi code yang digunakan bisa lebih dari 64 KB.
- d. COMPACT, data yang digunakan bisa lebih besar dari 64 KB tetapi codenya kurang dari 64 KB.
- e. LARGE, data dan code yang dipakai oleh program bisa lebih dari 64 KB.
- f. HUGE, data maupun code array yang digunakan bisa lebih dari 64 KB.

2. .CODE

Tanda directive yang digunakan untuk memberikan pemberitahuan mengenai code segmentnya. Code segment ini digunakan untuk menyimpan program yang akan dijalankan.

3. .ORG 100h

Tanda directive ini sering digunakan pada program COM. Perintah ini digunakan untuk memberitahukan kepada assembler supaya program pada saat dijalankan ditaruh mulai pada offset ke 100h (256) byte. Dapat diartikan juga bahwa kita menyediakan 100h byte kosong pada saat program dijalankan. Program kosong ini nantinya akan ditempati oleh PSP (*Program Segment Prefix*).

4. .JMP

Perintah JMP digunakan untuk melompat menuju tempat yang ditunjukkan oleh perintah JUMP.

Syntax:

```
JUMP Tujuan
```

5. INT 20h

Perintah ini digunakan untuk mengakhiri program dan menyerahkannya kembali ke DOS.

Meng-compile program assembly dengan TASM dan TLINK

Ada dua tahap untuk meng-compile program assembly, pertama adalah meng-assemble dengan TASM. Menga-assemble artinya mengubah text program menjadi data objek berupa data biner. Caranya sebagai berikut:

1. Menggunakan perintah TASM diikuti nama_program.asm

```
TASM <spasi> file.asm
```

Contoh

```
TASM prog0201.asm <enter>
```

Jika proses assembly ini tidak ada pesan error, maka proses assembly berhasil, dan menghasilkan file OBJ, yaitu prog01.obj

2. Melakukan link agar data biner tersebut dapat dieksekusi menjadi bertipe COM atau EXE dengan cara:

```
TLINK prog01.obj -t
```

Atau

```
TLINK prog01.obj/t
```

Maka akan diperoleh file prog01.map dan prog01.com. File yang dapat dieksekusi adalah yang bertipe com, yaitu prog01.com

3. Mengeksekusi program yaitu dengan memanggil program bertipe com, yaitu prog01.com. Caranya

```
Prog01.com <enter>
```

Atau

```
Prog01 <enter>
```

Demikian penjelasan tentang struktur program assembly dan tahap kompilasi dengan TASM dan TLINK. File OBJ dan MAP adalah file garbage (sampah) yang selanjutnya harus dihapus.

Batch File

Untuk menyingkat proses kompilasi dapat dibuat batch file yang berisi beberapa perintah yang akan dilaksanakan secara otomatis.

1. Buka editor, tulis baris-baris perintah berikut:

```
@echo off cls
echo Sedang meng-compile file ASM menjadi COM
echo -----
---
echo oleh Taufiq Ismail

tasm %1
tlink %1 -t

del %1.map del %1.obj

echo Selesai...
echo Jika tidak ada pesan error, coba eksekusi program tersebut
```

2. Simpan dengan nama file ASM2COM.BAT
3. Jalankan dengan menggunakan perintah berikut:

```
Asm2com <spasi> prog01 <enter>
```

Perhatikan, tipe file .ASM tidak disertakan dalam perintah tersebut. Selamat berlatih dan mencoba

Intrupsi (Mencetak Karakter Dan Kalimat)

Intrupsi adalah permintaan khusus kepada mikroprosesor untuk melakukan sesuatu. Mikroprosesor akan menghentikan dahulu apa sedang dikerjakan dan melayani permintaan khusus tersebut, bila terjadi intrupsi. Ada 2 jenis intrupsi yaitu intrupsi vector (yang dimiliki mikroprosesor) dan instruksi intrupsi (dimiliki PC).

Ada banyak intrupsi yg dimiliki oleh mikroprosesor dan PC tapi pada praktikum ini hanya terfokus pada intrupsi 20 dan 21.

1. Intrupsi 20 hanya memiliki 1 fungsi layanan yaitu untuk menghentikan program
2. Intrupsi 21 terdiri atas 2 layanan (*service*) yaitu:
 - a. *Service* nomor 02h berfungsi mencetak karakter ASCII
 - b. *Service* nomor 09h berfungsi mencetak String ASCII.

Untuk memanggil layanan nomor intrupsi harus dimasukan dulu ke register AH sebelum INT.

4.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. TASM
3. Text Editor (Notepad/Notepad++).
4. Command Prompt.

4.4. LANGKAH PRAKTIKUM

Latihan 1.

1. Buka Text Editor (Notepad).
2. Ketikkan program di bawah ini:

```
.MODEL SMALL
.CODE
ORG 100h
MULAI:
    jmp CETAK
    Hello DB 'SELAMAT DATANG '
           DB 'DI BHS ASSEMBLY'
           DB '$'

CETAK :
    MOV AH,09H
    MOV DX,OFFSET Hello
    INT 21H
HABIS:
    INT 20h
END MULAI
```

3. Simpan Program yang anda ketik di atas dengan nama program Hello.asm.
4. Gunakan perintah tasm untuk mengubah ke file objek.
C:/tasm>tasm Hello.asm
5. Gunakan perintah tlink untuk merubah ke file Com.
C:/tasm>tlink Hello.obj/t
6. Ketikkan nama program untuk menjalankan program comnya
C:/tasm>Hello
7. Hasil program berupa: SELAMAT DATANG DI BAHASA ASSEMBLY

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 17 D1 H5 T6
Z:\>MOUNT C ~/DOCUMENTS/DOSBOX
Drive C is mounted as local directory /Users/zaonly/DOCUMENTS/DOSBOX/
Z:\>C:
C:\>TASM HELLO.ASM
Turbo Assembler Version 2.0 Copyright (c) 1988, 1990 Borland International
Assembling file: HELLO.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 492k
C:\>TLINK HELLO.OBJ/T
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
C:\>HELLO
SELAMAT DATANG DI BHS ASSEMBLY
C:\>

```

Gambar 4.1. Tampilan hasil program assembly latihan 1.

Latihan 2.

1. Ketikkan program di bawah ini:

```

.MODEL SMALL
.CODE
ORG 100h
Proses :
    MOV AH,02h
    MOV DL,'A'
    INT 21h
    INT 20h
END Proses

```

2. Simpan Program yang anda ketik di atas dengan nama program Latihan1.asm.
3. Gunakan perintah TASM dan TLINK.
4. Jalankan program.

Latihan 3.

1. Ketikkan program di bawah ini:

```

.MODEL SMALL
.CODE
ORG 100h
Satu:
    JMP Dua
Kal1 DB 'Saya Lagi Belajar'
      DB 'Bahasa Assembly $'
Kal2 DB 'Ternyata...'
      DB 'Asik...'
      DB '$'
Dua :
    Mov AH,09h
    Lea DX,Kal1
    int 21h

```

```
Tiga :  
    Mov DX,Offset Kal2 int 21h  
    int 20h  
End Satu
```

2. Simpan Program yang anda ketik di atas dengan nama program `Latihan2.asm`.
3. Gunakan perintah TASM dan TLINK.
4. Jalankan program.

4.5. TUGAS

1. Dengan menggunakan Latihan 1 sebagai Referensi, Buatlah program untuk menampilkan karakter 'K'.
2. Dengan menggunakan Latihan 2 sebagai Referensi, Buatlah program untuk menampilkan karakter 'PRODI INFORMATIKA'.

PRAKTIKUM 5. PEMROGRAMAN ASSEMBLY DENGAN WORD PROCESS

Pertemuan ke	: 5
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

5.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu menjelaskan tentang konsep word process.
2. mampu menerapkan pemrograman assembly dengan word process.

Indikator ketercapaian diukur dengan:

1. kemampuan menjelaskan tentang konsep word process.
2. Kemampuan menerapkan pemrograman assembly dengan word process.

5.2. TEORI PENDUKUNG

Dalam pemrograman assembler, interupsi merupakan hal yang penting untuk menghasilkan sesuatu yang diinginkan oleh user sehingga dikenali oleh komputer. Interupsi ini tidak berjalan sendiri, interupsi selalu berjalan berbarengan dengan service yang menyediakan fungsi-fungsi yang berbeda untuk setiap interupsi yang sama tetapi dengan service yang berbeda-beda.

Dalam mencetak kalimat dengan fungsi DOS, interupsi yang digunakan adalah interrupt 21h dengan service 09h. Pemakaian interrupt 21h dengan service 09h ini dapat digunakan untuk mencetak kalimat dengan aturan:

```
INPUT
AH = 9
DS:DX = Alamat String ($) yang diinginkan
```

Catatan bahwa karakter \$ dijadikan tanda akhir dalam tulisan. Khusus untuk mencetak kalimat ini, \$ merupakan akhir pada kalimat yang sering digunakan untuk mengakhiri pendefinisian dari sebuah string, sehingga kalimat tersebut akan dicetak sampai tanda \$. Karakter \$ dijadikan sebagai tanda akhir pada string sehingga string yang akan dicetak adalah string yang berada disebelum tanda \$ dan selainnya yang berada disesudah tanda \$ tidak akan dicetak. Selain tanda \$ yang digunakan untuk mengakhiri kalimat, ada juga tanda 10 dan tanda 13 yang masing-masingnya digunakan untuk pindah baris dan menuju kolom 0 kembali.

Sebuah karakter disertai dengan warna tentunya akan lebih menarik. Untuk itu anda bisa menggunakan interupsi ke 10h dengan aturan pemakaiannya:

```
INPUT
AH = 09h
AL = Kode ASCII dari karakter yang akan dicetak
BH = Nomor halaman (0 untuk halaman 1)
BL = Atribut atau warna dari karakter yang akan dicetak
CX = Banyaknya karakter tersebut akan dicetak
```

Karakter Kontrol merupakan karakter yang digunakan sebagai kontrol dari program khusus mencetak kalimat. Karakter kontrol ini digunakan untuk keperluan komunikasi komputer dengan periferalnya. Karakter kontrol yang sering digunakan dapat dilihat pada Tabel 5.1.

Tabel 5.1. Karakter control yang sering digunakan

Code	Nama	Fungsi
07	Bel	Memberikan suara BEEP
08	Backspace (BS)	Memindahkan kursor 1 kolom ke belakang
09	Horizontal Tab	Memindahkan kursor 8 kolom ke kanan
10	Line Feed (LF)	Memindahkan kursor 1 baris ke bawah
13	Carriage Return (CR)	Memindahkan kursor menuju awal baris

Selain karakter kontrol pada Tabel 5.1, karakter kontrol yang tersedia pada ASCII secara lengkap dapat dilihat pada Tabel 5.2.

Tabel 5.2. Karakter Kontrol pada ASCII

Code	Nama
0	Null
1	Start of heading
2	Start of text
3	End of text
4	End of transmission
5	Enquiry
6	Acknowledge
7	Bel
8	Backspace
9	Horizontal tabulation
10	Line feed
11	Vertical tabulation
12	Form feed
13	Carriage return
14	Shift out
15	Shift in
16	Data link escape
17	Device control
18	Negative acknowledge
19	Synchronous table

20	End of transmission block
21	Cancel
22	End of medium
23	Substitute
24	Escape
25	File separator
26	Group separator
27	Record separator
28	Unit separator
29	Space
30	Delete

5.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. TASM
3. Text Editor.
4. Command Prompt.

5.4. LANGKAH PRAKTIKUM

Latihan 1.

1. Buka Text Editor.
2. Ketikkan code program di bawah ini:

```
.MODEL SMALL
.CODE
ORG 100h
Proses:
    MOV AH,09h      ; Nilai servis untuk mencetak karakter
    MOV AL,'A'      ; AL = Karakter yang akan dicetak
    MOV BH,00h      ; Nomor Halaman layar
    MOV BL,93h      ; Warna atau atribut dari karakter
    MOV CX,03h      ; Banyaknya karakter yang ingin dicetak
    INT 10h         ; Laksanakan!!!

    INT 20h         ; Selesai! kembali ke DOS
END Proses
```

3. Simpan Program yang anda ketik di atas dengan nama_program.asm.
4. Gunakan perintah TASM dan TLINK.
5. Jalankan program.
6. Hasilnya adalah huruf "A" sebanyak 3 kali dengan warna dasar biru kedip dan warna tulisan Cyan.

Latihan 2.

1. Buka Text Editor.
2. Ketikkan code program di bawah ini:

```
.MODEL SMALL
.CODE
ORG 100h
Proses:
```

```

MOV AH,02h      ; Nilai servis
MOV DL,'A'      ; DL=karakter 'A' atau DL=41h
MOV CX,10h      ; Banyaknya pengulangan yang akan
Ulang:
INT 21h         ; Cetak karakter!!
INC DL          ; Tambah DL dengan 1
LOOP Ulang      ; Lompat ke Ulang

INT 20h
END Proses

```

3. Simpan Program yang anda ketik di atas dengan nama_program.asm.
4. Gunakan perintah TASM dan TLINK.
5. Jalankan program.
6. Hasilnya adalah "ABCDEFGHJKLMNOP"

Latihan 3.

1. Buka Text Editor.
2. Ketikkan code program di bawah ini:

```

.model SMALL
.code
    ORG 100h
tdata: jmp proses
        username      db 13,10,'Username : $'
        lpassword     db 13,10,'Password : $'
        lditerima     db 13,10,'diterima $'
        lditolak      db 13,10,'ditolak $'

        ; menyiapkan memori (var) untuk menerima inputan keyboard
        vusername     db 23,?,23 dup(?)
        vpassword     db 23,?,23 dup(?)

proses:
        mov ah,09h          ; tampilkan tulisan username
        lea dx,username
        int 21h

        mov ah,0ah         ; menunggu masukkan keyboard
        lea dx,vusername
        int 21h

        mov ah,09h          ; tampilkan tulisan password
        lea dx,lpassword
        int 21h

        mov ah,0ah
        lea dx,vpassword
        int 21h

```

```

    lea si,vusername
    lea di,vpassword

    cld
    mov cx,23
    rep cmpsb ; repeat while equal & compare string byte
    jne gagal ; jump not equal

    mov ah,09h
    lea dx,lditerima
    int 21h
    jmp exit

gagal :
    mov ah,09h
    lea dx,lditolak
    int 21h
    jmp proses

exit :
    int 20h
end tdata

```

3. Simpan Program yang anda ketik di atas dengan nama_program.asm.
4. Gunakan perintah TASM dan TLINK.
5. Jalankan program

5.5. TUGAS

1. Analisis Program pada Latihan 1, 2, dan 3 dengan menggunakan trace. Jelaskan perubahan yang terjadi!
2. Ubahlah Latihan 3 dengan menukar data lditolak dan lditerima.

PRAKTIKUM 6. ARITMATIKA

Pertemuan ke	: 6
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

6.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu menjelaskan konsep Aritmatika pada pemrograman Assembly.
2. mampu menerapkan penambahan, pengurangan, perkalian, dan pembagian dengan menggunakan Bahasa pemrograman Assembly.

Indikator ketercapaian diukur dengan:

Kemampuan menerapkan penambahan, pengurangan, perkalian, dan pembagian dengan menggunakan Bahasa pemrograman Assembly.

6.2. TEORI PENDUKUNG

Operasi Penambahan

1. 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.
```

Sebagai contohnya :

```
MOV AH,15h    ; AH:=15h
MOV AL,4      ; AL:=4
ADD AH,AL     ; AH:=AH+AL, jadi AH=19h
```

Perlu diperhatikan bahwa pada perintah ADD ini antara Tujuan dan Asal harus mempunyai daya tampung yang sama, misalnya register AH(8 bit) dan AL(8 bit), AX(16 bit) dan BX(16 bit).

2. 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). Pertambahan yang demikian bisa memecahkan masalah seperti yang pernah kita kemukakan, seperti pertambahan pada bilangan 12345678h+9ABCDEF0h. Seperti yang telah kita ketahui bahwa satu register hanya mampu menampung 16 bit, maka untuk pertambahan seperti yang di atas bisa anda gunakan perintah ADC untuk memecahkannya, Contoh:

```
MOV AX,1234h ; AX = 1234h CF = 0
MOV BX,9ABCh ; BX = 9ABCh CF = 0
MOV CX,5678h ; BX = 5678h CF = 0
MOV DX,0DEF0h ; DX = DEF0h CF = 0
ADD CX,DX ; CX = 3568h CF = 1
ADC AX,BX ; AX = AX+BX+CF = ACF1
```

Hasil penjumlahan akan ditampung pada register AX:DX yaitu ACF13568h. Adapun flag-flag yang terpengaruh oleh perintah ADD dan ADC ini adalah CF, PF, AF, ZF, SF dan OF.

3. 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 pemakaiannya 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. Contoh:

```
INC AL ; AL=AL+1.
```

Adapun flag yang terpengaruh oleh perintah ini adalah OF,SF,ZF,AF dan PF.

Operasi Pengurangan

1. SUB

Untuk Operasi pengurangan dapat digunakan perintah SUB dengan syntax:

```
SUB Tujuan,Asal
```

Hasil yang didapat akan ditaruh pada Tujuan, dalam bahasa pascal sama dengan instruksi

Tujuan:=Tujuan-Asal.

Contoh :

```
MOV AX, 15 ; AX:=15
MOV BX, 12 ; BX:=12
SUB AX, BX ; AX:=15-12=3
SUB AX, AX ; AX=0
```

Untuk menolkan suatu register bisa anda kurangkan dengan dirinya sendiri seperti SUB AX,AX

2. 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).

3. 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. Contoh:

```
DEC AL ; AL=AL-1
```

Operasi Perkalian

4. MUL

Untuk perkalian bisa digunakan perintah MUL dengan syntax:

```
MUL Sumber
```

Sumber disini dapat berupa suatu register 8-bit (Misal: BL,BH,...), register 16-bit (Misal: BX,DX,..) atau suatu variabel. 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.

Contoh:

```
.MODEL SMALL
.CODE
ORG 100h
Tdata:
    JMP Proses
    A    DW    01EFh
    B    DW    02FEh
    HslLo DW    ?
    HslHi DW    ?
Proses:
    MOV  AX,A      ; AX=1EF
    MUL  B          ; Kalikan 1EF*2FE
    MOV  HslLo,AX  ; AX bernilai C922 sehingga HslLo=C922
    MOV  HslHi,DX  ; DX bernilai 0005 sehingga HslHi=0005

    INT  20h       ; Kembali ke DOS
END    Tdata
```

Operasi Pembagian**5. 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.

Contoh:

```
.MODEL SMALL
.CODE
ORG 100h
Tdata:
    JMP Proses
    A    DW    01EFh
    B    DW    02FEh
    HslLo DW    ?
    HslHi DW    ?
Proses:
    SUB  DX,DX     ;
    MOV  AX,A      ; AX=1EF
    DIV  B          ; Bagi 1EF:2
    MOV  Hsl,AX    ; AX bernilai 00F7 sehingga Hsl=00F7
    MOV  Sisa,DX   ; DX berisi 0001 sehingga Sisa=0001
```



```

        INT 20h          ; Kembali ke DOS
END      Tdata

```

6.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. TASM
3. Text Editor.
4. Command Prompt.

6.4. LANGKAH PRAKTIKUM

1. Buka Command Prompt.
2. Ketikkan "Debug" lalu tekan Enter.
3. Ketikkan huruf "A100" lalu tekan enter.
4. Selanjutnya ketikkan program di bawah ini:

```

MOV AH,15h
MOV AL,4
ADD AH,AL
MOV AX,1234h
MOV BX,0F221h
ADD AX,BX
MOV AX,1234h
MOV BX,9ABCh
MOV CX,5678h
MOV DX,0DEF0h
ADD CX,DX
ADC AX,BX
INC AL
INT 20h

```

5. Untuk melihat perubahan Register terhadap instruksi yang diberikan program di atas tekan enter, ketikkan "t" dan kemudian tekan enter.
6. Untuk menyimpan program ke disk dengan cara menyetikkan huruf "N [nama_program].com" kemudian Ketikkan "W" lalu tekan Enter.

6.5. TUGAS

1. Analisis Program di atas dengan menggunakan trace. Jelaskan perubahan yang terjadi!
2. Ubahlah program di atas untuk register AH 20 dan register AL 10, dan tentukan hasil register AL terakhir.

PEKAN UJI KOMPETENSI II

PRAKTIKUM 7. MOUSE

Pertemuan ke	: 7
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

7.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu menjelaskan tentang konsep mouse.
2. mampu menerapkan pemrograman assembly dengan mouse.

Indikator ketercapaian diukur dengan:

1. kemampuan membuat program assembly untuk pengendalian mouse

7.2. TEORI PENDUKUNG

Mouse dikendalikan menggunakan fungsi mouse di interupsi 33 jam. Ada banyak fungsi, tetapi bagian ini hanya akan mencakup set dasar yang diperlukan untuk menyelesaikan segala sesuatunya. Untuk informasi lebih lanjut, lihat referensi di halaman web.

Untuk menggunakan mouse, terlebih dahulu memanggil Fungsi 0000h (Reset Driver dan Baca Status). Ini menginisialisasi driver dan perangkat keras. Cursor mouse awalnya akan disembunyikan, jadi harus menggunakan Fungsi 0001h (Tampilkan Mouse Cursor) untuk membuatnya terlihat. Sejak saat itu, cukup hubungi Fungsi 0003h (Posisi Kembali dan Status Tombol) untuk mendapatkan posisi dan status tombol setiap kali program memerlukannya. Pastikan menyembunyikan cursor mouse sebelum program Anda keluar.

Fungsi ini menginisialisasi perangkat keras dan perangkat lunak sehingga mouse siap digunakan. Mouse awalnya akan disembunyikan.

Inputs

AX = 0000h

Outputs

AX = Status

0000h : Error. Hardware/software not installed.

FFFFh : OK. Hardware/software installed.

BX = Number of buttons

```
FFFFh : Two buttons.
0000h : Other than two buttons.
0003h : Three
```

Menampilkan Cursor Mouse

Fungsi ini membuat kursor mouse terlihat di layar. Jika Anda memprogram teks atau grafik dengan menulis langsung ke memori video, Anda harus menyembunyikan kursor mouse sebelum melakukannya untuk menghentikan mouse meninggalkan sampah grafis di layar.

```
Inputs
AX = 0001h
```

Menampilkan Cursor Mouse

Fungsi ini membuat kursor mouse tidak terlihat. Beberapa panggilan ke fungsi ini memerlukan beberapa panggilan ke Fungsi 0001h (Tampilkan Mouse Cursor) sebelum kursor mouse akan muncul lagi, karena driver mouse terus menghitung berapa kali mouse telah disembunyikan.

```
Inputs
AX = 0002h
```

Status Button

Fungsi ini mengembalikan posisi kursor mouse saat ini dan status tombol. Posisi diukur dalam piksel, dengan titik asal (0,0) di sudut kiri atas layar. Dalam mode teks, setiap karakter diasumsikan oleh driver mouse untuk sesuai dengan delapan piksel secara horizontal dan delapan piksel secara vertikal. Jadi, untuk mendapatkan posisi baris dan kolom kursor mouse dalam mode teks, bagilah nilai dalam CX dan DX dengan delapan.

```
Inputs
AX = 0003h
Outputs
BX = Button status (1 = corresponding button pressed)
Bit 0 : Left mouse button.
Bit 1 : Right mouse button.
Bit 2 : Middle mouse button (if present).
Bits 3-15 : Cleared to 0.
CX = Pixel column position.
DX = Pixel row position.
```

Position Mouse Cursor

Fungsi ini akan memposisikan kursor mouse di layar. Seperti dalam fungsi 0003h, posisi diukur dalam piksel, dengan titik asal (0,0) di sudut kiri atas layar. Lihat deskripsi fungsi 0003h untuk informasi lebih lanjut.

```
Inputs
AX = 0004h
CX = Column position
DX = Row position
```

Define Horizontal Cursor Range

Fungsi ini akan membatasi posisi horizontal kursor mouse ke bagian yang ditentukan di layar. Posisi kolom diberikan dalam piksel.

```
Inputs
```

```
AX = 0007h
CX = Batas kolom paling kiri
DX = Batas kolom paling kanan
```

Define Vertical Cursor Range

Fungsi ini akan membatasi posisi vertikal kursor mouse ke bagian yang ditentukan di layar. Posisi baris diberikan dalam piksel.

```
Inputs
AX = 0008h
CX = Batas baris atas
DX = Batas baris bawah
```

Mengubah Kursor Mouse

Fungsi ini akan mendefinisikan ulang tampilan kursor mouse saat layar berada dalam mode grafis.

```
Inputs
AX = 0009h
CX = Kolom hotspot kursor dalam bitmap (-16 hingga 16)
DX = Baris hotspot kursor dalam bitmap (-16 hingga 16)
ES:DX = Pointer ke kursor bitmap
```

Hotspot adalah istilah yang diberikan ke lokasi piksel dalam gambar kursor mouse yang koordinatnya pada layar sama dengan posisi kursor mouse. Pada dasarnya, hot spot ini memungkinkan kita untuk mengetahui dimana seluruh gambar berada di layar relatif terhadap posisi mouse (dikembalikan dalam fungsi 0003h, misalnya). Awalnya, hot spot ada di sudut kiri atas kursor mouse default (panah).

Bitmap kursor dapat berupa gambar 16x16 piksel yang didefinisikan dalam memori sebagai berikut:

Tabel 7.1. Tabel bitmap kursor

Offset	Size	Description
00h	16 words	Screen Mask
20h	16 words	Cursor Mask

Setiap kata mendefinisikan enam belas piksel berturut-turut, dengan pixel paling kanan menjadi bit paling signifikan. Gambar didefinisikan dimulai dengan deretan teratas piksel dalam gambar dengan ukuran sesuai gambar 7.1.

Gambar terbentuk pada layar dengan pertama ANDing piksel pada layar dengan gambar Screen Mask, lalu XOR piksel pada layar dengan gambar Mask Kursor.

7.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. TASM
3. Text Editor
4. Command Prompt.

7.4. LANGKAH PRAKTIKUM

1. Buka Command Prompt.
2. Ketikkan Code Program di bawah ini:


```
.MODEL SMALL
.CODE
```

```

    org 100h

proses:
    jmp start
    oldX dw -1
    oldY dw 0

start:
    ; ubah ke mode grafik
    mov ah, 00
    mov al, 13h          ; layar 256 warna, berukuran 320x200
pixel.
    int 10h             ;

    ; reset mouse status saat ini:
    mov ax, 0
    int 33h
    cmp ax, 0

    ; tampilkan kursor mouse
    mov ax, 1
    int 33h

check_mouse_button:
    mov ax, 3
    int 33h
    shr cx, 1          ; x/2 - in this mode the value of CX is
doubled.
    cmp bx, 1
    jne xor_cursor
    mov al, 1010b      ; pixel color
    jmp draw_pixel

xor_cursor:
    cmp oldX, -1
    je not_required
    push cx
    push dx
    mov cx, oldX
    mov dx, oldY
    mov ah, 0dh        ; get pixel.
    int 10h

    xor al, 1111b      ; pixel color
    mov ah, 0ch        ; set pixel
    int 10h
    pop dx
    pop cx

```

```

not_required:
    mov ah, 0dh      ; get pixel.
    int 10h
    xor al, 1111b   ; pixel color
    mov oldX, cx
    mov oldY, dx

draw_pixel:
    mov ah, 0ch     ; set pixel
    int 10h

check_esc_key:
    mov dl, 255
    mov ah, 6
    int 21h
    cmp al, 27      ; esc?
    jne check_mouse_button

stop:
    ;mov ax, 2 ; hide mouse cursor.
    ;int 33h
    mov ax, 3 ; back to text mode: 80x25
    int 10h
    ; show box-shaped blinking text cursor:
    mov ah, 1
    mov ch, 0
    mov cl, 8
    int 10h
    mov dx, offset msg
    mov ah, 9
    int 21h
    mov ah, 0
    int 16h
    ret
msg db " press any key....    $"
end proses

```

3. Simpan Program yang anda ketik di atas dengan nama_program.asm.
4. Gunakan perintah TASM dan TLINK.
5. Jalankan program.

7.5. TUGAS

Analisis Program di atas dengan menggunakan trace. Jelaskan perubahan yang terjadi!

PRAKTIKUM 8. PEMROGRAMAN ASSEMBLY MENGGUNAKAN EMU8086

Pertemuan ke	: 8
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

8.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

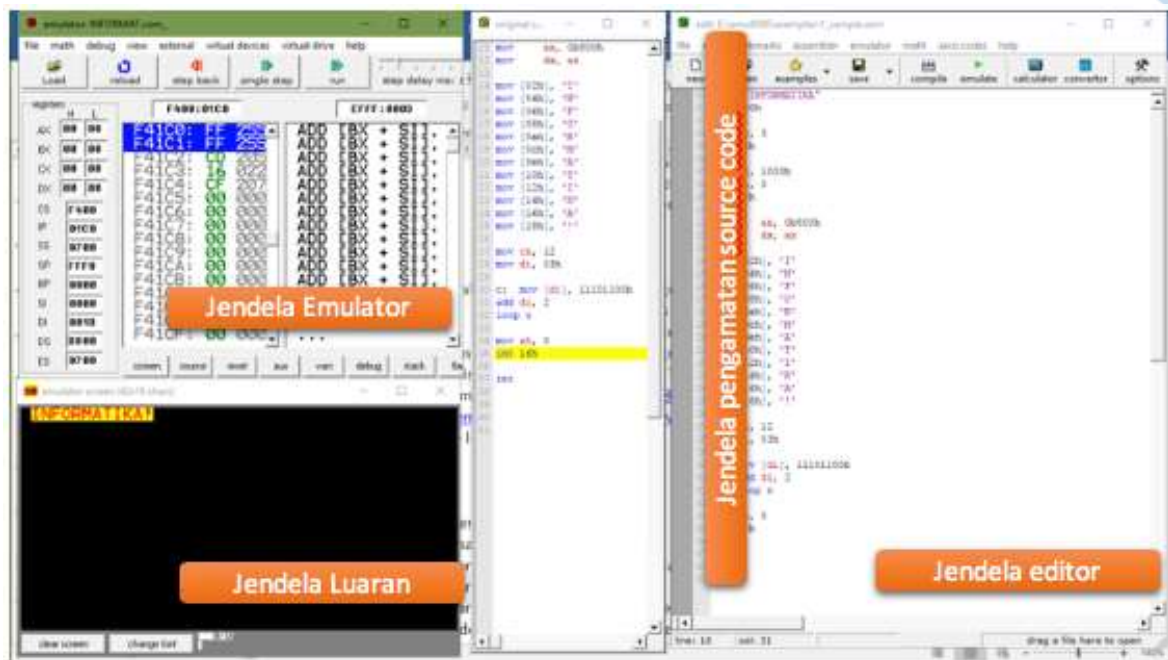
1. mampu mengenal salah satu *tools* pemrograman bahasa assembly.
2. mampu menggunakan editor emu8086 untuk menulis kode bahasa assembly.
3. mampu meng-*compile* dan menjalankan program assembly sederhana dengan emu8086.

Indikator ketercapaian diukur dengan:

Penguasaan tool EMU8086 untuk membuat program assembly sederhana

8.2. TEORI PENDUKUNG

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. Dengan menggunakan aplikasi Emu8086, kita dapat mensimulasikan apakah program yang kita buat sudah benar atau masih salah sekaligus bisa melakukan debugingnya. Tools emu8086 dapat diperoleh secara gratis di web pengembangnya pada link berikut: <http://bit.ly/2LhRomf>



Gambar 8.1. Tampilan emu8086

8.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Emu8086

8.4. LANGKAH PRAKTIKUM

1. Buka aplikasi Emu8086.
2. Ketikkan code program berikut di editor emu8086, setelah mengklik tombol NEW dan template Empty Workspace

```
name "INFORMATIKA"
org 100h
    mov ax, 3
    int 10h
    mov ax, 1003h
    mov bx, 0
    int 10h
    mov ax, 0b800h
    mov ds, ax
    mov [02h], 'I'
    mov [04h], 'N'
    mov [06h], 'F'
    mov [08h], 'O'
    mov [0ah], 'R'
    mov [0ch], 'M'
    mov [0eh], 'A'
    mov [10h], 'T'
    mov [12h], 'I'
    mov [14h], 'K'
    mov [16h], 'A'
```

```
    mov [18h], '!'  
    mov cx, 12  
    mov di, 03h  
c:   mov [di], 11101100b  
    add di, 2  
    loop c  
    mov ah, 0  
    int 16h  
Ret
```

3. Klik di sistem menu tombol/menu: Emulate, di jendela emulator: klik Run dan amati jendela output.

8.5. TUGAS

Modifikasilah program tersebut di atas sehingga mampu memunculkan tulisan INFORMATIKA-UAD-JAYA

PRAKTIKUM 9. PENELUSURAN PROGRAM ASSEMBLY MENGGUNAKAN EMU8086

Pertemuan ke	: 9
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

9.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu mengenal fitur-fitur debugging emu8086
2. mampu menggunakan emu8086 untuk melakukan tracing/penelusuran kode program assembly dan langsung mengamati luaran yang dihasilkan.

Indikator ketercapaian diukur dengan:

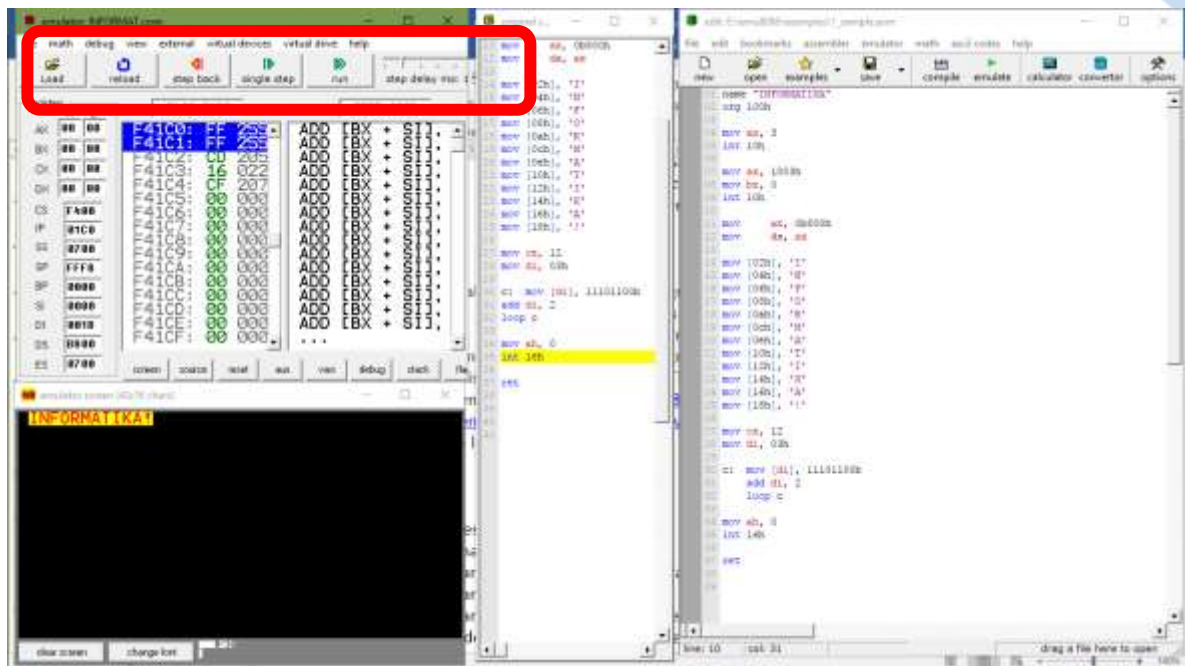
1. kemampuan menggunakan emu8086 untuk melakukan tracing/penelusuran kode program assembly dan langsung mengamati luaran yang dihasilkan.

9.2. TEORI PENDUKUNG

Emu8086 dirancang sebagai emulator yang salah satunya dimanfaatkan sebagai tools untuk melakukan debugging program assembly. Proses tracing/ penelusuran baris demi baris program assembly dapat diamati secara langsung luarannya sesuai dengan perintah pada baris program yang bersangkutan. Keunggulan lainnya di samping dapat menelusuri baris demi baris dan hasil keluarannya, satu fitur visualisasi proses yang terjadi dalam prosesor (terutama penggunaan alamat register) dapat dipantau juga.

Fitur yang terkait hal tersebut dalam emu8086 disediakan beberapa menu, antara lain:

1. Compile, berguna untuk membuat (compile) "binary executable file" atau ekstensi ".com" atau ".exe" dari kode program.
2. Emulate, berguna untuk menjalankan emulator kode program yang kita buat
3. Run, menjalankan aplikasi dari emulator yang dihasilkan (execute).
4. Single Step, berfungsi untuk menjalankan aplikasi dengan cara *tracing* (diproses perbaris kode program).
5. Step Delay (dlm milidetik), yang berfungsi untuk memperlambat eksekusi jalannya program (baris program) sehingga bisa dipantau lebih seksama.



Gambar 9.1. Fasilitas debug di emu8086

9.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Emu8086

9.4. LANGKAH PRAKTIKUM

1. Bukalah hasil modifikasi dari praktikum ke-7 (hasil tugas post test praktikum ke-7) dalam editor emu8086 (dengan menu open)
2. Klik di sistem menu tombol/menu: Emulate, akan tampil jendela emulator
3. Lakukan tracing/ penelusuran secara:
 - Otomatis, dengan tombol Run, angkahnya:
 - Di Jendela emulator: klik dan geser step delay/ waktu penundaan sesuai selera (dalam mili detik)
 - Di jendela emulator: klik Run dan amati jendela output yang ditampilkan.
 - Manual, dengan tombol Single Step dan Step Back, langkahnya:
 - Di jendela emulator: klik Single Step dan amati jendela output yang ditampilkan.
 - Penelusuran akan berlangsung baris demi baris dan pengamatan luaran bisa lebih seksama
 - Lakukan secara berulang klik Single Step hingga program berakhir, dan jika dalam jendela luaran meminta interaksi dengan user harus dilakukan tindakan agar proses penelusuran berjalan.
 - Jika perlukan untuk kembali ke langkah penelusuran sebelumnya, bisa menggunakan tombol Step Back.

9.5. TUGAS

Lakukan penelusuran dan capture/ screenshot dari beberapa proses di program tersebut (minimal 3 statement program) dan jelaskan analisisnya berdasarkan baris program-alamat register-output program.

PRAKTIKUM 10. PEMROGRAMAN ASSEMBLY MENGGUNAKAN MICROPROCESSOR SIMULATOR

Pertemuan ke	: 10
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 15 menit
• Praktikum	: 45 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 30 %
• Post-Test	: 50 %

10.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. mampu memahami cara kerja program assembly menggunakan Microprocessor Simulator
2. mampu menjelaskan cara kerja sebuah program assembly

Indikator ketercapaian diukur dengan:

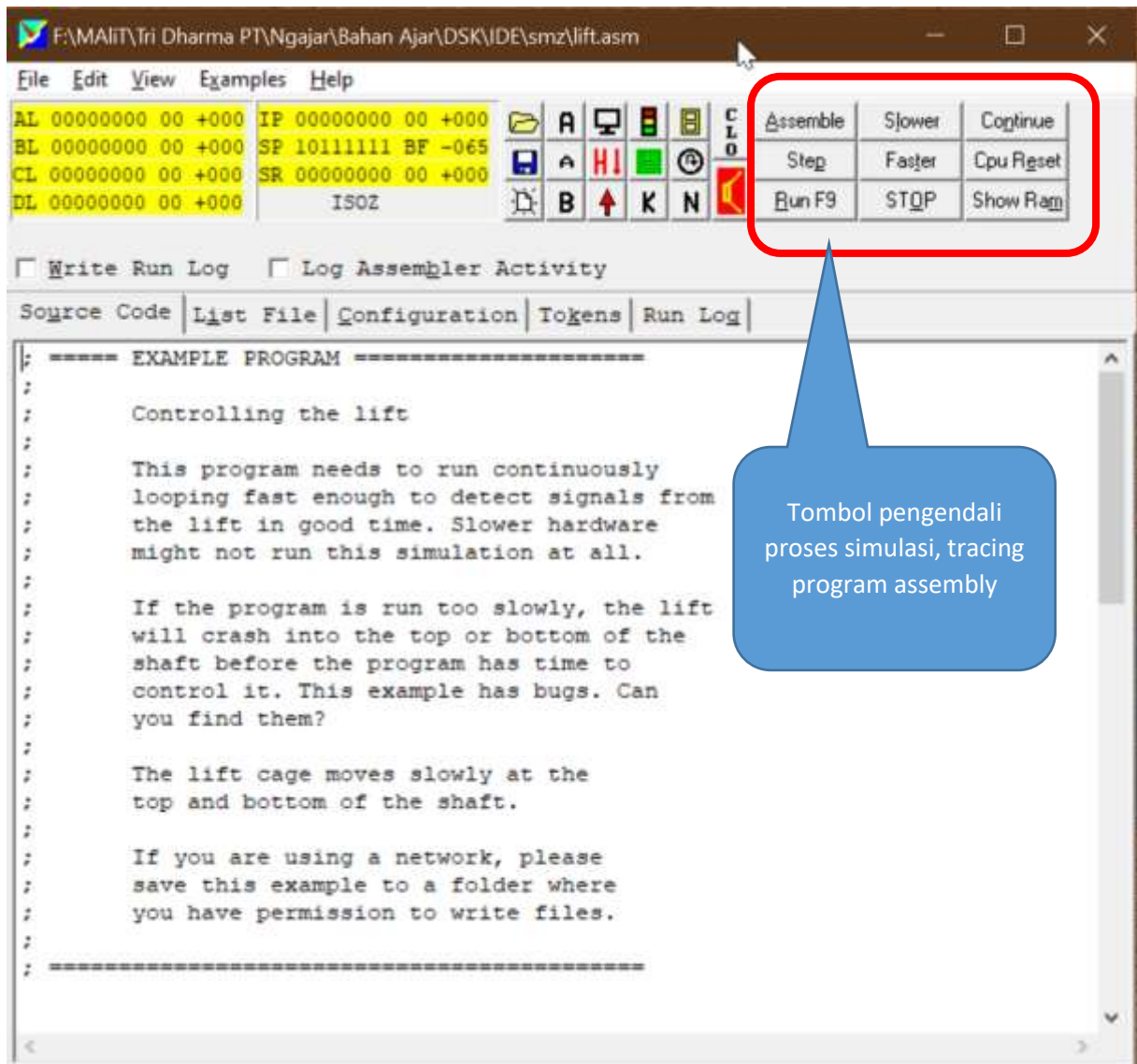
1. menjalankan Microprocessor Simulator dan mengamati perilaku program
2. menerapkan program assembly di Microprocessor Simulator

10.2. TEORI PENDUKUNG

Bahasa assembly yang ditulis dengan bahasa mendekati bahasa mesin membutuhkan pemahaman tentang kode-kode yang cukup rinci dan membutuhkan waktu lama untuk mengerti alur jalannya program. Melalui sebuah tools, salah satunya Microprocessor Simulator, penulisan hingga penelusuran jalannya program aplikasi dari bahasa assembly semakin mudah digunakan. Untuk mendapatkan aplikasi Microsof Simulator dapat mengunjungi halaman web berikut:

<http://www.softwareforeducation.com/sms32v50/>

J



Gambar 10.1. Tampilan antarmuka Microprocessor Simulator

10.3. ALAT DAN BAHAN

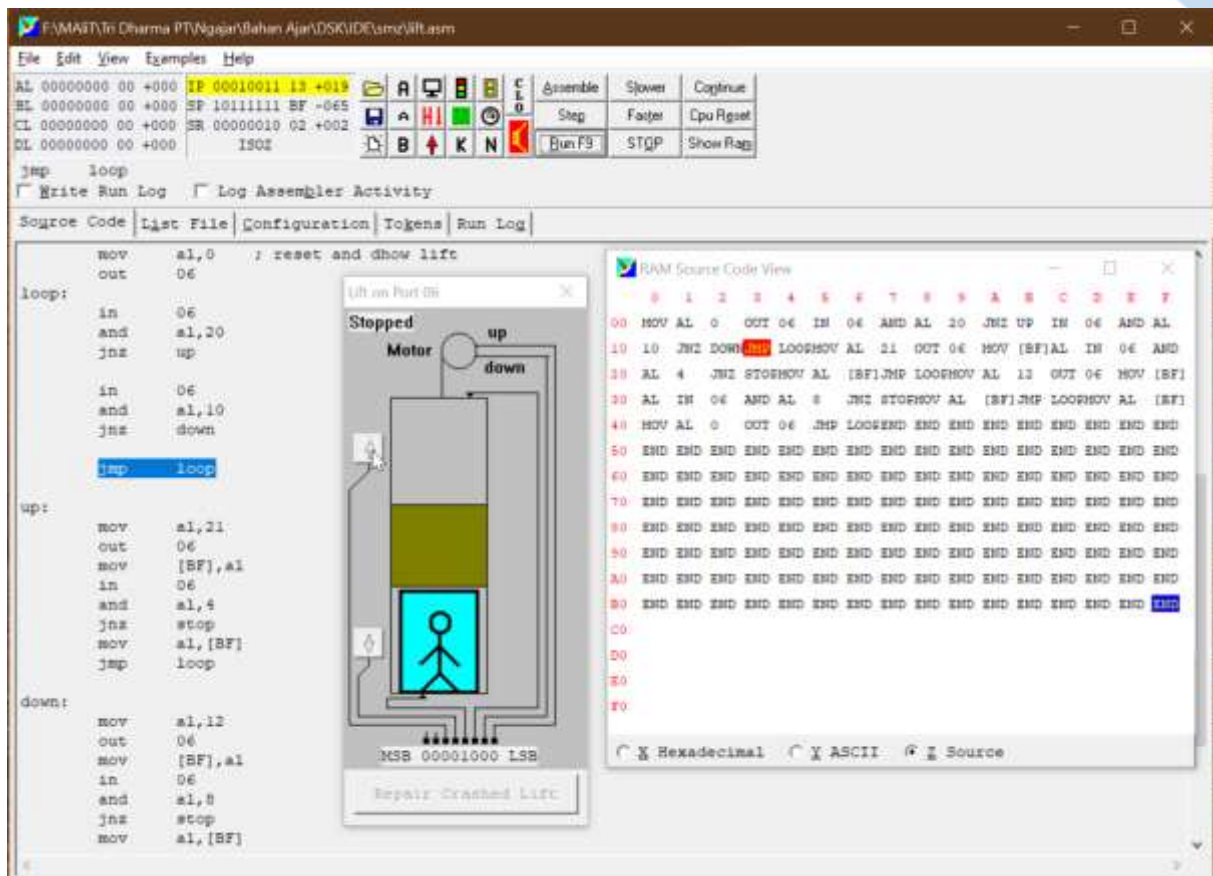
Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Microprocessor Simulator

10.4. LANGKAH PRAKTIKUM

Berdasarkan program simulator (sekaligus editor) bahasa assembly Microsof Simulator (dipastikan sudah diekstrak dalam satu folder):

1. Jalankan aplikasi Microsof Simulator dengan cara, klik 2x file aplikasi (sms32v50.exe)
2. Bukalah project file sample: lift.asm, sehingga isi kode assembly muncul di jendela editor
3. Tekan tombol RUN F9



Gambar 10.2. Proses simulasi untuk program lift.asm

4. Program assembly di atas adalah simulasi sebuah LIFT, lakukan percobaan dengan menekan tombol panah ke atas/bawah pada lift dan amati listing program dan pergerakan lif.
5. Untuk mempercepat/memperlambat tracing program dapat memanfaatkan tombol FASTER/SLOWER.
6. Jika pengamatan sudah cukup, simulasi dihentikan dengan tombol STOP.

10.5. TUGAS

Berdasarkan simulasi program assembly lift.asm di atas jelaskan cara kerja listing program dan berikan alur cara kerja dengan kondisi-kondisi yang disimulasikan pada gambar aplikasi lift tersebut.

LEMBAR JAWABAN PRE-TEST / POST-TEST

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

DAFTAR PUSTAKA

1. S'to, 2014, Pemrograman Bahasa Assembly versi 1.0, Jasakom.
2. Bob Nevelin, 1015, Assembly Language Programming, Prentice-Hall
3. El Rewini, Hesam. Abd-El-Barr, Mostafa. 2015, Advanced Computer Architecture and Parallel Processing, Wiley.
4. Tanenbaum, Andrews, 2015, Computer Network Five Edition, Prentice Hall

