



**PERENCANAAN MODUL PRAKTIKUM
Z80 SIMULATOR SEBAGAI ALAT BANTU PEMBELAJARAN
SISTEM MIKROPROSESOR Z80**

SKRIPSI

**Diajukan dalam Rangka Penyelesaian Studi Strata-I
Untuk Memperoleh Gelar Sarjana Pendidikan**

oleh:
Alwi Nur Faizal
5301410056

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2016**



UNNES

UNIVERSITAS NEGERI SEMARANG

PERNYATAAN

Saya menyatakan bahwa yang tertulis dalam skripsi ini merupakan benar-benar hasil karya sendiri, bukan plagiat dari karya tulis orang lain, baik sebagian maupun seluruhnya. Pendapat atau temuan yang terdapat pada skripsi ini dikutip atau dirujuk sesuai dengan kode etika ilmiah.



Semarang, Februari 2016

Penulis,

A handwritten signature in black ink, appearing to be 'Alwi Nur Faizal', written over a white background.

Alwi Nur Faizal
NIM. 5301410056

UNNES
UNIVERSITAS NEGERI SEMARANG

PENGESAHAN

Skripsi ini telah dipertahankan dihadapan Sidang Panitia Ujian Skripsi Fakultas Teknik Universitas Negeri Semarang pada:

Hari : Rabu

Tanggal : 27 April 2016

Panitia Ujian Skripsi

Ketua



Dr. Ing. Dhidik Prastivanto, S.T., M.T.
NIP. 197805312005011002

Sekretaris



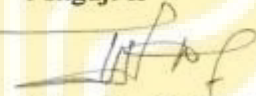
Drs. Agus Suryanto, M.T.
NIP. 196708181992031004

Penguji I



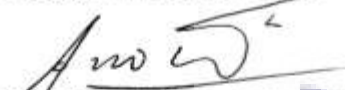
Dr. Ing. Dhidik Prastivanto, S.T., M.T.
NIP. 197805312005011002

Penguji II



Riana Defi Mahadji Putri, S.T., M.T.
NIP. 197609182005012001

Penguji III / Pembimbing



Drs. Slamet Seno Adi, M.Pd., M.T.
NIP. 195812181985031004

UNNES

UNIVERSITAS NEGERI SEMARANG

Mengetahui
Dekan Fakultas Teknik



Dr. Nur Qudus, M.T.
NIP. 196911301994031001

MOTTO DAN PERSEMBAHAN

MOTTO

- “Sesungguhnya Allah tidak akan mengubah nasib suatu kaum kecuali kaum itu sendiri yang mengubah apa-apa yang terjadi pada diri mereka.”

(Q.S. Ar-Ra'd : 11)

- “Urusan kita dalam kehidupan bukanlah mendahului orang lain, tetapi maju mendahului diri kita sendiri.”

(Stuart B. Johnson)

- Keajaiban adalah buah dari usaha yang sungguh-sungguh dan doa dengan keyakinan teguh.

PERSEMBAHAN:

Skripsi ini dipersembahkan untuk:

1. Bapak dan Ibu tercinta (Bapak Wiyono dan Ibu Umi Nur Hanik) yang telah selalu memberikan semangat, kasih sayang dan doa restunya.
2. Kakak (Alfian Nur Affandi) dan adik-adikku tercinta (Arif Nur Choeroni & Aswin Nur Fauzi) yang tak henti-hentinya selalu mengingatkan, memberi support dan semangat.
3. Sahabatku (M. Imam Hanif) yang telah menyadarkan dan mengajarkan arti penting dari usaha.
4. Teman-teman seperjuangan, PTE 2010.

KATA PENGANTAR

Alhamdulillah hirabbil Alamin, puji syukur dipanjatkan kehadirat Allah SWT atas limpahan nikmat, rahmat serta hidayah-Nya yang telah memberikan kemudan dan bimbingan-Nya sehingga terselesaikannya skripsi **“PERENCANAAN MODUL PRAKTIKUM Z80-SIMULATOR SEBAGAI ALAT BANTU PEMBELAJARAN SISTEM MIKROPROSESOR Z80”**.

Selama penyusunan dan penulisan skripsi ini banyak nasihat, arahan, bimbingan, semangat dan bantuan dari berbagai pihak, oleh karena itu dengan kerendahan hati ucapan terimakasih diberikan kepada:

1. Bapak Drs. Slamet Seno Adi, M.Pd., M.T., selaku dosen pembimbing yang telah memberikan arahan, motivasi dan bimbingan dalam penyusunan skripsi ini.
2. Prof. Dr. Nur Qudus, M.T., selaku Dekan Fakultas Teknik Universitas Negeri Semarang.
3. Bapak Dr.-Ing. Dhidik Prastiyanto, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang.
4. Bapak Drs. Sugeng Purbawanto M.T., selaku dosen wali.
5. Bapak, Ibu Dosen dan staf di Jurusan Teknik Elektro UNNES yang telah memberikan ilmu dan pengetahuan kepada penulis.
6. Sahabatku Imam Hanif yang senang hati peduli, memberikan semangat dan mendengarkan keluh kesah.
7. Teman-teman seperjuangan PTE 2010 yang telah membantu, memberikan dorongan dan semangat, serta kerja samanya.

Semoga Allah SWT melimpahkan barokah, rahmat dan hidayah-Nya atas segala amal dan budi baik yang telah dilakukan sehingga skripsi ini dapat diselesaikan. Akhir kata semoga skripsi ini dapat bermanfaat dan berharap dapat dimanfaatkan sebagaimana mestinya. Amin.

Semarang, Februari 2016

Penyusun

ABSTRAK

Faizal, Alwi Nur. 2010. *Perencanaan Modul Praktikum Z80-Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80.* Skripsi, Jurusan Teknik Elektro, Universitas Negeri Semarang. Drs. Slamet Seno Adi, M.Pd, M.T.

Pemahaman akan mikroprosesor sangat dibutuhkan dalam pengembangan sebuah mikrokomputer. Zilog 80 dipilih sebagai media pembelajaran mikroprosesor dikarenakan arsitektur yang sederhana dan pemrogramannya yang masih dasar. Jurusan Teknik Elektro, Fakultas Teknik UNNES mempergunakan Zilog 80, melalui Mata Kuliah Praktik Sistem Mikroprosesor dipelajari tentang bagaimana cara merancang, memprogram, mengontrol serta mengoperasikan mikroprosesor Z80. Terbatasnya *kit trainer* MPF-1 seringkali mengakibatkan mahasiswa mengalami kesulitan dalam praktik sehingga pemahaman akan sistem mikroprosesor kurang optimal. Berdasarkan permasalahan tersebut penelitian ini mencoba memfokuskan tentang menyusun modul praktikum sistem mikroprosesor Z80 menggunakan alat bantu program aplikasi Z80 simulator. Penyusunan modul praktikum ini disesuaikan dengan silabus dan RPP dari Mata Kuliah Praktik Sistem Mikroprosesor. Tujuan penelitian ini untuk menguji kelayakan modul praktikum mikroprosesor Z80 dengan alat bantu Z80-Simulator sebagai media pendukung alternatif pada Mata Kuliah Praktik Sistem Mikroprosesor, khususnya tentang mikroprosesor Z80.

Penelitian ini menggunakan metode penelitian dan pengembangan (*Research and Development*). Analisa data yang digunakan meliputi validasi modul oleh pakar penyajian, modul yang diuji berupa aspek teknik penyajian, materi penyajian, dan penyajian pembelajaran serta pengujian di lapangan yang dilakukan dalam perkuliahan praktik sistem mikroprosesor dengan melibatkan 30 orang mahasiswa.

Hasil validasi modul dari angket pakar ahli aspek penyajian diperoleh penilaian kelayakan teknik penyajian 92.5%, kelayakan materi penyajian 89.3%, dan kelayakan penyajian pembelajaran 92.5% hal ini disimpulkan bahwa modul telah layak. Pada tanggapan siswa 4 item soal yang memiliki penilaian lebih rendah dari item yang lain. Item soal no. 2 dinilai 73%, item no. 6 karakteristek modul praktikum untuk belajar secara mandiri tanpa bantuan guru memperoleh nilai sebesar 70%, item no. 8 kemenarikan penyajian modul praktikum 76%, no. 11 modul dapat mempermudah mempelajari sistem mikroprosesor sebesar 76%. Walau demikian berdasarkan patokan nilai $\geq 70\%$ tanggapan siswa dinyatakan tanggapan positif. Berdasarkan analisis data dapat disimpulkan modul praktikum Z80 Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80 dinyatakan valid dan dapat digunakan dalam kegiatan pembelajaran praktik sistem mikroprosesor untuk membantu pemahaman terhadap pemrograman mikroprosesor khususnya Z80 yang didukung oleh respon tanggapan positif siswa dengan rata-rata 80% atau sangat setuju.

Kata Kunci: Mikroprosesor, Modul, Z80-Simulator.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERNYATAAN	iii
LEMBAR PENGESAHAN	iv
MOTTO DAN PERSEMBAHAN	v
KATA PENGANTAR	vi
ABSTRAK	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah	4
C. Batasan Masalah	5
D. Rumusan Masalah	5
E. Tujuan Penelitian	6
F. Manfaat Penelitian	6
G. Penegasan Istilah	8
H. Sistematik Penulisan	10

BAB 2 LANDASAN TEORI	12
A. Pembelajaran	12
1. Pengertian Pembelajaran	12
2. Model Pembelajaran untuk Pengembangan Bahan Ajar	13
B. Modul	18
1. Pengertian Modul	18
2. Karakteristik Modul	19
3. Tahap Penyusunan Modul	20
C. Praktik Sistem Mikroprosesor	23
1. Pengertian Praktik Sistem Mikroprosesor	23
2. Mikroprosesor Z80	24
3. Instruksi-Instruksi Mikroprosesor Z80	28
4. Z80-Simulator	35
D. Penyusunan Modul Praktikum Z80 Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80	47
1. Mengidentifikasi Tujuan Instruksional	47
2. Melakukan Analisis Pembelajaran	48
3. Menganalisis Karakteristik Siswa dan Konteks Pembelajaran	48
4. Merumuskan Tujuan Khusus	48
5. Mengembangkan Instrumen Penilaian	48
6. Mengembangkan Strategi Pembelajaran	49
7. Mengembangkan dan Memilih Materi Ajar	49
8. Merancang dan Mengembangkan Evaluasi Formatif	49

9. Melakukan Revisi Terhadap Program Pembelajaran	50
10. Merancang dan Mengembangkan Evaluasi Sumatif	50
E. Kerangka Berfikir	50
BAB 3 METODE PENELITIAN	54
A. Tempat Dan Waktu Penelitian	54
B. Desain Penelitian	54
1. Tahap Analisis (<i>Define</i>)	56
2. Tahap Perancangan (<i>Desaign</i>)	57
3. Tahap Pengembangan (<i>Development</i>)	58
C. Alat dan Bahan Penelitian	61
1. Alat Penelitian	61
2. Bahan Penelitian	61
D. Parameter Penelitian	62
E. Pengumpulan Data	62
1. Metode Observasi	62
2. Metode Dokumentasi	63
3. Metode Kuisisioner (angket)	63
F. Kalibrasi Instrumen	65
G. Teknik Analisis Data.....	66
1. Angket Validator	66
2. Angket Responden Siswa	68

BAB 4 HASIL PENELITIAN DAN PEMBAHASAN	69
A. Deskripsi Data	69
B. Analisis Data	69
1. Hasil Pengembangan Modul	69
2. Hasil Validasi Modul	75
3. Hasil Uji Coba	77
C. Pembahasan	80
1. Modul Praktikum	80
2. Validasi dan Ujicoba Modul Praktikum	82
BAB 5 PENUTUP	90
A. Simpulan	90
B. Saran	90
DAFTAR PUSTAKA	92
LAMPIRAN	93



DAFTAR GAMBAR

	Halaman
Gambar 1. Siklus Experiential Learning pada Kolb's Learning Style.....	16
Gambar 2. Tampilan <i>Register</i> Mikroprosesor Z80	27
Gambar 3. Tampilan Jendela Utama Simulator Z80.....	37
Gambar 4. Tampilan Jendela <i>Assembler</i>	39
Gambar 5. Tampilan Jendela <i>Memory Editor</i>	40
Gambar 6. Tampilan Jendela <i>Peripheral Device</i>	40
Gambar 7. Tampilan Jendela <i>Breakpoint Manager</i>	41
Gambar 8. Tampilan Jendela <i>Display LCD</i>	42
Gambar 9. Tampilan Mengaktifkan Simulator Z80 melalui Start	42
Gambar 10. Tampilan mengaktifkan <i>Tool Assembler</i> pada jendela utama simulator	43
Gambar 11. Tampilan penulisan program pada lembar kerja <i>Assembler</i>	43
Gambar 12. Tampilan me- <i>assemble</i> program yang telah ditulis	44
Gambar 13. Tampilan kolom hasil <i>assemble program</i>	44
Gambar 14. Tampilan proses memasukkan data ke memori simulasi	45
Gambar 15. Tampilan program location box jika data berhasil di load/dimasukkan Pada memori simulasi	45
Gambar 16. Tampilan isi Memory Editor setelah program di- <i>load</i>	46
Gambar 17 Tampilan memulai simulasi	46
Gambar 18. Tampilan mengatur kecepatan simulasi	46
Gambar 19. Model desain penyusunan modul menurut Dicky dan Carry	47

Gambar 20. Kerangka berfikir penelitian	53
Gambar 21. Alur desain penelitian pengembangan modul yang di adaptasi Dari Thiagarajan dan Semmel.....	55
Gambar 22. Grafik Penilaian Hasil Angket Tanggapan Siswa Terhadap Modul praktikum	79



DAFTAR TABEL

	Halaman
Tabel 1. Aktivitas Yang Siswa dalam proses Belajar Experiential Learning	17
Tabel 2. Keterangan <i>Register</i> Mikroprosesor Z80	27
Tabel 3. Contoh Penulisan Instruksi Transfer Data	30
Tabel 4. Contoh Instruksi Aritmatika dan Logika	31
Tabel 5. Instruksi-Instruksi <i>jump</i> bersyarat	35
Tabel 6. Gradasi Nilai Angket	64
Tabel 7. Uji Validasi Butir Soal Instrumen	66
Tabel 8. Jenjang Kategori Skor Kualitatif	67
Tabel 9. Hasil Evaluasi dan Revisi Modul praktikum	77
Tabel 10. Persentase Hasil Penilaian Tanggapan Siswa	78



DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Keputusan Dosen Pembimbing	93
Lampiran 2. Surat Ijin Penelitian	94
Lampiran 3. Surat Tugas Panitia Ujian	95
Lampiran 4. Silabus Praktik Sistem Mikroprosesor	96
Lampiran 5. Kisi-kisi angket validasi pakar ahli	98
Lampiran 6. Lembar angket validasi modul pakar ahli	99
Lampiran 7. Data Hasil rekapitulasi penilaian angket validasi modul pakar ahli	106
Lampiran 8. Kisi-kisi angket penilaian tanggapan siswa	107
Lampiran 9. Lembar angket penilaian tanggapan siswa	108
Lampiran 10. Data Hasil rekapitulasi penilaian angket tanggapan siswa	110
Lampiran 11. Dokumentasi Pelaksanaan Penelitian	111
Lampiran 12. Modul praktikum Mikroprosesor Z80 Dengan Alat Bantu Z80-Simulator	112



BAB 1

PENDAHULUAN

A. LATAR BELAKANG

Keberadaan komputer atau mikrokomputer sangatlah lekat dengan kegiatan manusia sehari-hari hampir disemua aspek kehidupan saat ini. Ketergantungan ini mendorong manusia untuk berusaha mengembangkan komputer ini ketaraf yang lebih canggih dan hebat. Untuk mendorong pengembangan komputer ini manusia diharuskan dapat memahami dan mengerti segala sesuatu akan komputer.

Pemahaman akan mikroprosesor sangat dibutuhkan dalam pengembangan sebuah mikrokomputer. Layaknya komputer, di dalam sebuah mikroprosesor pun memiliki struktur bangunan yang sama, mempunyai unit pemroses data, memiliki memori sebagai unit penyimpan data serta unit untuk operasi-operasi aritmatik dan logika. Sampai saat ini telah tercipta mikroprosesor dimulai dari mikroprosesor 8 bit, 16 bit, 32 bit dan 64 bit. Perkembangan mikroprosesor ini memiliki catatan yang panjang dan sejarah mencatat beberapa mikroprosesor yang terkenal pada masanya bahkan sampai sekarang masih di pelajari sebagai dasar dari pengembangan mikroprosesor lebih lanjut, seperti Zilog 80 (Z80), Intel 8080/8086, 6800, dan lain sebagainya.

Zilog 80 dipilih sebagai media pembelajaran mikroprosesor yang sering digunakan, hal ini dikarenakan struktur dan cara kerja dari mikroprosesor tersebut. Arsitektur yang sederhana dan pemrogramannya yang masih dasar sangat cocok digunakan sebagai alat belajar dalam memahami sebuah mikroprosesor. Jurusan

Teknik Elektro, Fakultas Teknik Universitas Negeri Semarang merupakan salah satu perguruan tinggi yang menggunakan Z80 (Zilog 80) dalam proses pembelajaran perkuliahannya. Melalui mata kuliah Praktik Sistem Mikroprosesor dipelajari tentang bagaimana cara merancang, memprogram, mengontrol serta mengoperasikan mikroprosesor Z80 menggunakan *kit trainer* MPF-1. Namun dengan terbatasnya *kit trainer* MPF-1 yang dimiliki Jurusan Teknik Elektro UNNES, seringkali mengakibatkan mahasiswa mengalami kesulitan dalam praktik sehingga pemahaman akan sistem mikroprosesor kurang optimal. Hal ini diperburuk dengan adanya beberapa *kit trainer* yang mengalami kerusakan sehingga kinerjanya tidak stabil dan tidak maksimal. Maka diperlukan adanya tambahan *kit trainer* MPF-1 atau suatu media pendukung lainya yang dapat membantu praktik dan latihan mahasiswa untuk meningkatkan pemahaman mahasiswa terhadap sistem mikroprosesor khususnya mikroprosesor Z80.

Saat ini, program aplikasi Z80 Simulator dapat ditemukan secara mudah dan bebas di internet. Aplikasi Z80 Simulator, merupakan program berbasis simulasi yang dapat digunakan untuk mensimulasikan pemrograman sistem mikroprosesor Z80 secara virtual. Dengan aplikasi Z80 Simulator ini, dapat digunakan mahasiswa sebagai media pendukung untuk mempelajari pemrograman mikroprosesor Z80 secara mandiri. Agar penggunaan aplikasi Z80 Simulator lebih terarah dan sesuai dengan kompetensi yang diharapkan, maka perlu disusun kedalam sebuah modul praktikum. Berdasarkan permasalahan yang telah dikemukakan, maka perlu dilakukannya penelitian berupa **"PERENCANAAN**

MODUL PRAKTIKUM Z80-SIMULATOR SEBAGAI ALAT BANTU PEMBELAJARAN SISTEM MIKROPROSESOR Z80”.

B. IDENTIFIKASI MASALAH

Dalam merumuskan suatu identifikasi masalah perlu meninjau dari beberapa aspek permasalahan secara nyata yang terjadi dalam kegiatan pembelajaran. Kegiatan Praktik Sistem Mikroprosesor khususnya materi Mikroprosesor Z80 yang dilakukan oleh Jurusan Teknik Elektro, Fakultas Teknik UNNES menggunakan *kit training* mikroprosesor Z80 berupa MPF-1. Terdapat tujuh (7) buah *kit training* MPF-1 yang dimiliki oleh Lab Jurusan Teknik Elektro UNNES. Dari *kit training* MPF-1 yang ada terdapat beberapa yang mengalami kerusakan sehingga kurang stabil digunakan. Terbatasnya jumlah alat dan ketidak stabilan alat sering membuat mahasiswa membutuhkan waktu lebih untuk bergantian melaksanakan praktikum sehingga pemahaman akan materi kurang bisa optimal dipraktikkan. Oleh karena itu, diperlukan suatu media pendukung yang dapat *membackup* serta sebagai media latihan mahasiswa untuk memperlancar dan meningkatkan pemahaman mahasiswa terhadap praktik mikroprosesor khususnya mikroprosesor Z80. Maka peneliti mencoba menggunakan aplikasi Z80 Simulator untuk mengatasi permasalahan tersebut. Aplikasi Z80 Simulator ini dapat menampilkan pemrograman mikroprosesor Z80 dengan lebih mudah. Penggunaan aplikasi Z80 Simulator untuk digunakan dalam pembelajaran perlu disesuaikan dengan kurikulum yang ada, yang kemudian dapat disusun dalam sebuah modul praktikum. Dengan modul praktikum ini diharapkan mahasiswa

dapat menggunakan aplikasi Z80 Simulator untuk mempelajari mikroprosesor Z80 dengan lebih terarah dan terstruktur, sehingga tercapai pemahaman sistem mikroprosesor Z80 yang lebih baik.

C. BATASAN MASALAH

Berdasarkan penjabaran latar belakang dan identifikasi masalah, penelitian akan membatasi permasalahan tentang :

1. Materi yang disajikan dalam perencanaan modul praktikum berbasis simulasi program komputer hanya menyangkut Mata Kuliah Praktik Sistem Mikroprosesor bahasan mikroprosesor Z80.
2. Media pembelajaran berbasis simulasi komputer ini menggunakan program komputer (*software*) Z80-Simulator OSHONSOFT.
3. Jenis modul praktikum yang dibuat merupakan jenis tutorial praktikum yang berisi penyajian materi pembelajaran dan contoh soal.

D. RUMUSAN MASALAH

Berdasarkan identifikasi masalah yang telah diuraikan dalam penelitian ini dapat dirumuskan permasalahan yang akan dibahas, antara lain:

1. Bagaimana merencanakan, membuat dan menguji Modul Praktikum Z80-Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80?
2. Apakah Modul Praktikum Z80-Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80 layak digunakan untuk menjadi

media pendukung pembelajaran alternatif Mata Kuliah Praktik Sistem Mikroprosesor khususnya tentang mikroprosesor Z80?

E. TUJUAN PENELITIAN

Adapun tujuan yang ingin dicapai dari penelitian ini adalah untuk :

1. Syarat kelulusan penyelesaian Studi Strata I (S1) untuk memperoleh gelar Sarjana Pendidikan (S.Pd.).
2. Mendapatkan pengetahuan bagaimana merencanakan, membuat dan menguji Modul Praktikum Z80-Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80.
3. Mengetahui kelayakan Modul Praktikum Z80-Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80 untuk menjadi media pendukung pembelajaran alternatif pada Mata Kuliah Praktik Sistem Mikroprosesor, khususnya tentang mikroprosesor Z80.

F. MANFAAT PENELITIAN

Melalui penelitian ini diharap dapat memberikan manfaat sebagai berikut:

1. Bagi Dosen

Media pembelajaran berbasis simulasi program komputer ini dapat digunakan untuk membantu dosen dalam mempraktekkan dan menjelaskan penyampaian materi perkuliahan. Dosen juga dapat ikut membantu dalam mengembangkan dan memvariasikan media ini dalam

proses pembelajaran maupun ketika praktik, sehingga memberikan suasana baru yang menarik dalam proses perkuliahan.

2. Bagi Mahasiswa

Tersedianya Modul Praktikum Z80-Simulator sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80 yang layak digunakan dalam pembelajaran Praktik Sistem Mikroprosesor. Dengan media pembelajaran berbasis simulasi program komputer, mahasiswa diharapkan semakin mudah memahami cara kerja mikroprosesor khususnya mikroprosesor Z80, sehingga memberikan pemahaman yang lebih baik untuk mempermudah kegiatan praktik belajar, serta mampu untuk meningkatkan minat belajar mahasiswa dengan belajar aktif mandiri sehingga mendorong terciptanya peningkatan prestasi belajar.

3. Bagi Lembaga Pendidikan

Penelitian ini dapat digunakan sebagai salah satu sumber referensi media pendukung pembelajaran alternatif khususnya pada Mata Kuliah Praktik Sistem Mikroprosesor tentang mikroprosesor Z80 oleh perguruan tinggi. Selain itu pula dapat digunakan sebagai bahan awal untuk keperluan penelitian yang akan datang.

G. PENEKASAN ISTILAH

Beberapa istilah yang perlu ditegaskan untuk mendukung menjelaskan arah tujuan penelitian, diantaranya:

1. Perencanaan

Perencanaan adalah menyusun langkah-langkah yang dibutuhkan untuk mencapai tujuan yang telah ditentukan. Dijelaskan oleh Wiliam H. Newman pada Abdul Majid (2009:15) bahwa perencanaan adalah menentukan apa yang akan dilakukan. Perencanaan mengandung rangkaian-rangkaian keputusan yang luas dan penjelasan-penjelasan dari tujuan, penentuan kebijakan, penentuan program, penentuan metode-metode dan prosedur tertentu serta penentuan jadwal kegiatan berdasarkan jadwal sehari-hari.

2. Modul

Menurut Daryanto (2013:9) modul merupakan salah satu bentuk bahan ajar yang dikemas secara utuh dan sistematis, didalamnya memuat seperangkat pengalaman belajar yang terencana dan didesain untuk membantu peserta didik menguasai tujuan belajar yang spesifik. Modul minimal memuat tujuan pembelajaran, materi belajar dan evaluasi. Berfungsi sebagai sarana belajar yang bersifat mandiri, sehingga peserta didik dapat belajar mandiri sesuai kecepatan masing-masing.

3. Praktik Sistem Mikroprosesor Z80

Praktik Sistem Mikroprosesor adalah salah satu mata kuliah wajib yang harus ditempuh oleh mahasiswa Pendidikan Teknik Elektro UNNES.

Pada mata kuliah ini mahasiswa mempelajari tentang perancangan, pemrograman, serta mengoperasikan mikroprosesor seperti Mikroprosesor Zilog 80.

4. Alat Bantu Pembelajaran

Dalam kegiatan belajar mengajar, alat bantu pembelajaran adalah sesuatu yang dapat dijadikan sarana penghubung untuk mencapai pesan yang harus dicapai oleh siswa dalam kegiatan belajar. Seperti yang dikemukakan oleh Gagne' dan Briggs dalam Ashar Arysad (1997:4) bahwa secara implisit alat bantu pembelajaran meliputi alat yang secara fisik digunakan untuk menyampaikan isi materi pembelajaran, yang terdiri dari antara lain buku, tape rekorder, kaset, video camera, film, foto, gambar, grafik, televisi dan komputer. Sehingga dapat dikatakan pula bahwa alat bantu pembelajaran adalah komponen sumber belajar atau wahana fisik yang mengandung materi instruksional di lingkungan siswa yang dapat merangsang siswa untuk belajar.

5. Z80-Simulator

Z80 Simulator IDE adalah aplikasi yang kuat untuk menciptakan pengembang dan pendidik Mikroprosesor Z80 dengan pengembangan tampilan grafis yang user-friendly untuk Windows dalam bentuk simulator terintegrasi (emulator), compiler Basic, assembler, disassembler dan debugger untuk Zilog Z80 8-bit mikroprosesor. (Oshonsoft, 2001)

Z80-Simulator merupakan salah satu alat bantu pembelajaran yang dapat digunakan untuk menganalisis pemrograman Mikroprosesor Z80. Dengan

aplikasi ini diharapkan mahasiswa mempermudah peserta didik untuk memprogram Mikroprosesor Z80 dan memperkaya ilmu pemrograman mikroprosesor Z80 pada pembelajaran Praktik Sistem Mikroprosesor.

H. SISTEMATIKA PENULISAN

Sistematika penulisan dalam sebuah penelitian berperan untuk pedoman supaya penulisan lebih terarah dan sistematis dalam rangka menuju tujuan akhir yang hendak dicapai. Secara garis besar sistematika penulisan penelitian ini dibagi menjadi tiga bagian, yaitu: bagian awal skripsi, bagian inti skripsi dan bagian akhir skripsi:

1. Bagian awal skripsi

Bagian ini berisikan halaman judul, abstrak, lembar pengesahan, motto dan persembahan, kata pengantar, daftar isi, daftar tabel, daftar gambar dan daftar lampiran.

2. Bagian inti Skripsi

Bagian inti skripsi ini dibagi menjadi lima bab yaitu:

BAB I: Pendahuluan meliputi latar belakang masalah, identifikasi masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, penegasan istilah, dan sistematika penulisan.

BAB II: Landasan teori, berisi tentang landasan teori dan materi yang mendukung penelitian, penelitian yang relevan terkait penelitian sebelumnya berhubungan dengan penelitian yang sedang dilakukan dan kerangka berfikir.

BAB III: Metode Penelitian, membahas tentang rancangan penelitian, pengembangan modul, uji coba, metode pengumpulan data, metode analisis data.

BAB IV: Hasil Penelitian dan Pembahasan, menjelaskan tentang uraian Hasil Penelitian dan Pembahasan hasil penelitian.

BAB V: Penutup, berisi tentang kesimpulan dan saran.

3. Bagian akhir skripsi

Pada bagian akhir skripsi berisi daftar pustaka dan lampiran-lampiran.



BAB 2

LANDASAN TEORI

A. PEMBELAJARAN

1. Pengertian Pembelajaran

Pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Lingkungan pendidikan pembelajaran merupakan suatu proses penyampaian informasi dari guru ke siswa melalui media pembelajaran.

Dalam Undang-Undang No. 20 Tahun 2003 Tentang Sistem Pendidikan Nasional pasal 1 ayat 20 dinyatakan bahwa Pembelajaran adalah Proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pembelajaran dapat dilakukan dimana saja baik dalam lingkungan pendidikan/sekolah, masyarakat dan keluarga. Menurut Philips H. Coombs dalam Fuad Ihsan (2011) mengklasifikasikan pendidikan ke dalam tiga bagian, yaitu pendidikan informal (pendidikan di luar sekolah yang tidak dilembagakan) pendidikan formal (pendidikan sekolah) dan pendidikan non formal (pendidikan luar sekolah yang dilembagakan).

Dalam lingkungan pendidikan formal atau sekolah kegiatan pembelajaran akan lebih terarah karena dari materi, kurikulum, model dan metode belajar sudah disusun sedemikian rupa sehingga siswa dapat mengembangkan potensinya sesuai dengan kemampuannya masing-masing.

2. Model Pembelajaran untuk Pengembangan Bahan Ajar

Model pembelajaran adalah suatu perencanaan atau pola yang dapat digunakan untuk mendesain pola-pola mengajar secara tatap muka di dalam kelas atau mengatur tutorial, dan untuk menentukan material atau perangkat pembelajaran termasuk di dalamnya, buku, film, program-program media komputer, dan kurikulum. Setiap model mengarahkan kita untuk mendesain pembelajaran yang dapat membantu siswa untuk mencapai berbagai tujuan.

Fungsi model pembelajaran adalah sebagai pedoman bagi perancang pengajaran dan para guru dalam melaksanakan pembelajaran. Pemilihan model pembelajaran sangat dipengaruhi oleh sifat dari materi yang akan diajarkan, tujuan yang akan dicapai dalam pembelajaran tersebut, serta tingkat kemampuan peserta didik.

Sesuai dengan karakteristik praktik sistem mikroprosesor maka akan dikaji tentang model pembelajaran tuntas, model pembelajaran berbasis komputer dan model pembelajaran *experiential learning* untuk keperluan penyusunan Modul praktikum Z80-Simulator sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80.

a. Model Pembelajaran Tuntas

Pembelajaran tuntas menurut adalah sebuah pola pembelajaran yang menggunakan prinsip ketuntasan secara individual. Prinsip pembelajaran tuntas mengarahkan peserta didik untuk dapat menguasai ilmu pengetahuan (*knowledge*), sikap (*attitude*), dan ketarampilan (*skill*) agar mampu mencapai kompetensi yang diharapkan. Untuk melakukan

pembelajaran tuntas peserta didik belajar melalui (1) *learning by doing* (belajar melalui aktivitas/kegiatan nyata yang memberikan pengalaman bermakna) yang dikembangkan menjadi pembelajaran berbasis produksi, serta (2) melalui *individualized learning* (belajar sesuai keunikan individu) yang dilaksanakan dengan sistem modular.

Dalam pelaksanaan pembelajaran tuntas siswa diberikan waktu untuk mempelajari bahan yang diajarkan, untuk bahan yang diajarkan supaya setiap individu mampu untuk mempelajarinya secara tuntas maka diatur: (1) unit-unit belajar yang lebih kecil, (2) membuat tujuan khusus dari setiap unit, (3) membuat assesmen yang memadai, dan (4) merencanakan serta mengimplemantasikan stategi pembelajaran dengan memberikan alokasi yang cukup, kesempatan untuk berlatih dan mengulang sebagai perbaikan agar semua siswa (secara individu) dapat mencapai level ketuntasan yang diharapkan. Untuk itu sebagai media pembelajarannya dapat digunakan modul pembelajaran, yang tersusun dari unit-unit kecil dengan setiap unit berisi tujuan khusus dari materi ajar. Dengan demikian pelaksanaan pembelajaran dapat dilaksanakan sesuai sifat pembelajaran yakni untuk mengajarkan keterampilan.

b. Model Pembelajaran Berbasis Komputer

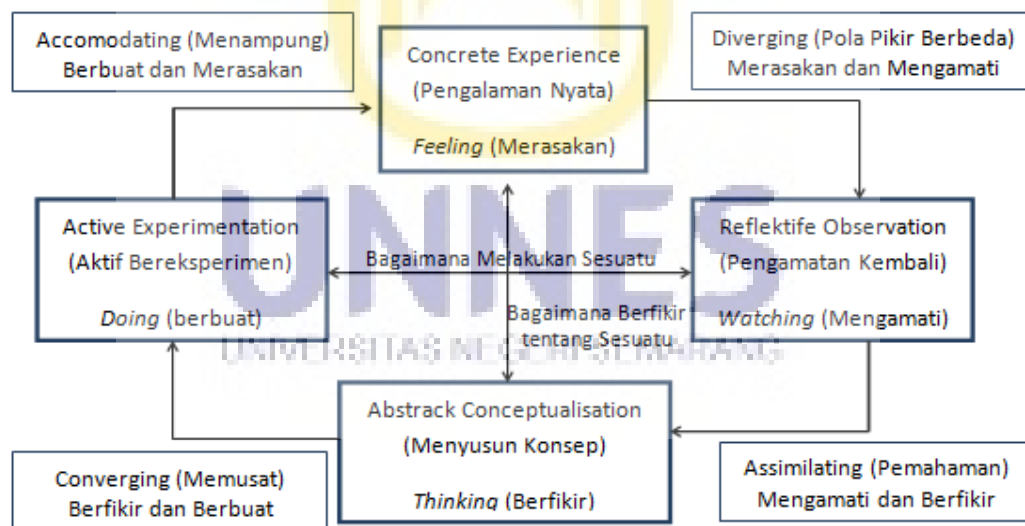
Dijelaskan oleh Muijs dan Reynold (2008) pada Bernadus (2012:14) bahwa teknologi informasi dan komunikasi dapat membantu belajar siswa dalam: menyajikan informasi, mengakses dan menangani informasi,

modeling dan kontrol, interaktifitas, dan memperluas sekolah ke rumah. Memanfaatkan TIK (Teknologi Informasi dan Komunikasi) dapat dilakukan sebagai sistem belajar yang mandiri atau juga dapat digunakan dengan mengkombinasikannya pada pembelajaran langsung yang memerlukan guru. Model penyampaian materi pada pembelajaran berbasis komputer terbagi menjadi: latihan dan praktik, tutorial, serta simulasi.

Banyak metode pelatihan yang digunakan dalam pembelajaran informasi dan komunikasi, seperti: *e-learning*, *intelligent tutoring*, internet, intranet, *distance learning*, dan *simulations and virtual reality*. Metode *simulations and virtual reality* merupakan metode yang cocok untuk proses pembelajaran dengan sistem yang kompleks seperti sistem mikroprosesor khususnya Zilog-80. Alur kerja dan operasi dari sistem mikroprosesor Zilog-80 dapat disimulasikan secara *virtual reality* menggunakan aplikasi Z80-Simulator dalam proses pembelajaran. Pembelajaran berbasis komputer menggunakan aplikasi Z80-Simulator ini dapat dilakukan dengan memberikan *jobsheet* yang berisi langkah-langkah percobaan yang dikemas dalam modul praktikum untuk membantu kegiatan belajar siswa dan sebagai fasilitator dalam proses pembelajaran. Pembelajaran menggunakan komputer ini dapat membuat siswa untuk belajar dengan kecepatan masing-masing dan dapat memberikan umpan balik langsung karena memiliki sistem panduan yang terintegrasi, sehingga siswa mampu mengoreksi kesalahan pribadi dan mengetahui kekurangan yang dimiliki untuk dasar perbaikan.

c. Model Pembelajaran *Experiential Learning*

Model pembelajaran *experiential learning* dikembangkan dari *Experiential Learning Theory* (ELT) oleh David Kolb di-era 1980-an. Pembelajaran *experiential learning* adalah pembelajaran yang membuat siswa aktif untuk membangun pengetahuan dan keterampilan melalui pengalaman secara langsung. Dalam teorinya Kolb menjelaskan terdapat empat aktifitas yang dilakukan dalam proses pembelajaran *experiential learning* yakni merasakan pengalaman nyata, melihat atau mengamati, menyusun konsep berdasar pengalaman dan aktif bereksperimen. Keempat aktifitas ini saling terhubung yang dapat digambarkan dalam sebuah siklus, seperti yang ditunjukkan pada Gambar1. Siklus aktifitas dalam *experiential learning* pada *Kolb's Learning Styles* berikut:



Gambar 1. Siklus Aktifitas Dalam *Experiential Learning* pada *Kolb's Learning Styles*
Sumber : Jon Ord (2012 : 59)

Empat aktifitas dalam proses belajar *experiential learning* akan dilakukan oleh siswa untuk membangun dan memperoleh pengalaman

belajar. Sesuai dengan Gambar1. uraian aktifitas yang dilakukan oleh siswa dalam proses belajar *experiential learning* dapat disusun dalam tabel sebagai berikut:

Tabel 1. Aktifitas Yang Dilakukan Siswa Dalam Proses Belajar *Experiential Learning*

Aktifitas	Uraian	Pengutamaan
<i>Concrete Experience (CE)</i>	Siswa melibatkan diri sepenuhnya dalam pengalaman baru	<i>Feeling</i> (Perasaan)
<i>Reflection Observation (RO)</i>	Siswa Mengobservasi merefleksi atau memikirkan pengalaman dari berbagai segi	<i>Watching</i> (Pengamatan)
<i>Abstract Conceptual (AC)</i>	Siswa menyusun konsep-konsep dari observasi menjadi teori	<i>Thinking</i> (Berfikir)
<i>Active Experimentation (AE)</i>	Siswa menggunakan teori untuk memecahkan masalah dan mengambil keputusan	<i>Doing</i> (Berbuat)

Sumber : diadaptasi dari Jon Ord (2012)

Pembelajaran *experiential learning* menitik beratkan pada pengalaman yang akan dialami oleh siswa, dengan siswa terlibat langsung dalam proses belajar mengajar untuk mengkonstruksi pengalaman nyata menjadi pengetahuan yang solid. Pembelajaran ini dapat digunakan dalam pembelajaran praktikum yang menuntut siswa untuk berlatih dan melakukan percobaan-percobaan terkait dengan materi belajar seperti halnya Praktik Sistem Mikroprosesor.

B. MODUL

1. Pengertian Modul

Modul adalah suatu paket pengajaran yang berkenaan dengan suatu unit terkecil terhadap mata pelajaran tersebut. Daryanto (2013:9) menjelaskan bahwa modul merupakan salah satu bentuk bahan ajar yang dikemas secara sistematis, di dalamnya memuat seperangkat pengalaman belajar terencana dan terdesain untuk membantu peserta didik menguasai tujuan pembelajaran yang spesifik. Ditegaskan pula dalam KBBI modul adalah kegiatan atau program belajar-mengajar yang dapat dipelajari oleh murid dengan bantuan yang minimal dari guru pembimbing, meliputi perencanaan tujuan yang akan dicapai secara jelas, penyediaan materi pelajaran, alat yang dibutuhkan, serta alat untuk menilai, mengukur keberhasilan murid dalam penyelesaian pelajaran.

Sebagai unit program pengajaran yang terkecil modul secara rinci menggariskan :

- a. Tujuan instruksional khusus
- b. Tujuan instruksional umum
- c. Pokok-pokok materi yang dipelajari
- d. Fungsi satuan dalam kesatuan program yang luas.
- e. Alat dan sumber yang dipakai
- f. Peran guru dalam proses belajar mengajar.
- g. Kegiatan belajar mengajar yang harus dilakukan murid.

2. Karakteristik Modul

Modul merupakan salah satu bentuk bahan ajar, agar pembelajaran berjalan sesuai dengan tujuan yang ingin dicapai menurut Daryanto (2013:9) modul hendaknya memiliki karakteristik sebagai berikut:

a. *Self Instruction*

Melalui modul peserta belajar atau peserta didik dapat belajar secara mandiri atau membelajarkan diri sendiri tidak tergantung pada pihak lain.

b. *Self Contained*

Keseluruhan materi pembelajaran dari satu unit kompetensi atau sub kompetensi yang dipelajari terdapat di dalam satu modul secara utuh. Hal ini bertujuan supaya peserta didik memiliki kesempatan untuk memperlajari pembelajaran secara tuntas.

c. *Stand Alone* (Berdiri Sendiri)

Modul yang dikembangkan tidak bergantung pada media lain atau tidak harus digunakan bersama-sama dengan media pembelajaran lain.

d. *Adaptive*

Modul hendaknya memiliki daya adaptasi atau dapat menyesuaikan terhadap perkembangan ilmu pengetahuan dan teknologi.

e. *User Friendly* (Bersahabat)

Modul hendaknya dapat bersahabat dengan pemakai atau peserta didik. Setiap instruksi dan paparan informasi yang ditampilkan

mampu/bersifat untuk membantu dan bersahabat dengan pemakainya, termasuk kemudahan pemakaian dalam merespon, mengakses sesuai dengan keinginan.

3. Tahapan Penyusunan Modul

Daryanto (2013:15) menjelaskan modul disusun berdasarkan prinsip-prinsip pengembangan yang meliputi analisis kebutuhan, pengembangan desain, implementasi, penilaian, evaluasi dan validasi, serta jaminan kualitas. Adapun penjelasannya diuraikan sebagai berikut:

a. Analisis Kebutuhan Modul

Analisa kebutuhan modul merupakan kegiatan menganalisis silabus dan RPP untuk memperoleh informasi modul yang dibutuhkan peserta didik dalam mempelajari kompetensi yang diprogramkan. Tujuan dari analisis ini ialah untuk mengidentifikasi dan menentukan judul modul dari suatu pengembangan program tertentu. Analisis kebutuhan modul ini dapat dilakukan dengan tahapan prosedur, sebagai berikut:

- 1) Menetapkan batasan waktu lingkup kegiatan belajar satuan program, pada penelitian ini adalah program satu semester.
- 2) Memeriksa apabila program kegiatan yang akan dibuat telah ada atau belum. Bila sudah pernah ada dapat digunakan sebagai bahan referensi.
- 3) Mengidentifikasi standar kompetensi yang akan dipelajari untuk memperoleh materi pembelajaran yang sesuai.

- 4) Berdasarkan materi pembelajaran yang telah diperoleh, kemudian disusun dalam satuan unit bahan ajar dengan setiap unit diberi nama dan dijadikan sebagai judul modul.
- 5) Dari daftar satuan unit tersebut diidentifikasi ketersediaanya dilingkungan sekolah. Kemudian disusun modul berdasarkan prioritas kebutuhannya.

b. Desain Modul

Desain modul yang dibuat berdasarkan Rencana Pelaksanaan Pembelajaran yang disusun oleh guru, di dalamnya memuat strategi pembelajaran dan media yang digunakan, garis besar materi pembelajaran dan metode penilaian serta perangkatnya.

Mendesain modul diawali dengan penyusunan konsep modul. Konsep modul inilah yang akan dikembangkan dan diuji coba serta dilakukan proses validasi. Untuk mendesain modul dapat dilakukan dengan tahapan-tahapan sebagai berikut:

- 1) Menetapkan kerangka bahan dari modul yang akan disusun.
- 2) Menetapkan tujuan akhir atau kemampuan yang akan dicapai peserta didik setelah mempelajari modul.
- 3) Menetapkan sistem evaluasi yang akan dibuat yang terdiri dari ketentuan/skema, metoda dan perangkat evaluasi.
- 4) Menetapkan garis-garis besar materi untuk mencapai tujuan yang telah ditetapkan.

- 5) Materi disusun berdasarkan prinsip, fakta penting yang terkait langsung dan mendukung pencapaian kompetensi yang harus dikuasai peserta didik.
- 6) Menetapkan tugas, soal, serta latihan-latihan/praktik yang harus dikerjakan oleh peserta didik.
- 7) Menetapkan evaluasi yang digunakan dalam mengukur kemampuan peserta didik dalam penguasaan materi modul.
- 8) Membuat kunci jawaban dari soal, tugas dan latihan yang disusun.

c. Implementasi

Implementasi modul dalam kegiatan belajar dilaksanakan sesuai dengan alur yang telah disusun dalam modul. Bahan, alat, media dan lingkungan belajar yang dibutuhkan dalam kegiatan pembelajaran diupayakan dapat dipenuhi sesuai dengan rancangan yang telah disusun agar tujuan pembelajaran dapat tercapai.

d. Penilaian

Penilaian digunakan untuk mengetahui tingkat penguasaan peserta didik setelah mempelajari seluruh materi dalam modul. Pelaksanaan penilaian disesuaikan dengan apa yang telah dirumuskan di dalam modul. Penilaian diambil berdasarkan instrumen yang telah dirancang sebelumnya.

e. Evaluasi dan Validasi

Modul yang telah disusun kemudian dievaluasi untuk mengukur apakah implementasi pembelajaran dengan modul dapat dilaksanakan sesuai dengan desain pengembangannya. Evaluasi ini ditujukan pada guru

dan peserta didik yang terlibat langsung dengan proses implementasi modul tersebut.

Selain dilakukan evaluasi dilakukan pula validasi untuk menguji kesuaian modul dengan kompetensi target belajar. Validasi dapat dilakukan dengan meminta bantuan ahli yang menguasai kompetensi bidang yang dipelajari.

f. Jaminan Kualitas

Untuk menjamin bahwa modul telah disusun sesuai dengan ketentuan dalam pengembangan modul maka selama pembuatannya, modul perlu dipantau agar modul disusun berdasarkan desain yang ditetapkan. Untuk penjaminan mutu dapat dikembangkan suatu instrumen untuk menilai kualitas modul tersebut.

C. PRAKTIK SISTEM MIKROPROSESOR

1. Pengertian Praktik Sistem Mikroprosesor

Praktik Sistem Mikroprosesor adalah salah satu mata kuliah wajib yang harus ditempuh oleh mahasiswa Pendidikan Teknik Elektro UNNES. Pada mata kuliah ini mahasiswa mempelajari tentang perancangan, pemrograman, serta mengoperasikan mikroprosesor seperti Mikroprosesor Zilog 80.

Sistem adalah satu kesatuan yang terdiri dari beberapa komponen atau elemen yang saling berubungan dalam aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Mikroprosesor merupakan suatu *chip*

(IC = *Integrated Circuit*) yang lebih dikenal sebagai CPU (*Central Processing Unit*). Di dalamnya terkandung rangkaian ALU (*Aritmatic Logic Unit*), rangkaian CU (*Control Unit*) dan *register-register* yang berfungsi sebagai pusat pengelola data-data digital. Maka dapat disimpulkan bahwa sistem mikroprosesor merupakan kesatuan dari komponen-komponen (CPU, memori unit, input/output) yang saling berhubungan dalam mengelola informasi atau data untuk mencapai tujuan suatu program tertentu.

2. Mikroprosesor Zilog 80

a. Pengertian Mikroprosesor Zilog 80

CPU Z80 adalah mikroprosesor generasi ke-empat yang dikembangkan untuk keperluan perhitungan. Mikroprosesor ini menawarkan *throughput* sistem yang lebih tinggi dan penggunaan memori yang efisien dibandingkan dengan mikroprosesor yang sama pada generasi kedua dan ketiga. Z80 CPU bekerja pada beberapa frekuensi (tergantung versi), seperti 2.5 MHz, 4 MHz, 6 MHz dan 8MHz, serta memiliki instruksi yang lebih banyak dari pada mikroprosesor intel 8080 yang terkenal sebelumnya, yakni terdiri dari 158 set instruksi.

Register internalnya terdiri dari 208-bit memori baca/tulis yang bisa diakses oleh *programmer*. *Register-register* tersebut termasuk dua set enam *register* serba-guna yang bisa digunakan secara sendiri-sendiri (*individual*) sebagai *register* 8-bit atau sebagai pasangan

register 16-bit. Sebagai tambahan, ada dua set *register* lagi, yaitu akumulator dan *register* bendera (flag). Grup perintah “Exchange” membuat set *register* utama atau *register* alternatif bisa diakses oleh *programmer*. Set alternatif memungkinkan operasi dalam mode nampak-tersembunyi (*foreground-background*) atau bisa digunakan sebagai cadangan bagi tanggapan interupsi sangat cepat.

CPU Z80 juga tersusun atas sebuah *Stack Pointer*, *Program Counter*, dua buah *Register* indeks, sebuah *Register Refresh (counter)*, dan sebuah *register* interupsi. CPU Z80 juga sangat mudah disertakan pada sistem karena hanya memerlukan sumber tegangan tunggal +5V. CPU Z80 didukung oleh perluasan keluarga pengontrol periferai.

b. Bagian-bagian Mikroprosesor Z80

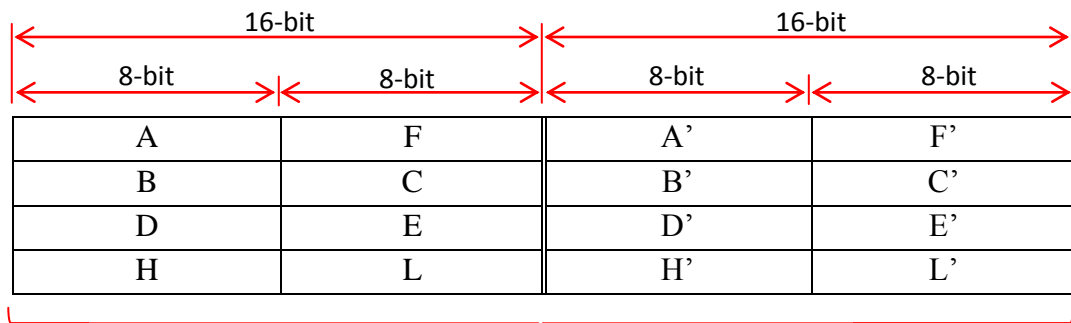
Bagian-bagian dari sebuah mikroprosesor dapat dipisahkan menjadi lima buah, meliputi: masukan (*input*), memori unit, aritmatik dan logika, kendali (CPU) dan keluaran (*output*).

Masukan dan keluaran biasa disebut sebagai unit *input output (i/o)*. Albert Paul & Ija May On (1994:8) menjelaskan *Input/Output* adalah bagian yang digunakan untuk memasukkan program atau data ke dalam unit pemroses (CPU), umumnya berupa tombol ketik yang berisi karakter, huruf atau bilangan namun dapat pula berupa sensor atau peralatan lainnya. Serta merupakan bagian yang mengeluarkan

hasil dari proses pengelolaan program atau data dari unit pemroses ke luar, bagian ini dapat berupa tampilan *display*, suara, dll.

Data-data dari unit masukan dan keluaran semuanya dimasukkan dan catatan pada unit memori. Kembali Albert Paul & Ija May On (1994:8) menjelaskan bahwa unit memori berfungsi untuk menyimpan program dan data sebelum komputer bekerja. Memori juga menyimpan bagian-bagian dari hasil penyelesaian selama komputer beroperasi. Di dalam mikroprosesor sendiri terdapat sebuah unit kecil memori yang disebut *register*. *Register* merupakan salah satu memori dalam CPU yang dapat dibaca dan ditulis (*Read/Write*) oleh sebuah mikroprosesor digunakan sebagai tempat penyimpanan data sementara untuk keperluan pemrosesan program agar proses komputasi lebih cepat. Mikroprosesor Z80 memiliki *register* sebanyak 208-bit (26 x 8-bit) di dalam CPU. Seperti yang ditunjukkan pada Gambar 2, *register* Z80 terbagi menjadi 18 buah *register* 8-bit, dan 4 buah *register* 16-bit.

Terlihat pada Gambar 2, terdapat *register* khusus yang berfungsi untuk melakukan tugas-tugas khusus dalam proses pemrograman. Register A (*Accumulator*) yang berfungsi sebagai *register* penampung hasil operasi ALU (*Arithmetic Logic Unit*), kemudian *register* F sebagai *register* status dari operasi ALU.



Register General Purpose

A (<i>Accumulator</i>)	F (<i>Flag Register</i>)
IX (<i>Index Register</i>)	
IY (<i>Index Register</i>)	
SP (<i>Stack Pointer</i>)	
PC (<i>Program Counter</i>)	
I (<i>Interrupt Vector</i>)	R (<i>Memory Refresh</i>)

Register Special Purpose

Gambar 2. Tampilan Register Mikroprosesor Z80

Tabel 2. Keterangan *Register* Mikroprosesor Z80

	<i>Register</i>	Ukuran (Bits)	Keterangan
A, A'	<i>Accumulator</i>	8-bit	Menyimpan Operand atau Hasil Operasi
F, F'	<i>Flag/Bendera</i>	8-bit	Menentukan set instruksi
B, B'	<i>General Purpose</i>	8-bit	Dapat digunakan terpisah (8-bit) atau sebagai 16-bit dengan <i>register C</i>
C, C'	<i>General Purpose</i>	8-bit	Dapat digunakan terpisah (8-bit) atau sebagai 16-bit dengan <i>register B</i>
D, D'	<i>General Purpose</i>	8-bit	Dapat digunakan terpisah (8-bit) atau sebagai 16-bit dengan <i>register E</i>
E, E'	<i>General Purpose</i>	8-bit	Dapat digunakan terpisah (8-bit) atau sebagai 16-bit dengan <i>register D</i>
H, H'	<i>General Purpose</i>	8-bit	Dapat digunakan terpisah (8-bit) atau sebagai 16-bit dengan <i>register L</i>

Lanjutan Tabel 1. Keterangan Register Mikroprosesor

<i>Register</i>		Ukuran (Bits)	Keterangan
L, L'	<i>General Purpose</i>	8-bit	Dapat digunakan terpisah (8-bit) atau sebagai 16-bit dengan <i>register H</i>
I	<i>Interrupt Register</i>	8-bit	Menyimpan 8-bit alamat memori tertinggi untuk proses interupsi
R	<i>Refresh Register</i>	8-bit	Menyediakan <i>refresh dynamic memory</i> bagi pengguna
IX	<i>Index Register</i>	16-bit	Digunakan untuk alamat index
IY	<i>Index Register</i>	16-bit	Digunakan untuk alamat index
SP	<i>Stack Pointer</i>	16-bit	Berisi alamat memori tertinggi Stack
PC	<i>Program Counter</i>	16-bit	Berisi alamat memori intruksi selanjutnya

3. Instruksi-Instruksi Pemrograman Zilog 80

Untuk membangun sebuah program diperlukan perintah-perintah yang mendasari program tersebut agar dapat berjalan. Sebuah mikroprosesor menggunakan bahasa mesin (*machine language*) untuk memahami sebuah instruksi. Bahasa mesin sendiri diimplementasikan dalam sebuah pola-pola bit yang nantinya akan diartikan tersendiri oleh *instruction manager* untuk diproses CPU.

Penggunaan pola-pola bit sangat susah untuk dipahami oleh manusia, maka digunakan suatu set kata pendek untuk mewakili pola bit tersebut yang disebut sebagai bahasa *assembly*. Agar bahasa *assembly* ini dapat dipahami juga oleh mesin maka dibutuhkan piranti penerjemah yakni *assembler*. Setelah instruksi diterjemahkan maka akan dimasukkan ke dalam memori prosesor yang kemudian dapat dengan mudah dieksekusi.

Mikroprosesor Z80 memiliki setidaknya 158 macam instruksi. Instruksi ini terdiri dari berbagai kumpulan seperti instruksi transfer data (*Load and*

Exchange), instruksi *block transfer*, instruksi *Arithmetic and Logical*, instruksi *Rotate and Shift*, instruksi *Bit Manipulation*, instruksi *looping (jum, call and return)*, instruksi *I/O* dan instruksi *Basic CPU Control*.

a. Instruksi transfer data

Transfer data pada sebuah mikroprosesor sama halnya dengan perintah *copy data*. Z-80 biasanya menggunakan perintah LD (*Load*) untuk melakukan operasi transfer data. Instruksi *load* dapat digunakan untuk mentransfer data 8-bit maupun 16-bit. Terdapat 134 perintah *load* yang dimiliki oleh Z80.

Operasi *Load* sebagai operasi transfer data terdiri dari dua (2) buah *operand*. *Operand* awal menunjukkan lokasi data yang akan disimpan (pada memori atau *register*), disebut sebagai *Destination* (tujuan). *Operand* kedua menunjukkan lokasi asal dari data yang akan ditransfer, disebut sebagai *Source* (sumber). *Operand* dapat berupa alamat memori, *register* atau *data immediate* (data langsung) yang memiliki lebar 8-bit atau 16-bit.

Data dengan lebar 16-bit maupun 8-bit dapat ditransfer dengan berbagai kemungkinan. Beberapa kemungkinan yang terjadi, seperti:

1. Transfer data dari *register* ke *register*
2. Transfer data dari memori ke *register*
3. Transfer data *immediate* (langsung) ke *register*
4. Transfer data dari *register* ke memori

5. Transfer data dari memori ke memori
6. Transfer data *immediate* (langsung) ke memori

Selain menggunakan perintah *load programmer* juga dapat melakukan transfer data dengan menggunakan 6 jenis perintah *Exchange* (EX dan EXX), kemudian 12 jenis *PUSH* dan *POP* yang berhubungan dengan transfer data pada *stack pointer*. Perintah *exchange* memungkinkan *programmer* untuk dapat mengakses *register* alternatif dan menggunakan *register* utama untuk keperluan lainnya. Sebagai contoh dapat dilihat pada Tabel 3 berikut ini.

Tabel 3. Contoh penulisan instruksi transfer data

Mnemonic	Symbol Operasi	Keterangan
LD A, B	$A \leftarrow B$	Register ke Register (8-bit)
LD A, (HL)	$A \leftarrow (HL)$	Memori ke Register (8-bit)
LD B, 44H	$B \leftarrow 44H$	Data immediate ke Register (8-bit)
LD (HL), A	$(HL) \leftarrow A$	Register (8-bit) ke Memori
LDD	$(DE) \leftarrow (HL)$ $DE \leftarrow DE-1$ $HL \leftarrow HL-1$ $BC \leftarrow BC-1$	Memori ke Memori ; nilai reg. DE dikurangi 1 ; nilai reg. HL dikurangi 1 ; nilai reg. BC dikurangi 1
LD (HL), 5AH	$(HL) \leftarrow 5AH$	Data immediate (8-bit) ke Memori
EX DE, HL	$DE \leftrightarrow HL$	Tukar data register dengan register
EX (SP), HL	$H \leftrightarrow (SP+1)$ $L \leftrightarrow (SP)$	Tukar data register dengan memori

b. Instruksi Aritmatik dan Logika

Instruksi aritmatik dan logika pada mikroprosesor Z80 dilaksanakan dalam *register A (Accumulator)*. Penggunaan *register* lain dapat digunakan sebagai *operand* bersamaan dengan *register A* pada instruksi LD. Transfer data antara memori ke *register* atau sebaliknya yang melibatkan reg.A biasa digunakan *register HL, IX dan IY* sebagai penunjuk alamat memori. Dicontohkan beberapa instruksi aritmatik dan logika pada Tabel 4 berikut:

Tabel 4. Contoh instruksi Aritmatika dan Logika

Instruksi/Mnemonik	Keterangan
ADD A, A	Data pada <i>register A</i> ditambahkan pada <i>register A</i> sendiri ($A+A$), dapat diartikan diduakalikan atau digeser 1-bit ke kiri.
ADD HL, BC	Data pada <i>register BC</i> di tambahkan ke <i>register HL</i> dan hasil disimpan pada reg. HL ($HL \leftarrow HL + BC$)
ADC A, B	Data <i>register B</i> ditambahkan pada <i>register A</i> bersamaan dengan <i>Carry</i> dan disimpan di reg.A ($A \leftarrow A + B + Carry$)
SUB C	Data <i>register A</i> dikurangi data <i>register C</i> ($A \leftarrow A - C$)
SBC (HL)	Data <i>register A</i> dikurangi isi memori yang ditunjuk HL dikurangi kembali dengan <i>Carry</i> ($A \leftarrow A - (HL) - Carry$)
INC D	Menambah isi <i>register D</i> dengan 1 ($D \leftarrow D + 1$)
INC BC	Menambah isi <i>register BC</i> dengan 1 ($BC \leftarrow BC + 1$)
INC (HL)	Menambah isi alamat memori yang ditunjuk HL dengan 1

Lanjutan tabel 4. Contoh instruksi aritmatika dan logika

Instruksi/Mnemonik	Keterangan
DEC B	Mengurangi isi <i>register</i> B dengan 1 ($B \leftarrow B - 1$)
DEC HL	Mengurangi isi <i>register</i> HL dengan 1 ($HL \leftarrow HL - 1$)
DEC (HL)	Mengurangi isi data alamat memori yang ditunjuk HL dengan 1
AND C	Operasi logika “AND” antara <i>register</i> C dan <i>register</i> A dengan hasil disimpan di reg. A
OR 67H	Operasi logika “OR” antara <i>register</i> A dan data 67H dengan hasil disimpan di reg. A
EXOR (HL)	Operasi logika “OR-Exklusif” antara <i>register</i> A dan isi data pada alamat memori yang ditunjuk HL dengan hasil disimpan di reg. A

Selain penggunaan instruksi penjumlahan (ADD dan ADC) serta pengurangan (SUB dan SBC), terdapat pula instruksi khusus dalam proses aritmatik. Beberapa instruksi khusus aritmatik yang disediakan seperti:

1) DAA (*Decimal Adjust Accumulator*)

Instruksi DAA digunakan untuk merubah isi *register* A menjadi bentuk BCD serta untuk memberikan factor koreksi pada saat bekerja dengan bilangan desimal.

Cara kerjanya ialah bila pada (bit-3, bit-2, bit-1, bit-0) > 9 (ada *half carry*), maka bit-3, bit-2, bit-1, bit-0 akan ditambah 6 = 0110. Kemudian bila pada (bit-7, bit-6, bit-5, bit-4) > 9 (ada *carry*), maka bit-7, bit-6, bit-5, bit-4 akan ditambah 6 = 0110. Sebagai contoh:

Penjumlahan Desimal		Penjumlahan Biner
29_{10}	=	$0010\ 1001_2$
26_{10}	=	$0010\ 0110_2$
55_{10}	\neq	$0100\ 1111_2 = 4F_{16}$ +

Hasil BCD seharusnya 55 namun tidak sesuai maka oleh instruksi

DAA dikoreksi menjadi:

Koreksi DAA	
$0100\ 1111_2$	= $4F_{16}$
$0000\ 0110_2$	+
$0100\ 0101_2$	= 55

2) CPL (*Complement Accumulator*/komplemen 1)

Instruksi *complement* digunakan untuk merubah isi *register A* (*Accumulator*) menjadi komplemen 1, dilakukan dengan cara menginverskan seluruh bit yang ada pada *register*. Sebagai contoh:

LD A, 29H ; isi *register A* = 29H = $0010\ 1001_2$
 CPL ; reg. A jadi = D6H = $1101\ 0110_2$

3) NEG (*Negate Accumulator*/komplemen 2)

Instruksi NEG digunakan untuk merubah isi *register A* menjadi bentuk negatifnya atau komplemen 2, hal ini dilakukan dengan cara menginverskan seluruh bit yang ada pada *register* kemudian ditambahkan dengan 1. Sebagai contoh:

LD A, 98H ; isi *register A* = 98H = $1001\ 1000_2$
 NEG ; reg. A jadi = 68H = $0110\ 1000_2$

Proses instruksi NEG	
	A = 98H = $1001\ 1000_2$
Dikomplemenkan	A = 67H = $0110\ 0111_2$
Ditambah 1 (+1)	$= 01H = 0000\ 0001_2$ +
Hasil negatif	A = 68H = $0110\ 1000_2$

c. Instruksi Loop dan percabangan

Sebuah program selalu melaksanakan setiap instruksi dengan urutan yang runtut, namun terkadang pada sebuah program terdapat instruksi yang membutuhkan inisialisasi kondisi sebelum diproses. Inisialisasi ini menciptakan beberapa opsi/pilihan dan akhirnya menimbulkan suatu kondisi percabangan. Hal ini membuat program harus meloncat pada suatu alamat tertentu sesuai opsi yang dipilih untuk melanjutkan pelaksanaan program. Seringkali instruksi percabangan digunakan untuk keperluan loop atau program perulangan.

Instruksi percabangan biasa menggunakan perintah *jump*, yang dapat dikategorikan menjadi *jump* bersyarat, *jump* tanpa syarat, *jump* absolut dan *jump* relatif.

1) *Jump* bersyarat

Instruksi *jump* bersyarat akan melompat ke alamat tertentu bila suatu syarat terpenuhi. Syarat ini mengacu pada kondisi status data dari *register flag*. Menggunakan fungsi ini suatu mikrokomputer dapat melaksanakan beberapa syarat yang diberikan oleh user. Hal ini juga merupakan suatu fungsi yang mutlak untuk keperluan perancangan program loop. Instruksi *jump* bersyarat diantaranya seperti pada Tabel 5.

Tabel 5. Instruksi-instruksi *jump* bersyarat

Instruksi	Keterangan
Jump absolut bersyarat	
JP C, XXXX	Jika <i>Carry</i> = 1, loncat ke alamat XXXX
JP NC, XXXX	Jika <i>Carry</i> = 0, loncat ke alamat XXXX
JP Z, XXXX	Jika <i>Zerro</i> = 1, loncat ke alamat XXXX
JP NZ, XXXX	Jika <i>Zerro</i> = 0, loncat ke alamat XXXX
JP PE, XXXX	Jika <i>parity</i> = 1, loncat ke alamat XXXX
JP PO, XXXX	Jika <i>parity</i> = 0, loncat ke alamat XXXX
JP M, XXXX	Jika <i>sign</i> = 1, loncat ke alamat XXXX
JP P, XXXX	Jika <i>sign</i> = 0, loncat ke alamat XXXX
Jump Relatif bersyarat	
JR C, XX	Jika <i>Carry</i> = 1, loncat pada instruksi $PC \leftarrow PC + XX$
JR NC, XX	Jika <i>Carry</i> = 0, loncat pada instruksi $PC \leftarrow PC + XX$
JR Z, XX	Jika <i>Zerro</i> = 1, loncat pada instruksi $PC \leftarrow PC + XX$
JR NZ, XX	Jika <i>Zerro</i> = 0, loncat pada instruksi $PC \leftarrow PC + XX$
Jump Relatif Khusus bersyarat	
DJNZ, XX	Reg. B dikurangi 1(-1). Jika reg. B \neq 0, lompat ke alamat $PC + XX$ dan jika reg B = 0 lanjutkan progrm.

2) *Jump* tanpa syarat

Instruksi *jump* tanpa syarat tidak membutuhkan perubahan status pada *register flag*. Instruksi ini melakukan lompatan berdasarkan data yang diberikan. Beberapa instruksi dari *jump* tanpa syarat seperti:

JP, XXXX ; meloncat ke alamat memori XXXX

JR, XX ; meloncat pada instruksi $PC \leftarrow PC + XX$

4. Z80-Simulator

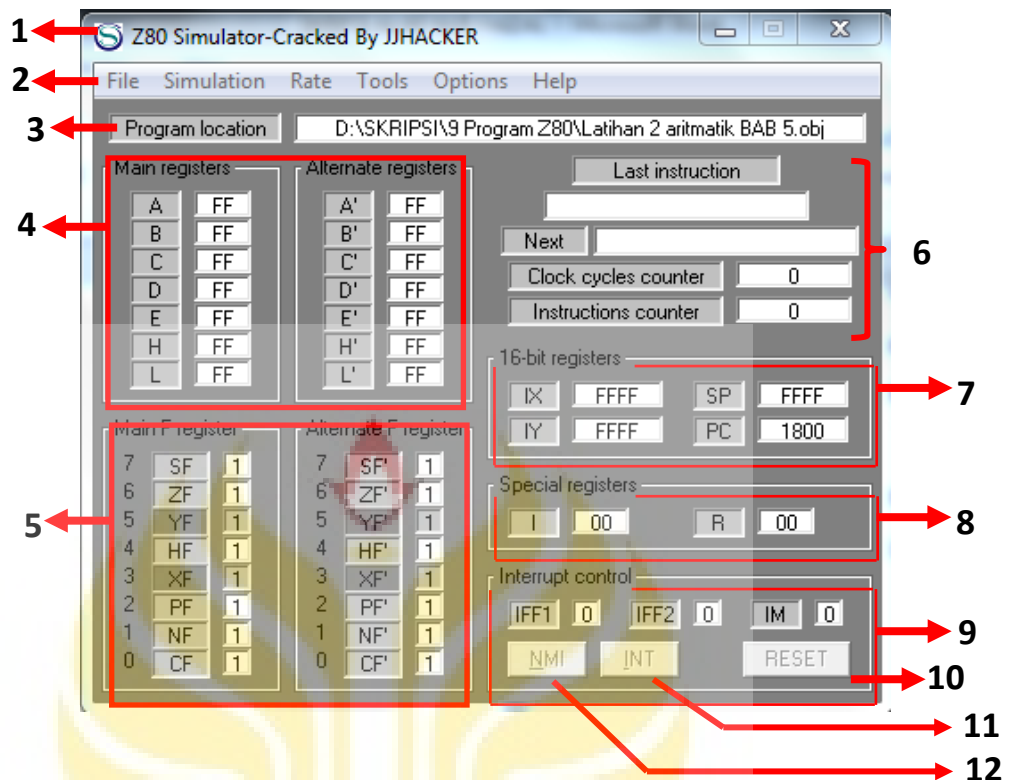
a. Pengertian Z80-Simulator

Z80 Simulator merupakan sebuah *software* simulasi yang digunakan untuk membantu *user* atau *programmer* dalam menganalisis program

yang akan digunakan/dibuat khususnya pada mikroprosesor Zilog80. Simulasi ini memberikan kemudahan kepada *programmer* untuk melihat proses kerja dari program yang sedang dikerjakan. Simulasi ini hanya dapat menunjukkan perubahan-perubahan *register* yang digunakan. Penggunaanya pun menggunakan bahasa pemrograman atau *assembly*. Selain dapat digunakan untuk melihat proses kerja prosesor dan perubahan *register*, program aplikasi ini juga berguna untuk mengoreksi kesalahan-kesalahan yang mungkin terjadi dalam pembuatan sebuah program.

b. Keterangan tampilan Z80-Simulator

Simulator Z80 banyak macamnya namun yang akan dibahas ialah Z80 simulator yang dikeluarkan oleh OSHONSOFT. Tampilannya yang sederhana dan simpel memberikan kemudahan untuk melihat proses dan hasil pemrograman Z80 secara langsung dan cepat. Terlihat pada Gambar.3 jendela utama Z80 simulator memiliki beberapa bagian pokok. Bagian ini diantaranya seperti *menu bar*, *program locator box*, *register status box*, *flag register status box*, *instruction status box*, *16-bit register box*, dan tombol-tombol pengatur seperti *reset*.



Gambar 3. Tampilan Jendela Utama Simulator Z80

Keterangan :

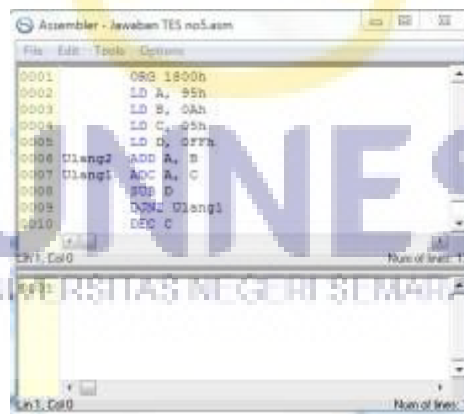
- 1) *Title Bar*, menunjukkan nama program dan status dari aplikasi Z80 Simulator.
- 2) *Menu Bar*, pilihan menu yang ada pada aplikasi Z80 Simulator.
- 3) *Program Locator Box*, menunjukkan alamat program yang sedang di-load pada aplikasi Z80 Simulator.
- 4) *Register Status Box*, menampilkan isi data general register pada aplikasi Z80 Simulator.
- 5) *Flag Register Status Box*, menampilkan isi data register bendera pada aplikasi Z80 Simulator.

- 6) *Instruction status Box*, menampilkan instruksi yang sedang dikerjakan dan yang akan dikerjakan aplikasi Z80 Simulator. Juga menampilkan perhitungan eksekusi instruksi dan *clock* dari setiap instruksi yang dikerjakan.
- 7) *16-bit register box*, menampilkan isi data *register* 16-bit IX dan IY serta SP (*Stack Pointer*), dan PC (*Program Counter*).
- 8) *Special Register box*, menampilkan isi data *register* reset yang berfungsi untuk melakukan penyegaran berkala (*refresh*) pada memori dinamik selama CPU mengkode dan melaksanakan pengambilan instruksi dari memori dan *register interrupt* untuk menampilkan kondisi interupsi suatu alat *peripheral* atau eksternal.
- 9) *Interrupt Control*, menampilkan kondisi mode interupsi IM (*Interrupt Mode*), serta *register interrupt* flip-flop (IFF) yang menunjukkan status interupsi.
- 10) *Tombol Reset*, untuk me-*reset* memori dinamik secara manual.
- 11) *Tombol INT*, digunakan untuk melakukan interupsi secara manual.
- 12) *Tombol NMI (Non Maskable Interrupt)*, merupakan mode interupsi yang tidak dapat dicegah.

c. Peralatan penunjang Z80-Simulator

1) *Assembler Tools*

Assembler adalah peralatan yang digunakan untuk menerjemahkan program *assembly* menjadi bahasa mesin. *Tool* ini berfungsi untuk mengedit atau membuat *file* yang nantinya diassemble dan dapat disimulasikan dengan di-load ke memori simulator terlebih dahulu. *File* yang disimpan akan berekstensi ASM. Setelah proses *assemble* selesai maka akan dihasilkan dua buah *file*, pertama berekstensi OBJ (*binary image*) yang dapat langsung di load ke memori CPU, kedua berekstensi LST yang merupakan *assembler listing*, dapat digunakan untuk *debugger*. *File* berekstensi HEX intel juga dapat dihasilkan jika diperlukan, dapat pula menggunakan *assembler* eksternal seperti TASM dan MASM.

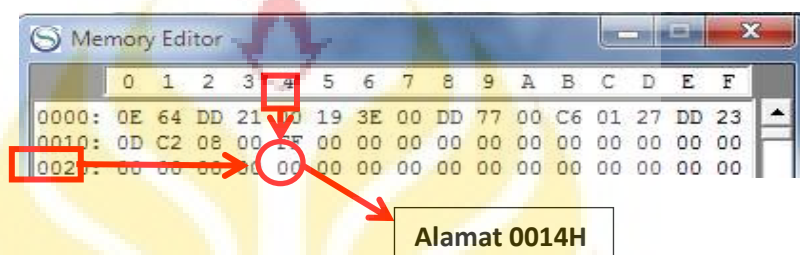


Gambar 4. Tampilan Jendela *Assembler*

2) *Memory Editor*

Berfungsi untuk menampilkan isi memori. Nilai dari setiap alamat memori akan terisi secara otomatis sesuai dengan nilai *opcode*

program atau juga dapat diubah secara manual dengan mengklik dan menuliskan isinya secara langsung. Untuk melihat alamat memori yang akan dituju dapat diketahui dengan melihat titik poin pertemuan antara alamat awal (*horizontal*) dan alamat akhir (*vertical*). Seperti yang ditunjukkan oleh gambar berikut. Sebagai contoh ingin melihat alamat memori 0014H:



Gambar 5. Tampilan Jendela *Memory Editor*

3) *Peripheral Device*

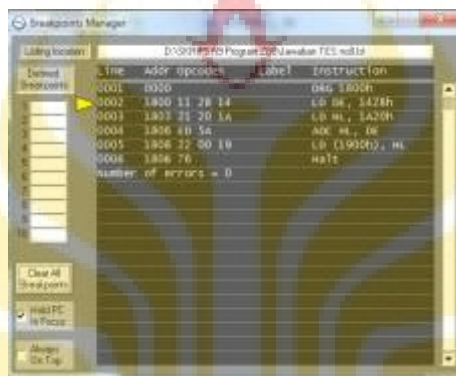
Peripheral Devices, merupakan tools yang digunakan untuk mengendalikan instruksi **IN** dan **OUT** dengan empat (4) buah peralatan I/O eksternal yang dapat disetup serta satu buah terminal yang dapat digunakan untuk melihat karakter ASCII dari data yang dikirimkan ke port tersebut.



Gambar 6. Tampilan jendela *Peripheral Devices*

4) Breakpoint Manager

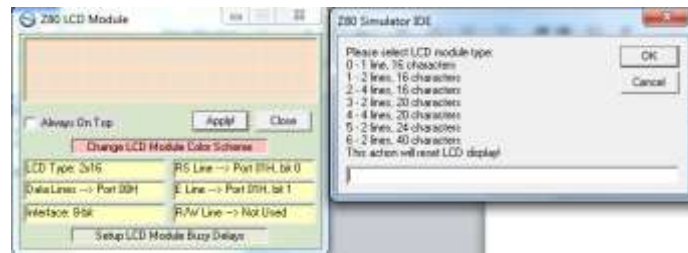
Breakpoints Manager digunakan untuk memulai program *debugger* yang digunakan untuk men-*debug* dan memonitor jalan program. Dengan *tool* ini *programmer* dapat melihat jalannya proses simulasi lebih mudah.



Gambar 7. Tampilan jendela *Breakpoint Manager*

5) Display LCD

Tool ini berfungsi untuk menampilkan hasil pemrograman berupa display karakter yang terintegrasi dengan modul LCD. *Display LCD* ini dapat diubah-ubah menjadi beberapa tipe seperti, 1x16 karakter, 2x16 karakter, 4x16 karakter, 2x20 karakter, 4x20 karakter, 2x24 karakter dan 2x40 karakter. Untuk port data, port E (*Enable*), port RS (*Reset*), port R/W (*Read/Write*) dapat diatur pula sesuai dengan kebutuhan, seperti yang terlihat pada Gambar 8, berikut:



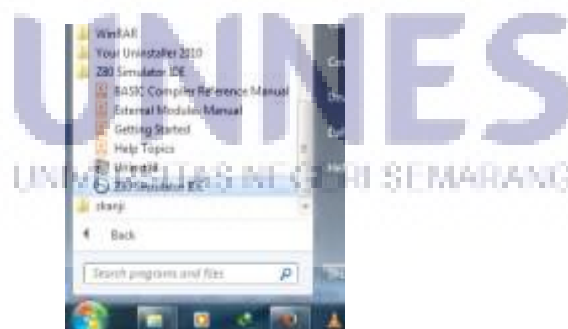
Gambar 8. Tampilan jendela *Display LCD*

d. Langkah-langkah menjalankan Z80-Simulator

Suatu program dapat dijalankan oleh prosesor jika data dari program tersebut telah dimasukkan kedalam memori, sama halnya dengan Z80-Simulator, program dapat disimulasikan jika data telah *diload* atau dimasukkan kedalam memori simulator.

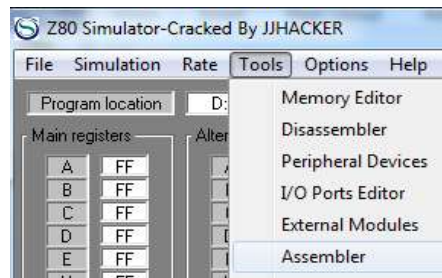
Langkah - langkah untuk menjalankan Z80-Simulator dapat dilakukan dengan cara, sebagai berikut:

- 1) Aktifkan Z80 Simulator dengan cara klik *Start>All Programs>Z80 Simulator>Z80 Simulator IDE*.



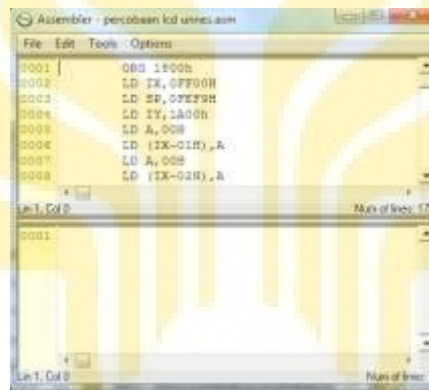
Gambar 9. Tampilan Mengaktifkan Simulator Z80 melalui Start

- 2) Setelah program Z80 Simulator berjalan klik *Menu Tools>Assembler*



Gambar 10. Tampilan mengaktifkan *Tool Assembler* pada jendela utama simulator

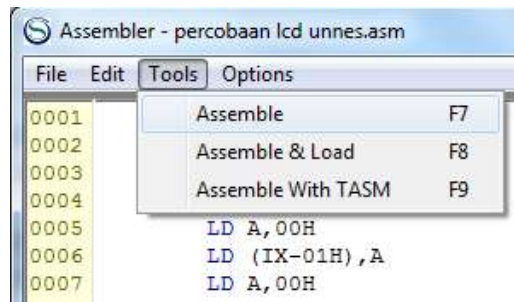
- 3) Setelah Jendela *Assembler* muncul tulis program yang akan dijalankan pada lembar kerja *assembler*.



Gambar 11. Tampilan penulisan program pada lembar kerja *Assembler*

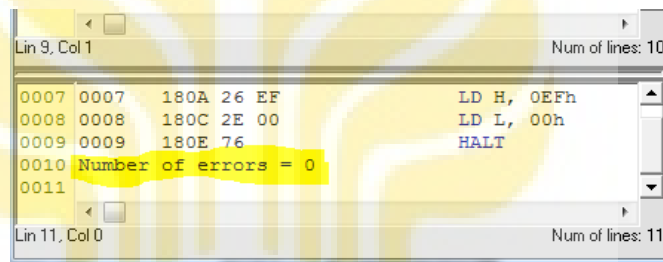
Lembar kerja *assembler* digunakan untuk penulisan program yang akan disimulasikan menggunakan simulator. Penulisan pada lembar kerja *assembler* menggunakan bahasa *assembly*, aturan penulisannya pun sesuai dengan aturan bahasa *assembly*.

- 4) Setelah program ditulis pada lembar kerja *assembler* simpan program dengan cara klik *Files>Save*. Kemudian jalankan fungsi *assemble* dengan cara klik menu *Tools>Assemble*.



Gambar 12. Tampilan me-*assemble* program yang telah ditulis

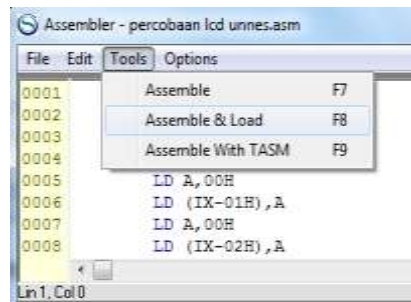
- 5) Setelah program selesai di *assemble* lihat kolom di bawah lembar kerja *assembler* yang akan menampilkan hasil *assemble* dan akan meinformasikan jika terjadi kesalahan atau sukses.



Gambar 13. Tampilan kolom hasil *assemble* program

Kolom dibawah lembar kerja ini akan menampilkan hasil *assemble* yang terdiri dari kode-kode bahasa mesin dari intruksi yang ditulis serta alamat memori dari kode-kode bahasa mesin tersebut. Dengan menampilkan alamat memori penyimpanan kode bahasa mesin ini dapat dimonitor keberadaan di *memory editor*.

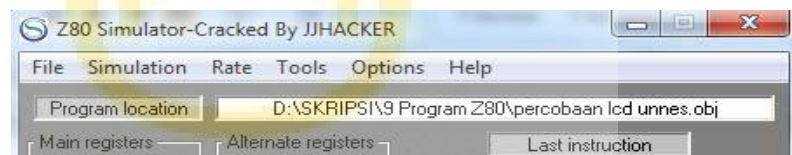
- 6) Setelah program sukses di *assemble* dan tidak ada kesalahan (*number of error = 0*), masukkan data hasil *assemble* ke memori simulasi dengan cara klik *Tools>Assemble & Load*.



Gambar 14. Tampilan proses memasukkan data ke memori simulasi

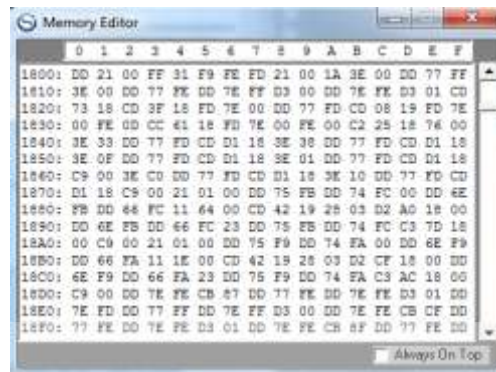
Proses memasukkan data hasil *assemble* ini perlu dilakukan sebelum simulasi dijalankan karena jika data hasil *assemble* tidak dimasukkan atau di *load* ke memori simulasi maka proses simulasi tidak dapat dijalankan.

- 7) Setelah program selesai dimasukkan pada memori simulasi maka pada jendela Z80 Simulator akan menunjukkan alamat file yang dimasukkan.



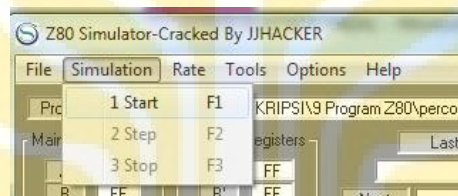
Gambar 15. Tampilan *program location box* jika data berhasil di *load*/dimasukkan pada memori simulasi

Isi data dari program yang telah dimasukkan pada memori simulator dapat dilihat pada *Memory Editor*. Data-data ini nantinya yang akan diproses oleh simulator untuk disimulasikan.



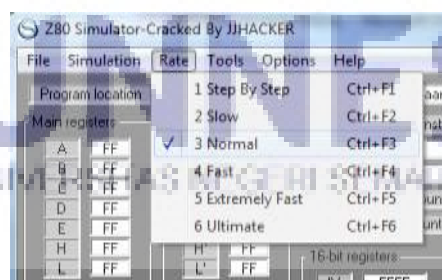
Gambar 16. Tampilan Isi *Memory Editor* setelah program di-load

- 8) Jalankan simulasi dengan cara klik *Menu Simulation>Start* pada jendela Z80 Simulator.



Gambar 17. Tampilan memulai simulasi

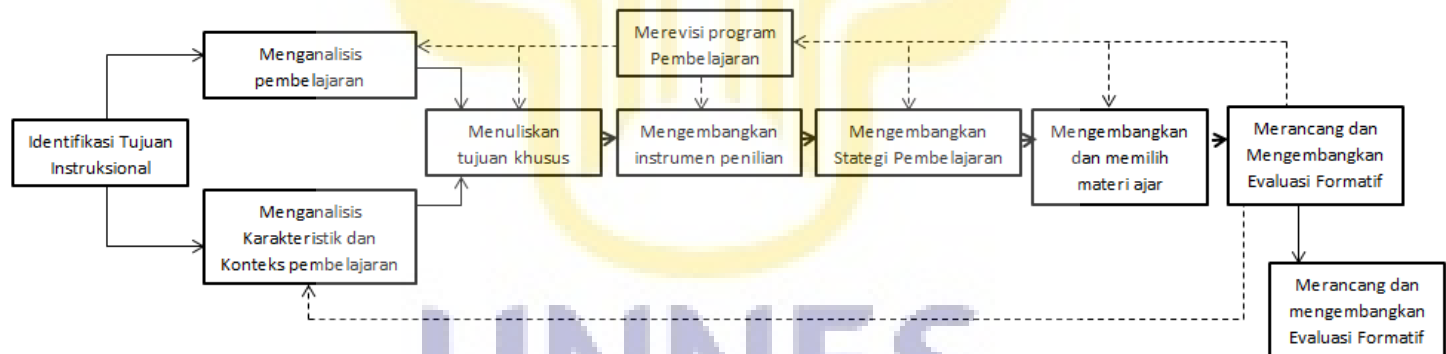
Simulasi akan segera dijalankan, untuk kecepatan proses simulasi dapat diatur melalui *Menu Rate*.



Gambar 18. Tampilan Mengatur kecepatan simulasi

D. PENYUSUNAN MODUL PRAKTIKUM Z80 SIMULATOR SEBAGAI ALAT BANTU PEMBELAJARAN SISTEM MIKROPROSESOR Z80

Pengembangan perangkat pembelajaran seperti modul merupakan serangkaian proses secara sistematis yang dilakukan untuk menghasilkan seperangkat pembelajaran berdasarkan teori pengembangan yang telah ada. Banyak desain pengembangan yang dikaji oleh para ahli. Merujuk pada desain pembelajaran yang digunakan oleh Dick dan Carey pada pengembangan perangkat pembelajaran modul praktikum ini terdapat sepuluh tahapan yang akan dilewati dalam proses perencanaan dan pengembangan pembelajaran, yang terlihat seperti gambar berikut:



Gambar 19. Model Desain Penyusunan Modul Menurut Dicky and Carry

Komponen serta langkah-langkah utama dari model desain penyusunan

modul oleh Dick dan Carey terdiri sebagai berikut:

1. Mengidentifikasi Tujuan Instruksional

Tujuan instruksional adalah rumusan pernyataan mengenai suatu perilaku yang harus dilakukan peserta didik sebagai hasil dari pembelajaran. Tujuan pembelajaran dapat dikembangkan baik dari tujuan pembelajaran yang sudah

ada pada silabus atau dari hasil analisa. Pada pengembangan modul praktikum ini digunakan tujuan instruksional yang dikembangkan dari silabus yang telah disusun.

2. Melakukan analisis pembelajaran

Setelah diperoleh tujuan instruksional yang hendak dicapai kemudian dilakukan analisis instruksional yakni dengan mengklasifikasi tujuan instruksional menjadi ranah kognitif, afektif dan psikomotorik.

3. Menganalisis karakteristik siswa dan konteks pembelajaran

Mengidentifikasi tingkah laku dan karakter peserta didik sangat diperlukan guna untuk memperkirakan kualitas perorangan sebagai petunjuk mempersiapkan strategi pembelajaran.

4. Merumuskan tujuan khusus

Selanjutnya ialah tahap merumuskan tujuan pembelajaran yang spesifik, maksudnya ialah merumuskan kompetensi dasar dan indikator yang akan digunakan dalam proses pembelajaran. KD diambil dari susunan silabus oleh pemerintah sedangkan indikator dirumuskan berdasarkan esensi KD yang diturunkan bersifat operasional dan lebih rinci.

5. Mengembangkan instrumen penilaian

Berdasarkan KI dan KD yang telah dirumuskan, selanjutnya yakni mengembangkan instrumen. Instrumen yang digunakan untuk menganalisis penguasaan siswa terhadap materi yang diberikan berdasarkan kemampuan belajar kognif, akfektif dan psikomotor

6. Mengembangkan strategi pembelajaran

Dalam strategi pembelajaran, untuk mengembangkan dan menjelaskan material pembelajaran haruslah berdasarkan karakteristik siswa. Dalam mengurutkan dan merumpunkan tujuan kedalam pembelajaran dilakukan sesuai dengan tahapan: 1) mengurutkan dan merumpunkan tujuan kedalam pembelajaran; 2) merencanakan pembelajaran, pengetesan, dan tindak lanjut; 3) menyusun alokasi waktu berdasarkan strategi pembelajaran.

7. Mengembangkan dan memilih materi ajar

Materi ajar dikembangkan dari silabus yang telah disusun. Sesuai dengan silabus yang telah dibuat (dapat dilihat pada lampiran 4. Silabus Praktik Sistem Mikroprosesor Z80) penyusunan dan pengembangan materi ajar didalam modul praktikum ini terbagi menjadi: (a) Penjelasan penggunaan Z80 simulator, (b) Proses membaca dan menulis isi data *register* atau memori, (c) konversi kode bahasa mesin dan perintah pada Mikroprosesor Z80, (d) Penulisan program aritmatika dan logika, (e) penulisan program percabangan dan pengulangan, (f) penulisan operasi program *subrutine*, serta (g) penggunaan *peripheral device* dan *port i/o*.

8. Merancang dan mengembangkan evaluasi formatif

Setelah modul selesai disusun, langkah selanjutnya dengan melakukan evaluasi formatif. Terdapat tiga evaluasi formatif yang digunakan dalam penelitian, yakni evaluasi perorangan, evaluasi kelompok kecil dan evaluasi lapangan.

9. **Melakukan revisi terhadap program pembelajaran**

Setelah dilakukan evaluasi formatif maka bahan pengajaran atau modul perlu dilakukan penyempurnaan agar tujuan pembelajaran dapat dicapai. Revisi dilakukan sesuai dengan hasil penilaian dari evaluasi formatif berupa penilaian perorangan, penilaian kelompok kecil serta uji coba lapangan. Revisi yang perlu dipertimbangkan, yaitu terkait akan (a) bahan pembelajaran agar lebih cermat, baik berupa isi atau substansinya maupun format bentukannya, (b) serta dapat pula revisi terhadap cara-cara yang dipakai dalam pembelajaran.

10. **Merancang dan mengembangkan evaluasi sumatif**

Dengan evaluasi sumatif dapat diketahui penilaian dari suatu desain pembelajaran dengan keputusan penilaian berdasarkan keefektifan dan efisiensi kegiatan belajar mengajar. Maka dari itu, keberhasilan tujuan yang ditetapkan diperlihatkan berdasarkan unjuk kerja siswa.

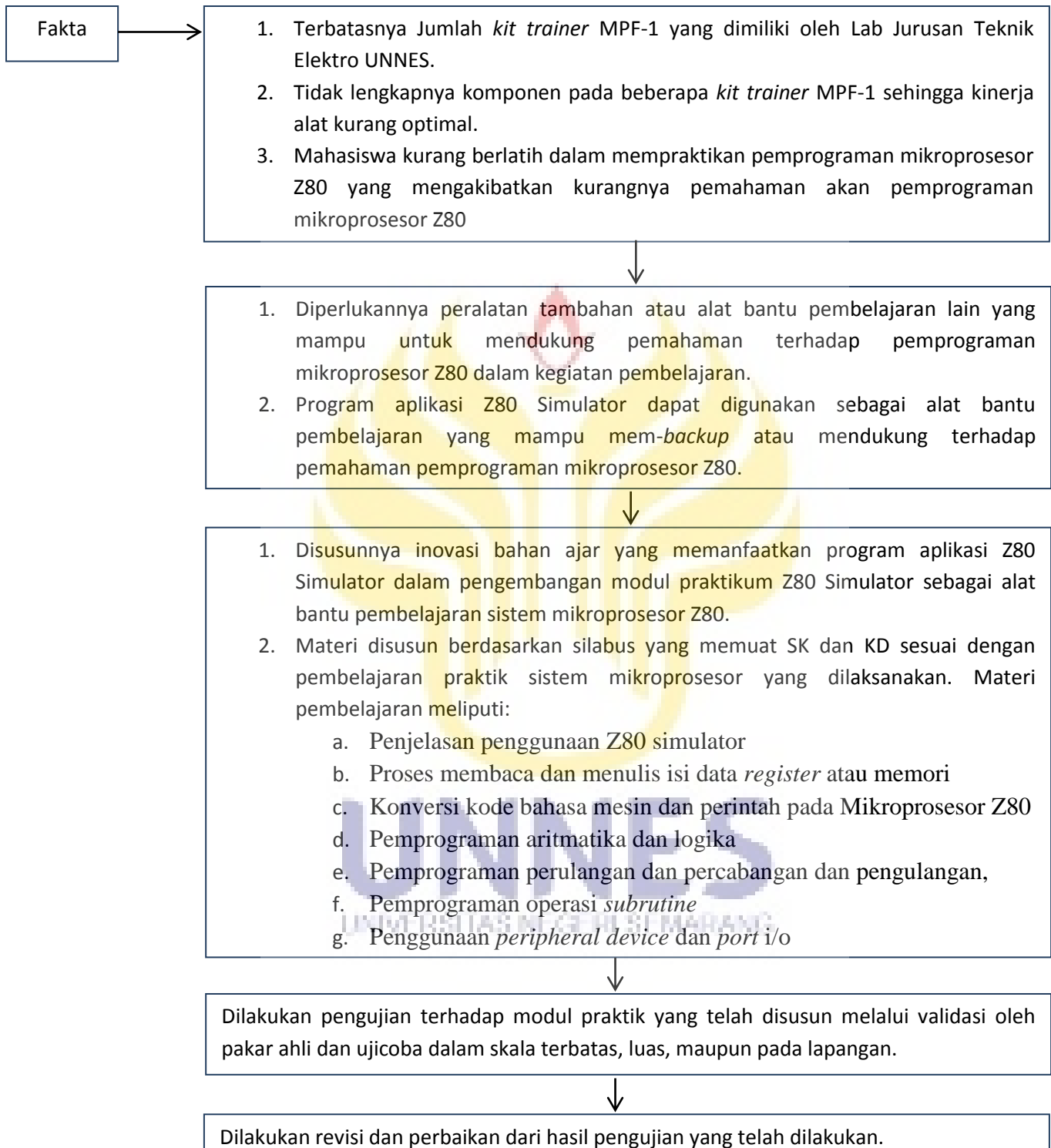
E. KERANGKA BERFIKIR

Mata Kuliah Praktik Sistem Mikroprosesor merupakan mata kuliah yang mempelajari tentang perancangan, pemrograman, serta mengoperasikan mikroprosesor seperti Mikroprosesor Zilog 80. Zilog 80 dipilih untuk dipelajari karena dapat memberikan kemudahan pada mahasiswa serta pemula dalam memahami proses kerja dan pemrograman sebuah mikroprosesor dengan arsitekturnya yang masih dasar dan sederhana.

Untuk mempelajari dan memahami pemrograman mikroprosesor Z80 dapat digunakan *kit trainer* MPF-1. Namun keterbatasan jumlah MPF-1 yang dimiliki di Lab Teknik Elektro UNNES disertai pula dengan tidak lengkapnya peralatan dan beberapa kerusakan yang mengakibatkan tidak optimalnya kinerja MPF-1, membuat mahasiswa terkadang kurang mampu memahami materi yang diberikan secara penuh. Untuk mendukung kegiatan praktik sistem mikroprosesor Z80, dengan terbatasnya *kit trainer*, sebenarnya mahasiswa dapat memanfaatkan sebuah program aplikasi berbasis simulasi kinerja dari mikroprosesor Z80 yang dikenal dengan Z80 Simulator. Melalui Z80 Simulator mahasiswa dapat mempelajari proses kerja dan pemrograman Mikroprosesor Z80 sama halnya ketika menggunakan *kit trainer MPF-1*, sebagai contoh dalam membaca isi data *register* maupun memori. Untuk menggunakan Z80 Simulator ini mahasiswa memerlukan adanya petunjuk kerja dan materi yang mendukung penggunaan program aplikasi ini. Maka melalui sebuah modul praktikum program aplikasi Z80 Simulator dapat digunakan untuk mendukung pembelajaran praktik sistem mikroprosesor Z80 dengan lebih terarah dan terstruktur sesuai dengan tujuan pembelajaran yang akan dicapai.

Berdasarkan permasalahan tersebut maka melalui penelitian ini mencoba memfokuskan tentang menyusun Modul Praktikum Z80 Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80. Penyusunan modul praktikum ini disesuaikan dengan silabus dan RPP dari mata kuliah praktik sistem mikroprosesor. Adapun sesuai dengan silabus yang telah disusun, dapat dilihat pada lampiran 2. Silabus Praktik Sistem Mikroprosesor, materi-materi yang

disajikan di dalam modul praktikum berupa: (a) Penjelasan penggunaan Z80 simulator, (b) Proses membaca dan menulis isi data *register* atau memori, (c) konversi kode bahasa mesin dan perintah pada Mikroprosesor Z80, (d) Pemrograman aritmatika dan logika, (e) pemrograman percabangan dan pengulangan, (f) Pemrograman operasi *subrutine*, serta (g) penggunaan *peripheral device* dan port i/o. Dari materi yang digunakan tersebut, kegiatan-kegiatan praktikum dan tugas disusun sejalan dengan tujuan pembelajaran. Dalam proses penyusunan modul praktikum ini, setelah modul disusun yang didalamnya berisikan materi, praktikum serta latihan dan tugas, selanjutnya akan dilakukan pengujian kepada pakar ahli untuk menilai kelayakannya, pembenahan-pembenahan atau revisi dilakukan hingga diperoleh modul yang layak untuk digunakan di dalam pembelajaran. Selanjutnya diteruskan dengan melakukan pengujicobaan ke dalam pembelajaran di kelas. Dari hasil ujicoba tersebut akan diperoleh penilaian dan tanggapan dari mahasiswa terhadap modul yang telah dipraktikan. Hasil tanggapan mahasiswa ini dapat digunakan untuk memperbaiki kekurangan-kekurangan yang terdapat pada modul sehingga diperoleh hasil akhir berupa modul yang layak dan sesuai dengan pembelajaran Praktik Sistem Mikroprosesor Z80. Dari penjelasan tersebut dapat disusun dalam sebuah bentuk diagram kerangka berfikir yang terlihat pada Gambar 20. Sebagai berikut:



Gambar 20. Diagram Kerangka berfikir penelitian

BAB 5

PENUTUP

A. SIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilaksanakan dapat disimpulkan bahwa:

1. Modul Praktikum Z80 Simulator Sebagai Alat Bantu Pembelajaran Sistem Mikroprosesor Z80 direncanakan berdasarkan silabus Praktik Sistem Mikroprosesor yang menghasilkan tujuh buah percobaan. Modul Praktikum dinyatakan layak/valid oleh pakar ahli struktur penyajian melalui penilaian angket dengan hasil kelayakan teknik penyajian memperoleh nilai 92.5%, kelayakan materi penyajian 89.3%, dan kelayakan penyajian pembelajaran 92.5% yang mengindikasikan bahwa modul praktikum layak untuk digunakan.
2. Modul praktikum hasil perbaikan dapat digunakan dalam kegiatan pembelajaran praktik sistem mikroprosesor untuk membantu pemahaman terhadap pemrograman mikroprosesor khususnya Z80 berdasarkan hasil angket tanggapan yang direspon positif siswa dari 13 item pertanyaan dengan memperoleh nilai rata-rata 80% atau sangat setuju.

B. SARAN

Berdasarkan penelitian dan pembahasan yang dilaksanakan, didapat saran yang dapat diberikan, yaitu:

1. Untuk mengatasi kesulitan penginstalan program dapat dilakukan dengan pembelian lisensi secara sah kepada pihak pengembang OSHONSOFT yang dilakukan secara kolektif oleh pihak jurusan Teknik Elektro, sehingga program aplikasi aman dan mudah digunakan.
2. Soal-soal yang terdapat pada modul hendaknya diuji validitas, reabilitas dan tingkat kesukaran serta daya beda sehingga soal tersebut benar-benar valid digunakan.



DAFTAR PUSTAKA

- Arikunto, Suharsimi. (2010). *Prosedur Penelitian Suatu Pendekatan Praktik*. Jakarta : Rineka Cipta.
- Arsyad, Azhar. (2007). *Media Pembelajaran*. Jakarta : PT. Raja Grafindo Persada
- BSNP (Badan Standar Nasional Pendidikan).(2006). *Instrumen Penilaian Tahap II Buku Teks Pelajaran Pendidikan Dasar dan Menengah*. Jakarta : BSNP
- Darmiatur, Suryatri. (2013). *Menyusun Modul: Bahan Ajar Untuk Persiapan Guru Dalam Mengajar*. Yogyakarta : Gava Media.
- Dewi, Ratna. (2007). Skripsi. *Desain dan Produksi Multimedia Pembelajaran Interaktif (MPI) Fisika Sekolah Menengah Atas (SMA) Kelas X dengan Pemrograman Macromedia Flash Profesional 8*. Semarang
- Ihsan, Fuad. (2011). *Dasar-Dasar Kependidikan*. Jakarta: Rineka Cipta.
- Majid, Abdul. (2009). *Perencanaan Pembelajaran: Mengembangkan Standar Kompetensi Guru*. Bandung : PT. Remaja Rosdakarya.
- Maulvino, Albert Paul & Tjia May On. (1994). *Elektronika Komputer Digital : Pengantar Mikrokomputer Edisi kedua*. Jakarta : Erlangga.
- Ord, Jon. (2012). John Dewey and Experiential Learning : Developing The Teori of Youth Work. *Jurnal of Youth & Policy* 108 (4): 55-72.
- Oshonsoft. (2001). *Z80 Simulator IDE: With Z80 Basic Compiler* [online]. Tersedia di : www.Oshonsoft.com/z80.html. 20 Juli 2016 (15:00).
- Sentot Wijanarka, Bernardus. (2012). Disertasi. *Pengembangan Modul dan Pembelajaran Kompetensi Kejuruan Teknik Permesinan CNC SMK*. Yogyakarta.
- Sugiyono. (2010). *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung : Alfabeta.
- Somantri, Yoyok. (tt). *Arsitektur Mikroprosesor Z80. Modul Belajar MPTK Universitas Pendidikan Indonesia*.
- UU RI No.20 Tahun 2003 *Sistem Pendidikan Nasional* pasal 1 ayat 20