

# BAB I

## PENDAHULUAN

### A. Deskripsi

Modul ini berisi materi untuk melakukan latihan kompetensi kemampuan pengembangan Algoritma Pemrograman menggunakan Bahasa Assembly pada sistim mikroprosesor bagi peserta diklat SMK program keahlian Elektronika Industri dan Teknik Audio Video.

Melalui modul ini anda dapat berlatih mengembangkan kompetensi mendefisikan masalah pemrograman, mengembangkan algoritma pemrograman dan flowchart pemrograman menggunakan bahasa pemrograman *Assembly*.

### B. Prasyarat

Untuk mempelajari dan melakukan latihan kompetensi dalam modul ini ada dua syarat yaitu :

#### Syarat Umum :

Anda harus belajar dan berlatih kompetensi dengan rumus “TePUK DisKo” yaitu Teratur, Percaya diri, Ulet, Kreatif, Disiplin dan Konsentrasi.

#### Syarat Khusus :

Anda harus sudah mempelajari Sistim digital, Logika, dan memahami matematika Sistim Bilangan Biner, Heksa Desimal

### C. Petunjuk Penggunaan Modul

#### 1. Bagi Guru/Fasilitator

- ✓ Baca dan cermati betul deskripsi Silabi
- ✓ Pelajari Kompetensi Dasar, Indikator, Materi Pembelajaran, Kegiatan Pembelajaran, dan Penilaian
- ✓ Pelajari Level Kompetensi Kunci dan Jabarannya
- ✓ Lihat Tujuan Akhir Pembelajaran apakah sudah sesuai dengan Indikator Silabus sebagai tuntutan Kriteria Kinerja deskripsi kompetensi.
- ✓ Cocokkan cakupan kegiatan belajar dengan deskripsi Materi Pembelajaran dan Kegiatan Pembelajaran. Cermati apakah materi kegiatan pembelajaran telah mencakup keseluruhan Kompetensi Dasar dalam aspek Sikap, Pengetahuan, dan Keterampilan.
- ✓ Fasilitasi peserta diklat untuk berlatih kompetensi mendefisikan masalah pemrograman, mengembangkan algoritma pemrograman dan flowchart pemrograman menggunakan

bahasa pemrograman Assembly sesuai prinsip pembelajaran KBK yaitu berbasis pada siswa, belajar secara terintegrasi, *Individual learning*, *Mastery learning*, *Problem Solving*, *Experience Based Learning*.

## 2. Bagi Peserta Diklat

- ✓ Baca dan pahami deskripsi modul dan prasyarat penggunaan modul
- ✓ Baca dan pahami tujuan akhir modul
- ✓ Isikan rencana kegiatan belajar dengan berkonsultasi dengan guru/fasilitator
- ✓ Baca dan laksanakan cek kemampuan dengan berkonsultasi dengan guru/ fasilitator

## D. Tujuan Akhir

Menguasai prosedur Pendefinisian masalah pemrograman, penyusunan Algoritma Pemrograman, Flowchart pemrograman, dan kaidah pemrograman dengan bahasa *assembly*.

## E. KOMPETENSI

BIDANG KEAHLIAN : Teknik Elektronika  
 PROGRAM KEAHLIAN : Teknik Elektronika Industri  
 STANDAR KOMPETENSI : Memprogram Peralatan Sistem Otomasi Elektronik yang Berkaitan I/O berbantuan : Mikroprosesor dan Mikrokontroler.  
 KODE : ELIND1  
 JAM PEMBELAJARAN : 45 (90) Jam @ 45 menit

LEVEL KOMPETENSI	A	B	C	D	E	F	G
KUNCI	3	3	3	3	2	3	3

KONDISI KINERJA	Unjuk kerja ketrampilan kognitif namun dengan imajinasi psiko-motorik seperti unit kompetensi ini bisa dicapai dengan kondisi: 1. Memiliki kemampuan dasar tentang konsep sistem 2. Memiliki kompetensi dasar elektronika 3. Memiliki kemampuan mengenai petunjuk keselamatan kerja secara umum 4. Memiliki kemampuan menulis laporan kerja yang baik.
-----------------	--

KOMPETENSI DASAR	INDIKATOR	MATERI PEMBELAJARAN	KEGIATAN PEMBELAJARAN	PENILAIAN	ALOKASI WAKTU			SUMBER BELAJAR
					TM	PS	PI	
I. Menguasai Prosedur Penyusunan Algoritma Pemrograman	I.1. Ditunjukkan kaidah-kaidah yang berlaku dalam penyusunan algoritma dan pemrograman.	Kaidah-kaidah penyusunan algoritma	<ul style="list-style-type: none"> <li>▪ Memahami pengertian Program Komputer</li> <li>▪ Menguraikan empat langkah Pengembangan Program Komputer</li> <li>▪ Berlatih mendefinisikan Permasalahan Program</li> <li>▪ Berlatih mengembangkan Algoritma Program</li> <li>▪ Berlatih memilih bentuk baku Flowchart</li> <li>▪ Berlatih menyusun Flowchart</li> </ul>	<ul style="list-style-type: none"> <li>• Tes Lesan</li> <li>• Tes Tulis</li> </ul>	10	10 (20)		<i>Microprocessor and Interfacing Programming and Hardware</i> MCGraw-Hill, 1992

KOMPETENSI DASAR	INDIKATOR	MATERI PEMBELAJARAN	KEGIATAN PEMBELAJARAN	PENILAIAN	ALOKASI WAKTU			SUMBER BELAJAR
					TM	PS	PI	
	1.2. Diuji algoritma dan pemrograman yang telah dikuasai dengan melakukan beberapa kasus.	Pengujian algoritma dan pemrograman	<ul style="list-style-type: none"> <li>▪ Menguji algoritma dan pemrograman</li> <li>▪ Menggunakan algoritma dalam pemrograman</li> <li>▪ Menguji studi kasus penggunaan algoritma dalam pemrograman</li> <li>▪ Memahami Bahasa Pemrograman Komputer</li> </ul>	<ul style="list-style-type: none"> <li>• Tes Lisan</li> <li>• Tes Tulis</li> </ul>	10	5 (10)	10 (40)	<i>Microprocessor and Interfacing Programming and Hardware</i> MCGraw-Hill, 1992

#### LEVEL KOMPETENSI KUNCI

No	Kompetensi Kunci	LEVEL 1	LEVEL 2	LEVEL 3
A	MENGUMPULKAN, MENGANALISA DAN MENGELOLA INFORMASI Kapasitas untuk mengumpulkan informasi, memindahkan dan menyeleksi informasi dalam rangka memilih informasi yang diperlukan untuk dipresentasikan, mengevaluasi sumber dan cara memperoleh informasi tersebut	Mengakses dan menyimpan dari satu sumber	Mengakses, memilih dan menyimpan dari beberapa sumber	<i>Mengakses, mengevaluasi dan mengatur dari berbagai macam sumber</i>
B	MENKOMUNIKASIKAN IDE-IDE DAN INFORMASI Kapasitas untuk berkomunikasi dengan orang lain secara efektif menggunakan beragam bahasa, tulisan, grafik dan ekspresi non verbal lainnya	Sederhana dengan aturan yang telah dikenal	Komplek dengan isi tertentu	<i>Komplek dengan isi beragam</i>
C	MERENCANAKAN DAN MENGORGANISIR KEGIATAN Kapasitas untuk merencanakan dan mengatur kegiatan kerja individu termasuk penggunaan waktu dan sumber yang baik, pemilihan prioritas dan pengawasan prestasi individu	Di bawah pengawasan	Dengan bimbingan	<i>Inisiatif sendiri dan mengevaluasi kegiatan yang komplek</i>

No	Kompetensi Kunci	LEVEL 1	LEVEL 2	LEVEL 3
D	BEKERJA DENGAN ORANG LAIN SERTA KELOMPOK DALAM SATU TIM Kapasitas untuk berhubungan secara efektif dengan orang lain baik antar pribadi ataupun kelompok termasuk mengerti dan memberikan respon akan keinginan klien dan bekerja secara efektif sebagai anggota kelompok untuk mencapai tujuan bersama.	Aktifitas yang telah diketahui	Membantu merancang dan mencapai tujuan	<i>Kolaborasi dalam kegiatan kelompok</i>
E	MENGGUNAKAN IDE-IDE SERTA TEKNIK DALAM MATEMATIKA Kapasitas untuk menggunakan konsep bilangan, spasi dan ukuran dan teknik seperti perkiraan untuk praktek	Tugas yang sederhana	<i>Memilih tugas yang kompleks dan sesuai</i>	Evaluasi dan mengadaptasi sebagai tugas yang sesuai
F	MENYELESAIKAN MASALAH Kapasitas untuk menjalankan strategi penyelesaian masalah baik untuk situasi di mana masalah dan solusi yang diinginkan memiliki bukti dan dalam situasi yang memerlukan pemikiran kritis dan pendekatan kreatif untuk mencapai hasil	Rutin, sedikit pengawasan eksplorasi – pengawasan melekat	Rutin, mandiri eksplorasi- dengan bimbingan	<i>Masalah yang kompleks, pelaksanaan pendekatan sistematis, menjelaskan proses</i>
G	MENGGUNAKAN TEKNOLOGI Kapasitas untuk menerangkan teknologi, mengkombinasikan keahlian fisik dan sensor yang diperlukan untuk menjalankan peralatan dengan pengertian ilmiah dan prinsip teknologi yang diperlukan untuk mengadaptasi sistem	Produksi ulang atau melaksanakan produk dasar atau jasa	Menyusun, mengatur atau mengoperasikan produk atau jasa	<i>Mendesain atau merangkai produk atau jasa</i>

## F. Cek Kemampuan

Berilah tanda thick (✓) pada kolom YA atau TIDAK sesuai dengan pernyataan berikut. Mintalah Catatan dan Tanda Tangan Guru/Pembimbing.

NO	PERNYATAAN	YA	TIDAK	CATATAN GURU/PEMBIMBING	TANDA TANGAN GURU/PEMBIMBING
01	Apakah Anda sudah memahami cara mendefinisikan masalah Pemrograman				
02	Apakah Anda sudah memahami Prosedur pengembangan algoritma dan flowchart pemrograman				
03	Apakah Anda sudah menggunakan bahasa pemrograman Assembly ?				

## BAB II PEMBELAJARAN

### A. Rencana Belajar Peserta Diklat

Rencana belajar peserta diklat diisi oleh peserta diklat dan disetujui oleh Guru. Rencana belajar tersebut adalah sebagai berikut :

NAMA PESERTA DIKLAT : \_\_\_\_\_

JENIS KEGIATAN	TANGGAL	WAKTU	TEMPAT BELAJAR	ALASAN PERUBAHAN	TANDA TANGAN GURU

# Kegiatan Belajar

## 1. Kegiatan Belajar I : Konsep Dasar Algoritma dan Flowchart

### a. Tujuan Kegiatan Belajar I

- ☞ Peserta diklat dapat menunjukkan kaidah-kaidah penyusunan algoritma pemrograman.
- ☞ Peserta diklat menggunakan prosedur penyusunan algoritma sesuai konsep dasar pengembangan program.
- ☞ Peserta diklat dapat memilih bentuk grafis Flowchart sesuai standar yang berlaku.
- ☞ Peserta diklat mendemonstrasikan prosedur penyusunan Flowchart menggunakan bentuk-bentuk grafis Flowchart standar berdasarkan algoritma program

### b. Uraian Materi



ernahkah anda berpikir bahwa komputer adalah sebuah peralatan yang sangat kompleks, sulit dipelajari, dan dapat berfikir melebihi manusia ?.

Jika ya maka dapat dikatakan pendapat ini terlalu berlebihan. Komputer hanya dapat bekerja jika ada perintah-perintah yang diberikan yang disebut **program**. **Program komputer** buatan manusia. Manusia pembuat program komputer disebut programmer. Program adalah susunan atau urutan perintah-perintah sederhana yang diberikan kepada komputer untuk memecahkan satu atau beberapa permasalahan.

Tanpa program komputer tidak dapat difungsikan. Perangkat keras komputer tidak ada gunanya tanpa program. Sebaliknya program juga tak berarti apa-apa tanpa perangkat keras.

Anda pasti sudah tahu program komputer banyak sekali jenisnya. Ada program untuk permainan/game yang pasti anda sukai. Di Amerika Serikat program game digunakan untuk anak Play group dan anak TK, agar tertarik dan muncul minatnya untuk belajar komputer. Setelah suka dan tertarik dengan komputer baru diajari program-program aplikasi.

Bagaimana dengan anda, apakah anda baru tertarik dengan game ?. Jika ya belajarlh lebih giat tentang program dan pemrograman. Dalam modul ini anda akan diajak belajar membuat program. Bersiap-siaplah untuk belajar dan berlatih membuat program. Sebelum menulis program ikuti terlebih dahulu langkah-langkah pembuatan program.

Perintah-perintah yang digunakan dalam pembuatan program disebut *Instruction Set*. Kebanyakan mikroprosesor memiliki kesamaan dalam perintah atau instruksi. Instruksi transfer data dapat

menggunakan perintah LOAD atau MOVE, Instruksi aritmetika sederhana menggunakan perintah ADD, SUB, MUL, DEV. Instruksi lengkap dibahas dalam pendalaman materi *Instruction Set*.

## Langkah-Langkah Pengembangan Program

Menurut Douglas ada empat langkah yang harus dilakukan dalam mengembangkan program komputer yaitu :

1. Pendefinisian permasalahan,
2. Representasi kerja program,
3. Penemuan instruksi-instruksi yang benar, dan
4. Penulisan program

### ❖ PENDEFINISIAN PERMASALAHAN

Langkah pertama yang harus dilakukan dalam menulis program adalah memikirkan secara cermat permasalahan yang ingin diselesaikan menggunakan program komputer. Dengan kata lain apa yang ingin dikerjakan oleh sebuah program. Jika anda telah melakukan identifikasi permasalahan dan mendefinisikan permasalahan dengan jelas dan benar maka ini merupakan langkah awal yang sangat baik dalam menulis apa yang diinginkan dalam pembuatan program. Mari kita lihat satu contoh permasalahan:

"Menyeberang di jalan yang sangat ramai "

### ❖ REPRESENTASI KERJA PROGRAM

#### Algoritma

Untuk memecahkan permasalahan bagaimana menyeberang di jalan yang sangat ramai diperlukan langkah-langkah atau sekuen atau formula kerja. Formula kerja yang digunakan untuk memecahkan masalah pemrograman disebut *Algoritma program*. Seorang programmer harus menggunakan daftar urutan pekerjaan. Dalam kasus permasalahan menyeberang jalan step perintah-perintah sederhana dapat dinyatakan seperti Gambar 1 berikut :

STEP	PERINTAH
1.	Start
2.	Berjalanlah ke sudut jalan dan berhenti
3.	Lihat dan cermati lampu pengatur lalu lintas
4.	Apakah lampu penyeberangan pada arah anda menyala merah
5.	Jika “ya”, kembali ke Step 3 (Untuk keadaan lain teruskan ke Step 6)
6.	Lihat ke arah kiri
7.	Apakah masih ada kendaraan yang lewat
8.	Jika “ya”, kembali ke Step 6
9.	Lihat ke arah kanan
10.	Apakah masih ada kendaraan yang lewat
11.	Jika “ya”, kembali ke Step 9
12.	Menyeberanglah dengan hati-hati
13.	Stop

Gambar 1. Algoritma Menyeberang Jalan

Setelah anda membaca urutan step-step perintah pada Gambar 1 apa komentar anda ?.

Perintah dalam program mengalir dari step awal (step 1) ke step 2, step 3 dan seterusnya. Kecuali ada perintah untuk melompat atau memanggil subrutin atau jika ada interupsi.

Pada step 6 dan step 7 merupakan step pengambilan keputusan atas perintah step 5. Pada step 7 secara tidak tertulis “jika tidak” proses kontinyu atau meneruskan ke step 8.

Jika kesebelas langkah ini dilakukan maka anda akan selamat dan sukses menyeberang jalan. Masalahnya maukah anda berpikir dan bertindak secara disiplin seperti algoritma di atas. Merubah cara berpikir dari level tinggi menjadi cara berpikir level rendah seperti bahasa bayi. Jika Ya maka niscaya anda akan bisa memprogram mikroprosesor dengan baik. Jika tidak disinilah hambatan saudara belajar memprogram.

Kesebelas langkah ini adalah bahasa bayi atau bahasa aras rendah, yang pada kenyataannya dilakukan pada setiap menyeberang jalan yang sibuk dan ada lampu mengatur lalu lintas. Kesebelas sekuen step perintah ini disebut juga dengan **Algoritma Program**.

Semakin detail algoritma suatu program semakin baik karena program menghendaki kesuksesan 100% sesuai tuntutan permasalahan. Bisa saja step 5, step 6 dan step 7 tidak perlu ada manakala jalur lalu lintas hanya satu arah. Sangat berbahaya jika step 2, step 3, dan step 4 diabaikan.

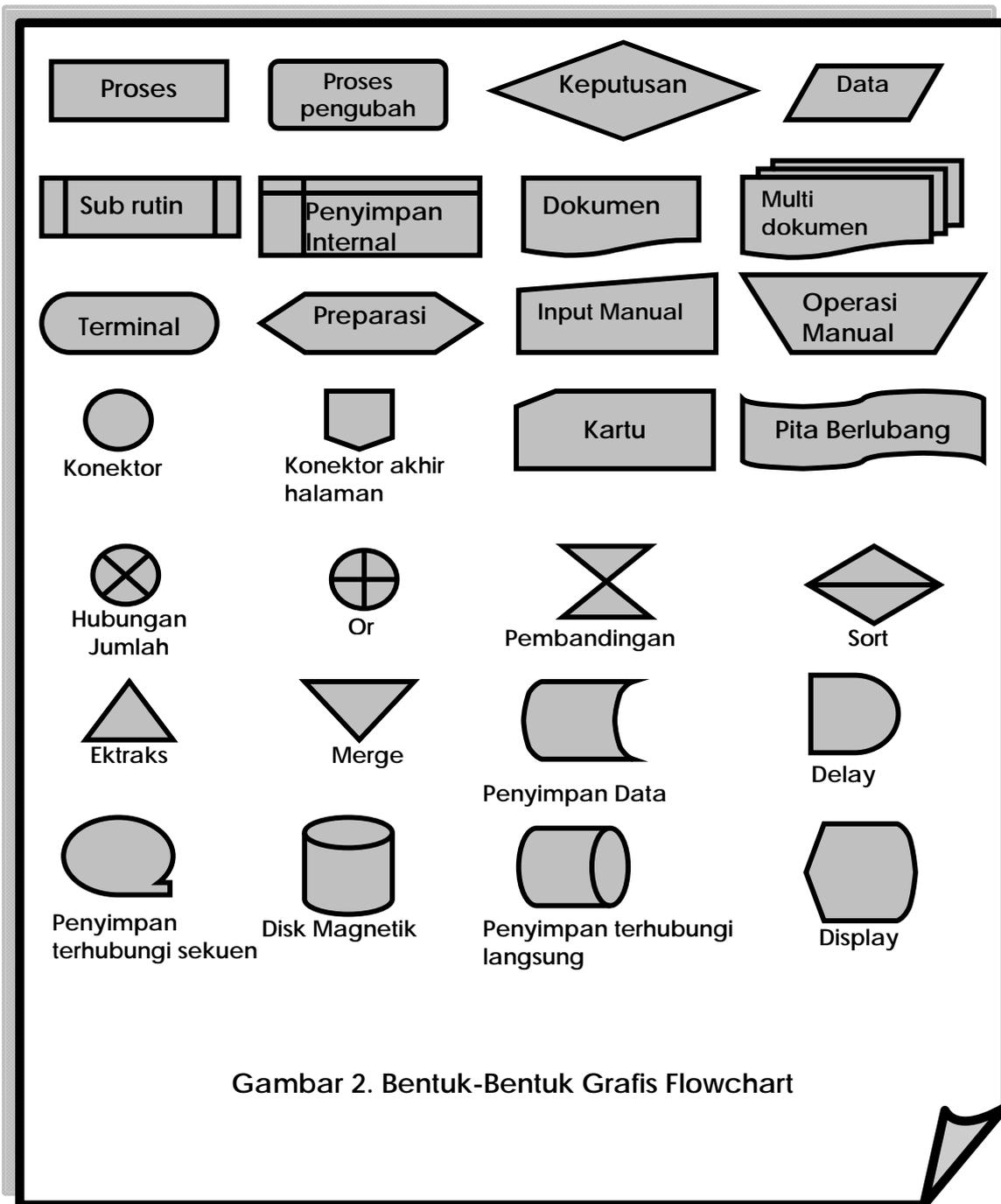
Kasus menyeberang di jalan yang sangat ramai dengan 11 step penyelesaian merupakan bentuk bahasa aras rendah. Mikroprosesor di dalam sistim bekerja dengan bahasa aras rendah. Agar anda sukses dan mudah belajar memprogram mikroprosesor maka anda harus memiliki kesiapan merubah kebiasaan berfikir pada aras tinggi kekebiasaan berfikir di aras rendah. Inilah kendala utama dalam belajar memprogram mikroprosesor.

## Flowchart

Flowchart atau diagram alir digunakan untuk menunjukkan aliran proses sebuah program. Untuk menyajikan jenis operasi sebuah program digunakan bentuk-bentuk grafis standar. Ada dua puluh delapan jenis bentuk grafis yang digunakan untuk menyusun Flowchart seperti pada Gambar 2. Bentuk-bentuk grafis penyusun Flowchart dapat dilihat pada AutoShapes Flowchart program Microsoft Word. Pemilihan bentuk-bentuk grafis Flowchart pada Gambar 2 tidak boleh sembarangan atau asal pilih apalagi membuat sendiri bentuk-bentuk lain secara bebas. Hal ini tidak dibenarkan karena semua bentuk grafis Flowchart telah disepakati dan distandarkan secara internasional sebagai alat komunikasi.

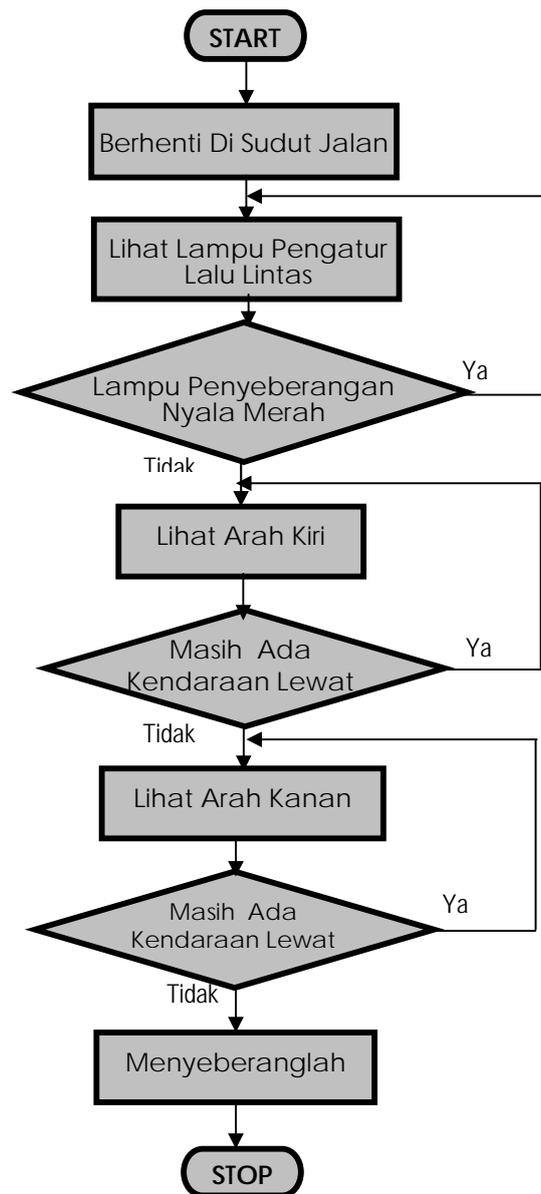
Cermati satu per satu bentuk-bentuk grafis Flowchart Gambar 2. Pahami bentuk dan fungsinya masing-masing. Jika ada yang tidak jelas atau kurang bisa dipahami bertanyalah kepada fasilitator atau guru anda.

Berlatih dan berlatih diikuti dengan mengkonsultasikan kebenaran hasil latihan anda kepada fasilitator atau guru akan sangat menentukan terbentuknya kompetensi pada diri anda.



Gambar 2. Bentuk-Bentuk Grafis Flowchart

Dua puluh delapan bentuk-bentuk grafis Flowchart pada Gambar 2 dapat dipilih salah satu sesuai jenis step perintah pada algoritma. Dari sebelas step algoritma program pada Gambar 1 dapat disusun Flowchart kasus menyeberang di jalan yang sangat ramai seperti Gambar 3 berikut :



Gambar 3. Flowchart Menyeberang di Jalan Ramai

Flowchart menyeberang di jalan ramai pada Gambar 3 diawali dengan perintah START sebagai tanda mulai. Perintah start sebagai tanda mulai menggunakan bentuk grafis terminal. Jika masih ragu dengan bentuk grafis terminal lihat kembali bentuk-bentuk grafis flowchart gambar 2.

Selanjutnya diteruskan dengan dua proses "berhenti di sudut jalan" dan "melihat lampu pengatur lalu lintas". Kedua proses tersebut menggunakan bentuk grafis segi empat. Pengambilan keputusan dengan pertanyaan apakah "Lampu Penyeberangan Nyala

Merah” menggunakan bentuk grafis belah ketupat. Proses pengambilan keputusan dengan pertanyaan apakah ”Lampu Penyeberangan Nyala Merah” bermakna jika ”ya” kembali atau tetap melihat lampu pengatur lalu lintas. Sebaliknya jika ”tidak” menyala merah berarti menyala hijau mengandung pengertian bisa meneruskan ke step berikutnya yaitu proses melihat arah kiri.

Pertanyaan apakah masih ada kendaraan yang lewat merupakan step pengambilan keputusan. Jika ya harus tetap melihat arah kiri dan jika tidak ada kendaraan yang lewat bisa meneruskan ke step lihat arah kanan. Pertanyaan apakah masih ada kendaraan yang lewat dari arah kanan merupakan step pengambilan keputusan. Sampai pada proses menyeberang jika tidak lagi ada kendaraan yang lewat. Flowchart diakhiri dengan terminal STOP. Step melihat arah kiri dan melihat arah akan dalam kasus ini bisa saja ditukarkan posisinya.

Dua langkah selanjutnya dalam proses pembuatan program yaitu penemuan jenis instruksi yang benar dan penulisan program akan dibahas pada kegiatan belajar modul berikutnya.

### c. **Rangkuman**

Mengembangkan program komputer agar efektif harus mengikuti empat langkah sesuai saran atau cara Douglas. Sebelum sampai pada penulisan program pertama definisikan permasalahan program secara jelas lalu representasikan kerja program dengan algoritma dan flowchart. Yakinkan algoritma dan flowchart anda benar dan sesuai tuntutan permasalahan program. Kemudian baru menulis program dengan memanfaatkan instruksi-instruksi yang benar dan tepat.

d. **Tugas/ Lembar Kerja**

Untuk membangun kompetensi penguasaan prosedur penyusunan algoritma pemrograman anda harus mengerjakan tugas-tugas lembar kerja berikut. Lakukan dengan sungguh-sungguh mulai dari kasus yang sederhana sampai nanti pada kasus yang kompleks.

**Tugas 1.**

Buatlah Flowchart dari permasalahan mengisi gelas dengan air dari keran jika algoritma programnya seperti Gambar 4.

STEP	PERINTAH
1.	Start
2.	Letakkan Gelas di bawah keran
3.	Buka keran
4.	Lihat Gelas
5.	Apakah Gelas belum terisi penuh
6.	Jika ya kembali ke STEP 4
7.	Tutup keran
8.	Pindahkan Gelas dari bawah keran
9.	Stop

Gambar 4. Algoritma Mengisi Gelas dengan Air dari Keran

Gambar 5. Flowchart Mengisi Gelas dengan Air dari Keran

## Tugas 2.

Buatlah Flowchart dari permasalahan memindahkan 100 buah Resistor dari Kotak A ke Kotak B, jika algoritma programnya seperti Gambar 6.

STEP	PERINTAH
1.	Start
2.	Buat cacahan jumlah Resistor $N = 100$
3.	Pindahkan Resistor dari Kotak A ke Kotak B
4.	Kurangi nilai $N = N - 1$
5.	Apakah Nilai $N$ tidak sama dengan 0
6.	Jika ya kembali ke STEP 3
7.	Stop

Gambar 6. Algoritma Memindahlkan 100 buah Resistor dari Kotak A ke Kotak B

Gambar 7. Flowchart Memindahlkan 100 buah Resistor dari Kotak A ke Kotak B

### Tugas 3.

Buatlah Flowchart dari permasalahan membangkitkan 100 data bilangan desimal mulai dari 0 pada memori mulai alamat 1900, jika algoritma programnya seperti Gambar 5.

STEP	PERINTAH
1.	Start
2.	Buat cacahan jumlah bilangan $N = 100$
3.	Set data awal $A = 0$
4.	Set alamat memori $M = 1900$
5.	Simpan data A di memori alamat M
6.	Tambahkan nilai alamat memori $M = M + 1$
7.	Tambahkan nilai data $A = A + 1$
8.	Kurangkan nilai cacahan $N = N - 1$
9.	Apakah nilai N tidak sama dengan 0
10.	Jika Ya kembali ke STEP 5
11.	Selesai

Gambar 8. Algoritma Membangkitkan 100 data bilangan desimal di memori alamat 1900

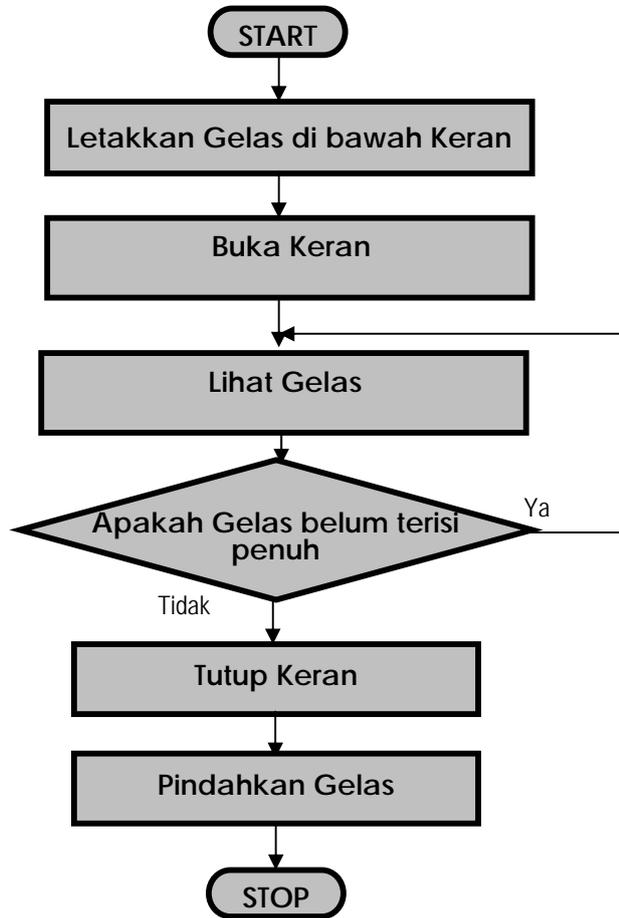
Gambar 9. Flowchart Membangkitkan 100 data bilangan desimal di memori alamat 1900

**e. Tes Formatif**

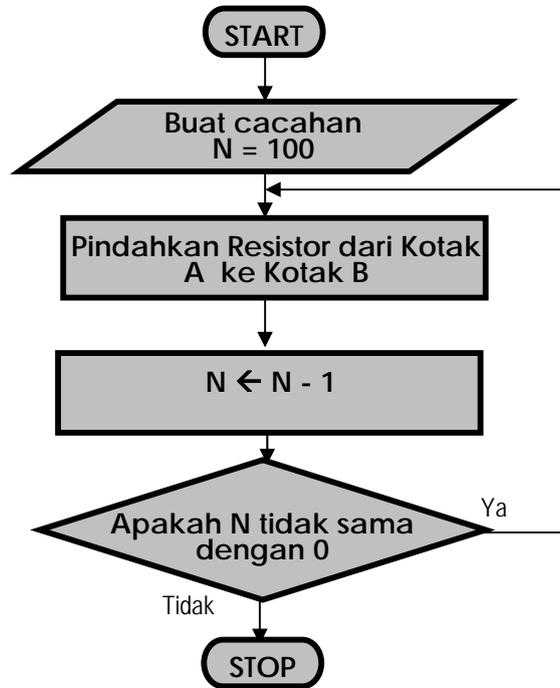
1. Komputer tidak bisa berfungsi jika tidak ada program. Menurut anda pernyataan ini benar atau salah. Jelaskan alasannya.
2. Permasalah program harus didefinikan terlebih dahulu sebelum menyusun algoritma. Menurut anda pernyataan ini benar atau salah. Jelaskan alasannya.
3. Jelaskan prosedur atau langkah-langkah penyusunan flowchart.

f. Kunci Jawaban

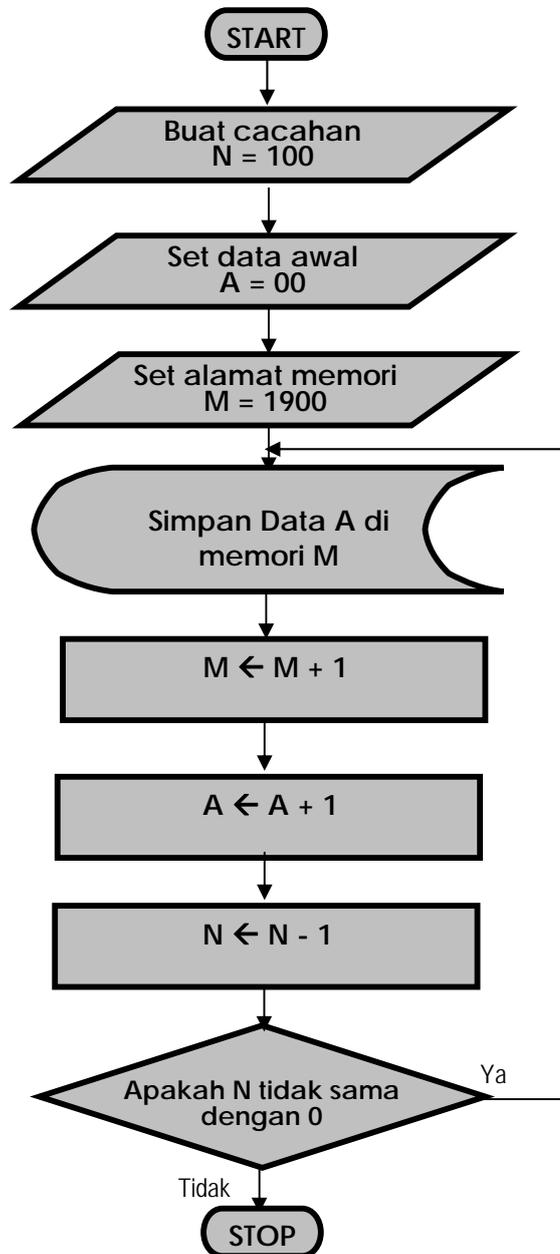
Tugas 1: Flowchart Mengisi Gelas dengan Air dari Keran



Tugas 2: Flowchart Memindahlkan 100 buah Resistor dari Kotak A ke Kotak B



Tugas 3: Flowchart Pembangkitan 100 data bilangan desimal mulai 0



### Kunci Jawaban Tes Formatif

1. Pernyataan bahwa komputer tidak bisa berfungsi jika tidak ada program adalah benar. Karena Komputer merupakan sistim yang bekerja berdasarkan program. Tanpa program komputer tidak bisa bekerja.
2. Pernyataan permasalahan program harus didefinikan terlebih dahulu sebelum menyusun algoritma adalah benar. Tanpa pendefisian masalah yang jelas tidak mungkin menyusun algoritma. Karena penyusunan algoritma menentukan langkah-langkah atau sekuen atau formula kerja. Formula kerja yang digunakan untuk memecahkan masalah pemrograman.
3. Memahami perintah setiap step pada algoritma. Memilih bentuk grafis flowchart yang sesuai dengan jenis perintah pada algoritma. Menyusun perintah-perintah sebagai aliran proses kerja program

## 2. Kegiatan Belajar II : Bahasa Pemrograman

### a. Tujuan Kegiatan Belajar II

- Peserta diklat memahami konstruksi program assembly
- Peserta diklat memahami proses kerja assembly
- Peserta diklat dapat menggunakan berbagai jenis mnemonik
- Peserta diklat dapat menggunakan assembler directive

### b. Uraian Materi



Untuk menjalankan fungsi dan kerja, sebuah mikrokomputer memerlukan program. Dalam perangkat keras Sistem Mikroprosesor program tersimpan dalam format biner pada suatu lokasi-lokasi unit memori. Format biner program komputer disebut sandi operasi yang sering juga diberi sebutan bahasa mesin. Ada tiga tingkat level bahasa yang dapat digunakan untuk menulis program pada sebuah mikrokomputer yaitu:

- Bahasa MESIN
- Bahasa ASSEMBLY
- Bahasa Aras Tinggi (High Level)

#### ❖ BAHASA MESIN

Bahasa mesin adalah bahasa dalam bentuk kode-kode biner sebagai sandi operasi (Operation Code) dari sebuah mikroprosesor. Bahasa mesin adalah bahasa yang langsung berhubungan dengan mikroprosesor yang ditulis dan dikembangkan dari set instruksi. Tanpa bantuan set instruksi bahasa mesin sangat sulit dimengerti atau difahami. Untuk dapat menulis bahasa mesin maka penguasaan set instruksi sebuah mikroprosesor adalah wajib.

#### ❖ BAHASA ASSEMBLY

Bahasa Assembly adalah bahasa komputer yang kedudukannya di antara bahasa mesin dan bahasa aras tinggi misalnya bahasa C, C++, Pascal, Turbo Basic, Java, dan sebagainya. Bahasa C atau Pascal dikatakan sebagai bahasa level tinggi karena memakai kata-kata dan pernyataan yang mudah dimengerti manusia, meskipun masih jauh berbeda dengan bahasa manusia sesungguhnya.

Assembler adalah program yang bekerja membantu penulisan instruksi dalam format bahasa inggris sehingga mudah dibaca dan dipahami.

Untuk membuat proses pemrograman menjadi lebih mudah, kebanyakan programmer menuliskan program dalam bentuk bahasa assembly. Kemudian mereka menterjemahkan bahasa *assembly* yang ditulisnya menjadi bahasa mesin sehingga dapat di download ke memori dan di run atau dijalankan. Penterjemahan bahasa *assembly* menjadi kode biner bahasa mesin dapat dilakukan secara manual atau menggunakan program yang disebut dengan *assembler*.

Bahasa *assembly* menggunakan sejumlah mnemonik untuk merepresentasikan instruksi-instruksi. Mnemonik adalah singkatan dari suatu perintah atau instruksi sebagai piranti untuk membantu ingatan.

Sebagai contoh :

LOAD	disingkat	LD	artinya copy-kan
ADD	disingkat	ADD	artinya jumlah-kan
ADD With Carry	disingkat	ADC	artinya jumlah-kan sertakan carry
SUBTRACT	disingkat	SUB	artinya kurangkan
SUBTRACT With Carry	disingkat	SBC	artinya kurangkan sertakan carry
COMPLEMENT	disingkat	CPL	artinya not-kan

Konstruksi program bahasa assembly biasanya ditulis dalam bentuk standar seperti pola berikut.

Alamat	Kode Operasi	No. Urut	Label	Assembly	Keterangan
		1	Program 03 : Menyalakan Tulisan 123456		
		2	Selama 1 detik		
		3	SCANI	EQU 0624H	
		4		ORG 1800	
1800	DD 21 0C 18	5		LD IX, ANGKA	
1804	06 64	6		LD B, 64H	64H=100D > Pengali
		7			
1806	CD 24 06	8	Nyala	CALL SCANI	
1809	10 FB	9		DJNZ Nyala	
180B	76	10		HALT	
		11			
180C	AF	12	ANGKA	DB AFH	Data angka 6
180D	AE	13		DB AEH	Data angka 5
180E	36	14		DB 36H	Data angka 4
180F	BA	15		DB BAH	Data angka 3
1810	9B	16		DB 9BH	Data angka 2
1911	30	17		DB 30H	Data angka 1
		18		END	

Untuk memudahkan melakukan pengenalan program diberi judul atau identitas program. Program 03: bekerja menyalakan Tulisan 123456 selama 1 detik. EQU – singkatan dari *EQUATE*, dipakai untuk menentukan nilai sebuah simbol. EQU 0624H menyatakan subrutin SCAN1 berada pada alamat awal 0624H. ORG singkatan dari *ORIGIN*, untuk menyatakan awal alamat program di memori program. Misalnya **ORG 1800H** menyatakan program dimulai di memori alamat 1800H. Kode "H" berarti heksadesimal.

Kolom alamat menunjukkan alamat dari setiap perintah program. Perintah Assembly LD IX,ANGKA dengan kode operasi bahasa mesin DD 21 0C 18 menempati empat lokasi alamat mulai 1800 s/d 1803. LD B,64H dengan kode operasi 06 64 harus ditempatkan berurutan pada alamat 1804 dan 1805. Demikian juga dengan perintah lain seterusnya.

Kode operasi adalah kode bahasa mesin yang diterjemahkan dari tabel instruction set mikroprosesor. Kode operasi sama sekali tidak boleh salah sedikitpun. Kode operasi merupakan kode bahasa mesin yang dimasukkan ke memori komputer mikro untuk dieksekusi.

Nomor urut digunakan untuk menunjukkan urutan setiap perintah. Label digunakan untuk memberi keterangan alamat perintah. Label "ANGKA" berada pada alamat 180C. Perintah LD IX,ANGKA kode operasinya DD 21 0C 18. DD 21 kode operasi LD IX dan 0C 18 untuk label "ANGKA". Khusus untuk mikroprosesor Zilog Z-80 menggunakan format data LSB "0C" diikuti MSB "18".

Bagian komentar biasanya digunakan untuk memberi penjelasan singkat maksud atau sasaran dari instruksi disebelah kirinya.

Jika kita sudah menulis program dalam bahasa assembly, pertanyaannya bagaimana menterjemahkan menjadi bahasa mesin dalam bentuk kode operasi yang siap dieksekusi ke sistim mikroprosesor ?. Ada dua jawaban atas pertanyaan ini. Pertama secara manual dengan menterjemahkan setiap instruksi bahasa assembly menjadi bahasa mesin melalui set instruksi. Kemudian kedua secara otomatis dengan menggunakan *ASSEMBLER*. Assembler adalah sebuah program yang dapat dijalankan pada sebuah mikrokomputer atau pada sebuah *Microcomputer Development System*.

DB – singkatan dari *DEFINE BYTE*, dipakai untuk memberi nilai tertentu pada memori-program. Nilai tersebut merupakan nilai 1 byte, bisa berupa angka ataupun kode ASCII. DB merupakan Assembler Directive yang dipakai untuk membentuk teks maupun tabel.

DW – singkatan dari *DEFINE WORD*, dipakai untuk memberi nilai 2 byte ke *memori-program* pada baris bersangkutan. Assembler Directive ini biasa dipakai untuk membentuk suatu tabel yang isinya adalah nomor-nomor memori-program.

## ❖ BAHASA ARAS TINGGI

Cara lain dalam menulis program pada mikrokomputer adalah dengan bahasa aras tinggi seperti Bahasa C, bahasa Pascal, BASIC, FORTRAN, dan sebagainya. Secara umum bahasa aras tinggi dapat dikelompokkan menjadi dua yaitu sebagai **Compiler** atau **Interpreter**.

### c. Rangkuman

Sistim mikroprosesor atau komputer mikro bekerja menggunakan sandi operasi bilangan biner dalam kode-kode heksadesimal yang disebut dengan bahasa mesin. Karena alasan sulit membaca dan mengerti sandi operasi bahasa mesin maka diciptakan bahasa pemrogram assembly. Bahasa assembly adalah bahasa yang posisinya diantara bahasa mesin dan bahasa aras tinggi.

### d. Tes Formatif

**Pilihlah salah satu jawaban yang saudara anggap paling benar**

1. Bahasa Assembly adalah
  - a. Bahasa high level
  - b. Bahasa mesin
  - c. Bahasa diantara bahasa mesin dan bahasa high level
  - d. Bahasa Low level
2. Bahasa yang langsung dieksekusi di mesin komputer adalah
  - a. Bahasa high level
  - b. Bahasa mesin
  - c. Bahasa diantara bahasa mesin dan bahasa high level
  - d. Bahasa Low level
3. Bahasa mesin komputer dikodekan dalam
  - a. Perintah Assembly
  - b. Label
  - c. Sandi operasi
  - d. Keterangan
4. Perintah ORG 1900 menyatakan makna
  - a. Perintah Assembly 1900
  - b. Label 1900
  - c. Original 1900
  - d. Program dimulai dari alamat 1900
5. Pernyataan berikut yang benar
  - a. Bahasa Assembly bisa dieksekusi langsung
  - b. Bahasa mesin bisa dieksekusi langsung
  - c. Mnemonik merupakan perintah bahasa assembly
  - d. Mnemonik berarti kepanjangan

e. Kunci Jawaban

1. c
2. b
3. c
4. d
5. b