

Dasar-Dasar
Pemrograman Komputer

Trija Fayeldi

TRIJA FAYELDI, MSi

MILIK PRIBADI
MALANG
2014

KATA PENGANTAR

Pemrograman komputer telah menjadi salah satu kemampuan yang dituntut untuk dikuasai oleh para mahasiswa dari beragam bidang ilmu, termasuk pula dalam bidang matematika. Pemrograman komputer tidak sekedar mampu menuliskan beragam syntax dari suatu bahasa pemrograman, tetapi yang lebih mendasar adalah mampu memikirkan algoritma penyelesaian dari suatu permasalahan. Dengan menguasai algoritma penyelesaiannya, suatu permasalahan dapat dipecahkan atau diimplementasikan ke dalam berbagai bahasa pemrograman yang ada.

Buku ini akan mengupas masalah pemrograman komputer mulai dari dasar, yaitu algoritma. Kemudian, algoritma-algoritma yang telah dibahas ini akan diimplementasikan pada suatu bahasa pemrograman. Dalam hal ini, bahasa pemrograman yang dipilih adalah Matlab dengan alasan bahwa Matlab merupakan bahasa pemrograman yang paling umum digunakan pada matematika.

Buku ini merupakan awal dari suatu pekerjaan besar, yaitu membuat sebuah buku mengenai dasar-dasar pemrograman komputer yang cukup memadai bagi keperluan mahasiswa matematika dalam menempuh studi mereka. Beragam kritik dan saran dari pembaca tentu akan menjadi masukan bagi penulis.

Malang, September 2014

Trija Fayeldi, M.Si

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
I. Algoritma dan Komputer	1
1.1 Pengertian Algoritma	1
1.2 Kriteria Algoritma	1
1.3 Komputer	2
1.3.1 Input Unit	3
1.3.2 Output Unit	3
1.3.3 Main Memory	4
1.3.4 ALU	4
1.3.5 Control Unit	4
1.3.6 Secondary Memory	5
1.4 Program dan Pemrograman	5

1.5	Belajar Memprogram Versus Belajar Bahasa Pemrograman	6
1.6	Notasi Algoritmik	7
1.7	Kalimat Deskriptif	8
1.8	Flowchart	8
1.9	Pseudo-Code	12
1.10	Algoritma Tracing	15
II.	Tipe Data dan Variabel	18
2.1	Tipe Dasar	18
2.2	Tipe Bentukkan	19
2.3	Variabel	19
2.4	Pemberian Nilai	20
2.4.1	Assignment	20
2.4.2	Pembacaan	20
2.5	Menampilkan Nilai	21
2.6	Ekspresi	21
2.6.1	Ekspresi Aritmetika	21
2.6.2	Ekspresi Relasional	22
2.6.3	Ekspresi Logika	22
2.6.4	Ekspresi String	22
III.	Instruksi Pemilihan	24
3.1	Bentuk if ... then ... endif	25

3.2	Bentuk if ... then ... else ... endif	27
3.3	Bentuk Bersusun	29
3.4	Instruksi Case	31
IV.	Instruksi Pengulangan	35
4.1	Perulangan For	35
4.2	Perulangan While..do	38
4.3	Perulangan repeat...until	39
V.	Praktikum Matlab 1	43
5.1	Tipe Data Pada Matlab	43
5.2	Desktop Matlab	44
5.3	M-files	46
VI.	Praktikum Matlab 2	49
6.1	Perintah disp Versus fprintf	49
6.2	Instruksi Pemilihan	52
6.2.1	if...elseif...else...end	54
6.2.2	switch...case...otherwise...end	57
6.3	Instruksi Pengulangan	58
6.3.1	for...end	59
6.3.2	while..end	60
VII.	Praktikum Matlab 3	62
7.1	Fungsi dan Grafik	62

7.2	Animasi Grafik	64
-----	--------------------------	----

VIII	Soal Aneka Ragam	66
-------------	-------------------------	-----------

TRIJA FAYELDI, M.Si

DAFTAR GAMBAR

Gambar 1.1	Diagram blok komputer.	3
Gambar 1.2	Beberapa simbol flowchart	9
Gambar 1.3	Contoh flowchart	10
Gambar 1.4	Flowchart membandingkan dua bilangan berbeda	11
Gambar 3.1	Contoh flowchart 1.	26
Gambar 3.2	Contoh flowchart 2.	27
Gambar 3.3	Contoh flowchart 3.	28
Gambar 3.4	Flowchart bersusun 1.	29
Gambar 3.5	Flowchart bersusun 2.	30

Gambar 4.1	Flowchart For	37
Gambar 4.2	Flowchart while...do	38
Gambar 4.3	Flowchart repeat...until	40
Gambar 5.1	Tampilan Matlab.	45

TRIJA FAYELDI, M.Si

DAFTAR TABEL

Tabel 3.1	Ekspresi Relasional.	24
Tabel 5.1	Perbedaan struktur penulisan. .	46
Tabel 6.1	Format option pada fprintf . . .	50
Tabel 6.2	Ekspresi Relasional Pada Matlab.	52
Tabel 6.3	Aturan Penilaian.	56
Tabel 6.4	Daftar Harga.	58

Bab I

Algoritma dan Komputer

1.1 Pengertian Algoritma

Algoritma merupakan rangkaian instruksi yang dijalankan secara terurut untuk menyelesaikan suatu permasalahan. Algoritma diperkirakan berasal dari kata *Al-Khuwarizmi*, yaitu seorang ilmuwan yang menulis *Kitab Al Jabar Wal-Muqabala*.

1.2 Kriteria Algoritma

Kriteria-kriteria dari suatu algoritma yang baik antara lain sebagai berikut.

1. *Input*, yaitu memiliki masukan.
2. *Output*, yaitu memiliki keluaran.
3. *Definiteness*, yaitu memiliki instruksi yang jelas dan tidak ambigu .
4. *Finiteness*, yaitu memiliki titik henti .
5. *Effectiveness*, yaitu efektif dalam pelaksanaan

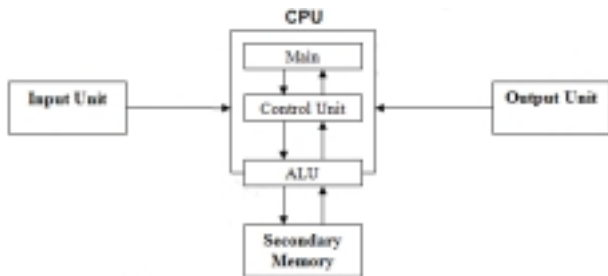
6. *Generality*, yaitu langkah-langkah algoritma yang diberikan berlaku untuk semua himpunan input yang diberikan, tidak hanya untuk himpunan input tertentu.

Contoh 1.2.1. *Contoh algoritma dalam bidang kimia misalnya "campurkan 10 ml asam asetat ke dalam 5 ml natrium klorida"*

Latihan 1.2.1. *Pikirkanlah algoritma untuk mencari bilangan terbesar dari tiga bilangan yang disediakan.*

1.3 Komputer

Diagram blok dari suatu sistem komputer dapat digambarkan seperti berikut.



Gambar 1.1: Diagram blok komputer.

Uraian dari setiap bagian adalah sebagai berikut.

1.3.1 Input Unit

Input unit (piranti masukan) berfungsi untuk memasukkan data dari pengguna ke memori komputer, contohnya keyboard dan card reader.

1.3.2 Output Unit

Output unit (piranti keluaran) berfungsi untuk mengeluarkan data dari memori komputer ke pengguna, contohnya monitor dan printer.

1.3.3 Main Memory

Main memory, disebut juga RAM (Random Access Memory) merupakan wadah untuk menampung semua data dan masukan yang akan diolah selanjutnya oleh komputer. Main memory bersifat temporer, artinya data yang tersimpan di memory tersebut akan hilang apabila komputer dimatikan.

1.3.4 ALU

ALU (Arithmetic and Logic Unit) berfungsi melakukan semua operasi aljabar dan logika pada komputer. ALU merupakan otak utama dari suatu komputer.

1.3.5 Control Unit

Control unit berfungsi mengatur seluruh aliran data yang ada di komputer. Control Unit dan ALU disebut juga sebagai CPU (Central Processing Unit).

1.3.6 Secondary Memory

Secondary Memory berfungsi untuk menyimpan semua data yang diinginkan agar tidak hilang saat komputer dimatikan.

1.4 Program dan Pemrograman

Algoritma baru akan efektif jika dijalankan oleh sebuah pemroses (*processor*). Pemroses itu dapat berupa manusia, mesin, dan komputer. Pemroses akan membaca setiap instruksi di dalam algoritma lalu mengerjakan instruksi tersebut. Sebuah pemroses harus memenuhi syarat-syarat berikut.

1. Mengerti setiap langkah di dalam algoritma.
2. Mengerjakan operasi yang bersesuaian dengan langkah tersebut.

Pada kuliah ini, pemroses yang dimaksud adalah sebuah komputer. Agar komputer dapat memahami algoritma yang diberikan maka algoritma harus ditulis dalam bahasa yang dipahami oleh komputer. Bahasa komputer yang digunakan untuk menulis program dinamakan bahasa pemrograman. Orang yang membuat suatu program komputer dinamakan pro-

grammer, dan kegiatan programmer dalam menulis program disebut pemrograman atau coding.

1.5 Belajar Memprogram Versus Belajar Bahasa Pemrograman

Belajar memprogram berbeda dengan belajar bahasa pemrograman. Belajar memprogram berarti mempelajari metodologi pemecahan masalah, kemudian menuliskannya dalam bentuk algoritma. Adapun belajar bahasa pemrograman berarti belajar menggunakan suatu bahasa pemrograman, termasuk aturan sintaks dan instruksinya.

Berdasarkan *tujuan aplikasinya*, bahasa pemrograman dapat dibagi menjadi dua kelompok, yaitu sebagai berikut.

1. Bahasa pemrograman bertujuan umum. Bahasa ini dapat digunakan untuk berbagai tujuan, misalnya Pascal, Visual Basic, dan Visual C++.
2. Bahasa pemrograman bertujuan khusus. Bahasa ini lebih spesifik digunakan untuk tujuan tertentu, misalnya Matlab dan SQL.

Adapun Bahasa pemrograman menurut *tingkatannya*, dapat dibagi menjadi tiga, yaitu sebagai berikut.

1. Bahasa mesin, yaitu bahasa yang dijalankan oleh komputer itu sendiri. Bahasa mesin ini tersusun atas rangkaian bilangan biner 0 dan 1.
2. Bahasa assembly, bahasa ini memiliki tingkatan yang sedikit di atas bahasa mesin. Bahasa ini menggunakan kata-kata sederhana, seperti MOV, ADD, atau STR. Agar dapat dimengerti oleh komputer, bahasa assembly perlu diterjemahkan menjadi bahasa mesin. Penerjemahnya dinamakan *assembler*.
3. Bahasa tingkat tinggi, bahasa ini menggunakan instruksi berupa kata-kata yang mirip dengan bahasa sehari-hari. Bahasa ini diterjemahkan ke dalam bahasa mesin dengan menggunakan *compiler*. Contoh bahasa tingkat tinggi antara lain Pascal, Fortran, dan Matlab.

1.6 Notasi Algoritmik

Notasi algoritmik adalah rancangan urutan langkah pencapaian solusi dalam bentuk deskriptif. Notasi algoritmik dapat ditulis dalam tiga cara berikut.

1. Runtunan kalimat deskriptif
2. Bagan alir/Flowchart

3. Pseudo-Code

1.7 Kalimat Deskriptif

Misalkan Anda akan membuat notasi algoritmik menghitung luas lingkaran berdasarkan jari-jarinya dengan menggunakan kalimat deskriptif.

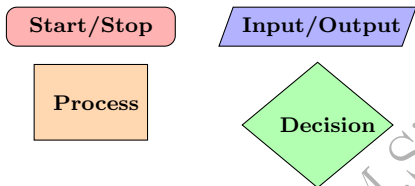
Program hitung_luas

Algoritma:

1. Masukkan r
2. kuadratkan r
3. kalikan dengan $3,14$

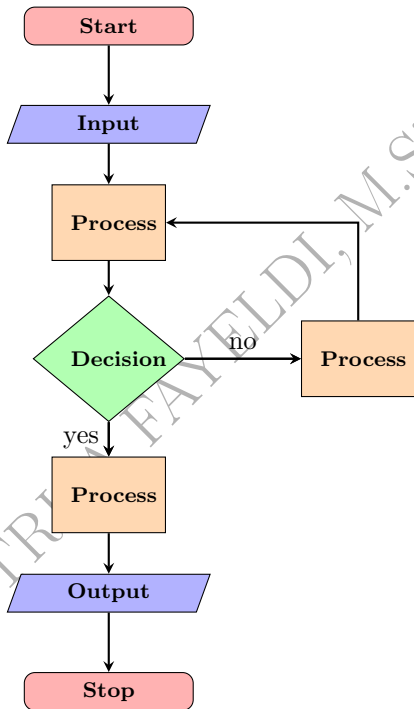
1.8 Flowchart

Flowcharting adalah rangkaian gambar yang menunjukkan aliran proses data/algoritma yang dibuat. Beberapa simbol flowchart yang sering digunakan antara lain sebagai berikut.



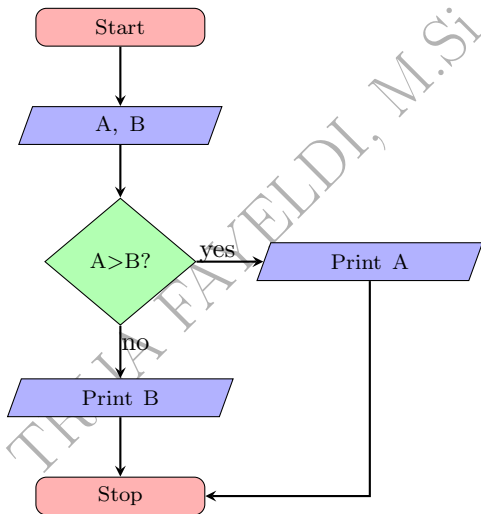
Gambar 1.2: Beberapa simbol flowchart

Suatu flowchart terdiri atas gabungan dari beberapa simbol yang dihubungkan dengan anak panah.



Gambar 1.3: Contoh flowchart

Contoh 1.8.1. Gambar berikut memperlihatkan flowchart untuk membandingkan dua bilangan berbeda.



Gambar 1.4: Flowchart membandingkan dua bilangan berbeda

Latihan 1.8.1. *Kerjakan soal-soal berikut.*

1. *Buatlah flowchart untuk menemukan bilangan terbesar dari tiga bilangan berbeda.*
2. *Buatlah flowchart untuk menghitung nilai rata-rata dari tiga nilai ujian seorang mahasiswa.*
3. *Seorang mahasiswa dikatakan lulus apabila nilai rata-rata UTS, UAS, dan Tugas lebih dari 55. Buatlah flowchart kelulusan mahasiswa tersebut.*
4. *Buatlah flowchart konversi suhu dari Celsius ke Fahrenheit dan Reamur dengan ketentuan user hanya dapat memilih salah satu jenis konversi saja.*

$${}^{\circ}F = \frac{9}{5} C + 32 \quad (1.1)$$

$${}^{\circ}R = \frac{4}{5} C \quad (1.2)$$

5. *Buatlah flowchart untuk menentukan apakah suatu bilangan itu merupakan bilangan genap atau ganjil.*

1.9 Pseudo-Code

Pseudo-Code adalah notasi algoritmik yang lebih menyerupai bahasa pemrograman tingkat tinggi, misalnya Pascal. Agar suatu algoritma dalam bentuk pseudo-code mudah dipahami, maka algoritma tersebut perlu ditulis

dalam struktur tertentu. Struktur dari suatu algoritma adalah sebagai berikut.

1. Header

Header memuat nama dan informasi mengenai algoritma yang ditulis.

2. Deklarasi

Deklarasi memuat definisi berbagai variabel dan tipe data yang digunakan dalam algoritma tersebut.

3. Deskripsi

Deskripsi memuat langkah-langkah penyelesaian masalah dengan menggunakan algoritma tersebut.

Berikut ini merupakan contoh algoritma menghitung luas lingkaran dengan menggunakan pseudo-code.

Contoh 1.9.1. *Berikut ini contoh algoritma menghitung luas lingkaran.*

Algoritma Luas Lingkaran

{Menghitung luas lingkaran dengan masukan jari-jari}

Deklarasi

real r, L

const π

Deskripsi

read(r)

$L \leftarrow \pi \times r^2$

write(L)

Latihan 1.9.1. *Jawablah soal-soal berikut.*

1. Sebutkan bagian-bagian yang merupakan struktur dari suatu algoritma.
2. Tuliskan algoritma untuk menghitung luas lingkaran dengan masukan berupa keliling lingkaran tersebut.
3. Diberikan algoritma sebagai berikut.

Algoritma coba

Deklarasi

real r, La, Va, Vb

const pi

Deskripsi

read(r)

$La \leftarrow pi \times r^2$

$Va \leftarrow \frac{1}{3} \cdot La$

$Vb \leftarrow 4 \cdot Va \cdot r$

write(Vb)

Berdasarkan outputnya, algoritma apakah ini? Bagaimanakah agar algoritma tersebut menjadi lebih efisien?

4. Apakah perbedaan antara input unit dan output unit?
5. Apakah perbedaan antara main memory dan secondary memory?

1.10 Algoritma Tracing

Definisi 1.10.1. *Algoritma tracing atau pelacakan algoritma adalah proses menjalankan suatu algoritma secara terurut langkah demi langkah.*

Tujuan dari algoritma tracing adalah untuk memeriksa aliran logika dan mencari kesalahan dari algoritma tersebut. Algoritma tracing dilakukan dengan mencoba berbagai kemungkinan masukan dari algoritma tersebut. Algoritma tracing tidak bertujuan untuk membuktikan kebenaran dari suatu algoritma, melainkan hanya memastikan bahwa tidak ditemukan kesalahan aliran logika pada algoritma tersebut.

Contoh 1.10.1. *Kerjakan contoh berikut.*

- 1. Buatlah algoritma untuk menemukan luas permukaan dari sebuah kubus dengan masukan panjang sisi kubus tersebut.*
- 2. Apakah algoritma yang Anda buat sudah sempurna?*
- 3. Bagaimana bila diberi masukan berupa bilangan negatif? Bagaimana caranya agar algoritma dapat menolak data negatif?*

Latihan 1.10.1. *Kerjakan soal-soal berikut.*

1. *Buatlah flowchart untuk menghitung luas segitiga dengan masukan panjang sisi a , b , dan c menggunakan Teorema Heron, yaitu*

$$L = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \frac{1}{2} \cdot \text{keliling.}$$

2. *Buatlah flowchart untuk menghitung keliling suatu persegi panjang dengan masukan panjang dan lebar persegi panjang tersebut.*
3. *Buatlah flowchart untuk membaca dua bilangan x dan y , kemudian pertukarkanlah isinya dengan menggunakan variabel tambahan.*
4. *Buatlah flowchart untuk membaca dua bilangan x dan y , kemudian pertukarkanlah isinya TANPA menggunakan variabel tambahan.*
5. *Buatlah flowchart untuk membaca tiga bilangan x , y , dan z . Kemudian, pertukarkan isinya dengan aturan isi x pindah ke y ; isi y pindah ke z ; dan isi z pindah ke x dengan menggunakan satu variabel tambahan.*
6. *Buatlah flowchart untuk membaca tiga bilangan x , y , dan z . Kemudian, pertukarkan isinya dengan aturan isi x pindah ke y ; isi y pindah ke z ; dan isi z pindah ke x TANPA menggunakan variabel tambahan.*
7. *Buatlah flowchart untuk menghitung akar-akar real*

dari suatu persamaan kuadrat dengan masukan berupa koefisien dari persamaan kuadrat tersebut. Jika diperkirakan akan menghasilkan akar yang bukan bilangan real maka pengguna harus memberi masukan baru.

8. Buatlah flowchart untuk menentukan nilai rata-rata \bar{x} dan standar deviasi σ dari lima buah data.

Rumus:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

Bab II

Tipe Data dan Variabel

Secara umum, terdapat dua macam tipe data pada suatu bahasa pemrograman, yaitu tipe dasar dan tipe bentukan.

2.1 Tipe Dasar

Tipe data yang termasuk pada tipe dasar antara lain sebagai berikut.

1. Integer, yaitu bilangan bulat misalnya -3 , 0 , dan 8 .
2. Real, yaitu bilangan real misalnya -9 , 12 , 2×10^8 , dan $0,12$.
3. Konstanta, yaitu bilangan yang nilainya selalu tetap di dalam suatu algoritma misalnya $\pi = 3,14$.
4. Karakter, yaitu semua huruf dan simbol yang tidak dioperasikan secara matematis misalnya a , C , dan $?$.
5. Boolean, yaitu tipe data yang hanya bernilai *benar* atau *salah*.

2.2 Tipe Bentuk

Tipe bentuk terdiri atas berbagai tipe berikut.

1. Larik, yaitu suatu tipe untuk menampung data sejenis. Dalam matematika, larik ini dapat dianalogikan sebagai sebuah vektor.
2. String, yaitu deretan karakter yang membentuk sebuah kata atau kalimat, misalnya nama dan alamat.
3. Record, yaitu tipe data untuk menampung elemen data yang tipenya tidak sama dengan tujuan untuk mewakili suatu objek, misalnya record data mahasiswa yang terdiri atas NIM, nama, dan umur.

2.3 Variabel

Variabel adalah identitas yang mewakili suatu elemen data, misalnya x , y , $nama$. Aturan dalam pemberian nama variabel antara lain sebagai berikut.

1. Harus dimulai dengan abjad, tidak diperkenankan dimulai dengan angka atau simbol.
2. Tidak boleh ada spasi.
3. Tidak menggunakan titik dua, titik, koma, dan sejenisnya.
4. Nama variabel berkaitan dengan elemen data.
5. Nama variabel pendek saja.

Contoh 2.3.1. *Contoh nama variabel yang benar antara*

lain Nama, TTL, dan Nilai_tes. Adapun contoh nama variabel yang salah antara lain $2x$, tanggal lahir, dan $x+y$.

2.4 Pemberian Nilai

Terdapat dua cara yang dapat digunakan untuk memberi nilai pada suatu variabel, yaitu assignment dan pembacaan. Berikut ini merupakan beberapa contohnya

2.4.1 Assignment

Pemberian nilai dengan assignment memiliki bentuk *variabel* \leftarrow *nilai*;

Contoh 2.4.1. Berikut beberapa contohnya.

1. $x \leftarrow 2$
2. $jarak \leftarrow 8$
3. $nama \leftarrow trija$
4. $x \leftarrow 5$
 $y \leftarrow 3$
 $z \leftarrow x + y$

2.4.2 Pembacaan

Pemberian nilai dengan pembacaan memiliki bentuk *read(variabel)*, contohnya *read(x)* dan *read(nama)*.

2.5 Menampilkan Nilai

Untuk menampilkan nilai dari suatu variabel digunakan perintah *write(variabel)*.

Contoh 2.5.1. Berikut beberapa contohnya.

1. *write("Nama Anda adalah ", nama);*
2. *write(x);*
3. *write("Jumlahnya adalah ", x + y)*

2.6 Ekspresi

Ekspresi adalah transformasi data atau variabel dalam bentuk persamaan. Berikut ini beberapa bentuk ekspresi.

2.6.1 Ekspresi Aritmetika

Ekspresi aritmetika yaitu ekspresi yang memuat operator aritmetika.

Contoh 2.6.1. Berikut beberapa contoh ekspresi aritmetika.

1. $x \leftarrow (y + 5) * t$
2. $total \leftarrow gaji pokok + lembur$
3. $luas \leftarrow 3,14 * r * r$

2.6.2 Ekspresi Relasional

Ekspresi relasional, yaitu ekspresi yang memuat operator relasional

Contoh 2.6.2. Berikut beberapa contoh ekspresi relasional.

1. $x > y$
2. $a < b$

2.6.3 Ekspresi Logika

Ekspresi ini memuat operator logika, yaitu AND dan OR.

Contoh 2.6.3. Berikut beberapa contoh ekspresi logika.

1. $x \leftarrow A \& \& B$
2. $n \leftarrow a || b$

2.6.4 Ekspresi String

Ekspresi string yaitu ekspresi yang memuat operator string.

Contoh 2.6.4. Berikut beberapa contoh ekspresi string.

1. $\text{nama} \leftarrow \text{"Trija"}$
2. $\text{email} \leftarrow \text{trija@gmail.com}$

Latihan 2.6.1. Kerjakan soal-soal berikut.

1. *Tulislah algoritma dari latihan-latihan yang terdapat pada Latihan 1.10.1*
2. *Buatlah algoritma harga total suatu barang, yaitu harga barang ditambah dengan pajaknya. Pajak barang tersebut adalah 25% dari harga barangnya.*
3. *Buatlah algoritma yang menampilkan NIM, nama, alamat, dan nomor handphone seorang mahasiswa.*
4. *Buatlah algoritma untuk menampilkan persentase keuntungan penjualan suatu barang dengan masukan harga beli dan harga jual barang tersebut.*
5. *Buatlah algoritma untuk menampilkan harga suatu barang setelah didiskon dengan masukan harga awal barang dan besaran diskon yang diberikan.*

Bab III

Instruksi Pemilihan

Instruksi pemilihan adalah suatu instruksi yang digunakan untuk memilih salah satu aksi bergantung pada terpenuhi atau tidaknya suatu syarat. Syarat yang diperiksa pada umumnya berupa ekspresi boolean, yaitu suatu ekspresi yang hanya bernilai *benar* atau *salah*, namun tidak keduanya.

Tabel 3.1: Ekspresi Relasional.

Notasi	Makna
<	kurang dari
>	lebih dari
=	sama dengan
<>	tidak sama dengan

Instruksi pemilihan yang akan Anda pelajari pada bagian ini antara lain instruksi **if ... then ... else endif**

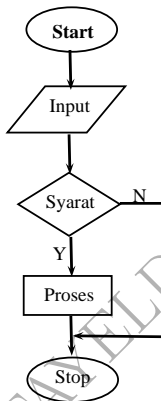
dan instruksi **case**.

3.1 Bentuk **if ... then ... endif**

Bentuk ini digunakan apabila terdapat satu syarat. Aksi tertentu akan dilakukan apabila syarat tersebut terpenuhi. Apabila syarat tidak terpenuhi, maka tidak ada aksi yang dilakukan. Deskripsi dari bentuk tersebut adalah sebagai berikut.

```
if (SYARAT)
    then (AKSI)
endif
```

Contoh flowchart dari bentuk tersebut dapat dilihat pada gambar berikut.



Gambar 3.1: Contoh flowchart 1.

Contoh 3.1.1. *Algoritma berikut akan menampilkan kata "LULUS" jika nilai mahasiswa lebih dari 60.*

Algoritma Lulus

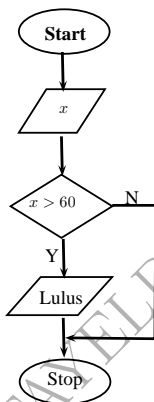
real : x

read(x)

if x > 60

 then write('LULUS')

endif

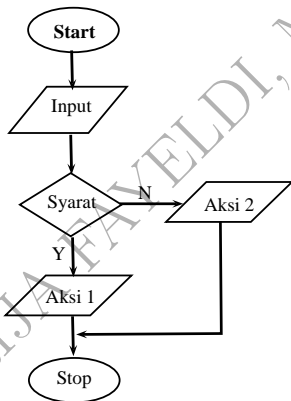


Gambar 3.2: Contoh flowchart 2.

3.2 Bentuk if ... then ... else ... endif

Bentuk ini digunakan apabila terdapat satu syarat. Aksi tertentu akan dilakukan apabila syarat tersebut terpenuhi. Apabila syarat tidak terpenuhi, maka ada aksi lain yang dilakukan. Deskripsi dari bentuk tersebut adalah sebagai berikut.

```
if (SYARAT)
  then (AKSI 1)
  else (AKSI 2)
endif
```



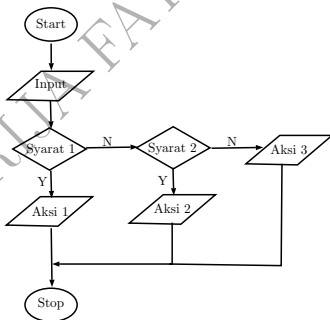
Gambar 3.3: Contoh flowchart 3.

Latihan 3.2.1. *Modifikasilah Contoh 3.1.1 agar dapat pula menampilkan kalimat "TIDAK LULUS" apabila syarat $x > 60$ tidak terpenuhi.*

3.3 Bentuk Bersusun

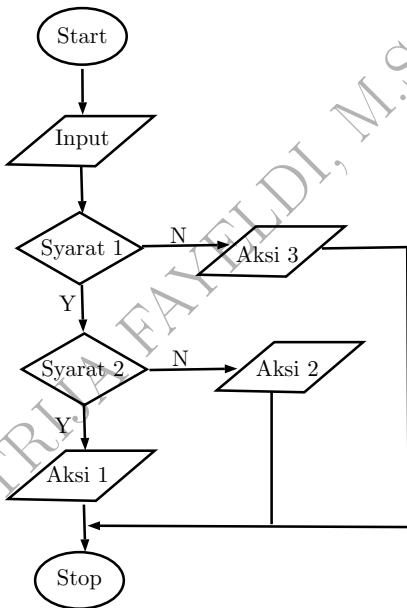
Selain kedua bentuk tadi, Anda dapat pula memodifikasi instruksi pemilihan ke dalam bentuk-bentuk lain sesuai dengan permasalahan yang dihadapi.

Latihan 3.3.1. *Tentukan deskripsi dari flowchart berikut.*



Gambar 3.4: Flowchart bersusun 1.

Latihan 3.3.2. Tentukan deskripsi dari flowchart berikut.



Gambar 3.5: Flowchart bersusun 2.

Latihan 3.3.3. *Buatlah flowchart dan deskripsi dari soal-soal berikut.*

1. *Tentukan semua akar real dari persamaan kuadrat $ax^2 + bx + c = 0$. Jika tidak ditemukan akar real, tampilkan output berupa kalimat Tidak memiliki akar real.*
2. *Menentukan semua akar dari $ax^2 + bx + c = 0$.*
3. *Menentukan jenis dari suatu segitiga dengan masukan panjang sisi a, b, c dengan c adalah sisi yang terpanjang dengan ketentuan:
Jika $c \geq a + b$ maka segitiga tidak mungkin terbentuk.
Jika $c^2 = a^2 + b^2$ maka jenisnya segitiga siku-siku.
Jika $c^2 > a^2 + b^2$ maka jenisnya segitiga tumpul.
Jika $c^2 < a^2 + b^2$ maka jenisnya segitiga lancip.*

3.4 Instruksi Case

Instruksi *case* digunakan sebagai instruksi pemilihan jika aksi yang akan dilakukan bergantung pada nilai variabel yang bersesuaian. Bentuk instruksi case adalah sebagai berikut.

case (VARIABEL)

 nilai 1: aksi 1

```
    nilai 2: aksi 2
    .
    .
    .
    default: aksi n
endcase
```

Sifat dari instruksi case antara lain sebagai berikut.

1. Terdapat n aksi.
2. Setiap aksi hanya dilakukan jika nilai variabel yang dimasukkan memenuhi syarat.
3. Apabila tidak ada satupun nilai variabel yang cocok maka aksi yang dijalankan adalah default.

Contoh 3.4.1. Berikut adalah contoh penggunaan case pada algoritma.

```
Algoritma upah
char: golongan;
integer: upah;
    read(golongan);
case (golongan)
'A': upah ← Rp1000
'B': upah ← Rp2000
'C': upah ← Rp3000
'D': upah ← Rp4000
'E': upah ← Rp5000
default: upah ← 0;
```

endcase
write(upah);

Latihan 3.4.1. *Kerjakan soal menentukan jenis segitiga pada Latihan 3.3.3 dengan menggunakan instruksi case.*

Latihan 3.4.2. *Kerjakan soal-soal berikut.*

- 1. Tulislah algoritma yang membaca sebuah bilangan bulat, lalu menuliskan pesan GENAP jika bilangan tersebut genap atau GANJIL jika bilangan tersebut ganjil.*
- 2. Buatlah algoritma untuk membaca dua bilangan bulat, lalu menentukan bilangan yang terbesar.*
- 3. Tuliskan algoritma untuk membaca tiga bilangan bulat lalu menentukan bilangan yang terbesar.*
- 4. Karyawan PT Kanjuruhan digaji berdasarkan jumlah jam kerjanya selama seminggu. Upah per jam Rp2000,00. Bila jumlah jam kerja lebih besar daripada 48 jam maka sisanya dianggap lembur. Upah lembur Rp3000,00 per jam. Tuliskan algoritma untuk menentukan upah mingguan karyawan dengan masukan jumlah jam kerja mereka.*
- 5. Suatu tahun disebut tahun kabisat jika memenuhi salah satu syarat berikut: habis dibagi 4 tetapi tidak habis dibagi 100; atau habis dibagi 400.*
- 6. Buatlah algoritma untuk menentukan apakah suatu bilangan bulat itu positif, negatif, atau nol.*

7. *Buatlah algoritma untuk menentukan wujud air (padat, cair, gas) pada suhu tertentu dengan masukan suhu air itu.*
8. *Buatlah algoritma untuk menentukan kuadran dari suatu titik dengan masukan koordinat titik tersebut.*
9. *Gunakan struktur CASE untuk mencetak nama bulan sesuai dengan angka bulan yang dimasukkan.*
10. *Buatlah program untuk menghitung salah satu dari hal berikut: 1)luas persegi panjang, 2)keliling persegi panjang, 3)panjang diagonal persegi panjang, atau 4)keluar dari program.*
11. *Buatlah algoritma untuk membaca sebuah bilangan bulat positif lalu menentukan apakah bilangan tersebut kelipatan 4.*
12. *Tulislah algoritma yang membaca tiga bilangan bulat, lalu mengurutkan ketiga bilangan itu mulai dari bilangan terkecil.*

Bab IV

Instruksi Pengulangan

Instruksi pengulangan adalah suatu instruksi untuk mengulangi pelaksanaan sederetan instruksi lain berulang kali sesuai dengan syarat yang ditetapkan. Instruksi pengulangan ada tiga jenis, yaitu **for**, **while..do**, dan **repeat...until**.

4.1 Perulangan For

Apabila dituliskan deskripsinya, maka skema perulangan **for** akan berbentuk seperti berikut.

```
for (var = awal to akhir step n)
    aksi
endfor
```

Contoh 4.1.1. *Algoritma berikut akan menampilkan bilangan asli 1 sampai dengan k dengan kenaikan sebesar 1.*

Algoritma tampil_bilangan

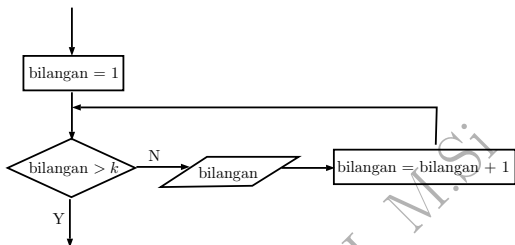
Deklarasi

```
integer: k, bilangan;
```

Deskripsi

```
read(k);
for (bilangan = 1 to k step 1)
    write(bilangan)
endfor
```

Flowchart dari Contoh 4.1.1 dapat dilihat pada gambar berikut.



Gambar 4.1: Flowchart For

Contoh 4.1.2. Algoritma berikut akan menampilkan bilangan asli k sampai dengan 1 dengan penurunan sebesar 1.

Algoritma tampil_bilangan

Deklarasi

```
integer: k, bilangan;
```

Deskripsi

```
read(k);
```

```
for (bilangan = k to 1 step -1)
```

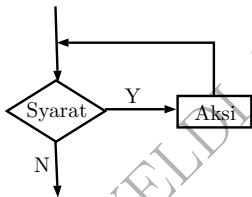
```
    write(bilangan)
```

```
endfor
```

Latihan 4.1.1. Tuliskan flowchart dari contoh di atas.

4.2 Perulangan While..do

Bentuk umum dari perulangan **while..do** adalah sebagai berikut.



Gambar 4.2: Flowchart while...do

Apabila dituliskan dalam bentuk deskripsi maka akan menjadi seperti berikut.

```
while (SYARAT) do
    AKSI
endwhile
```

Contoh 4.2.1. Berikut adalah algoritma modifikasi dari Contoh 4.1.1 dengan menggunakan **while ... do**.

Algoritma tampil_bilangan

Deklarasi


```
integer: k,bilangan;
```

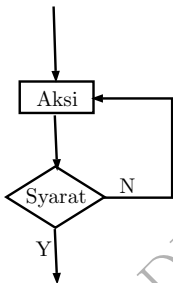
Deskripsi

```
read(k);  
bilangan <- 1;  
while (bilangan <= k) do  
    write(bilangan)  
    bilangan <- bilangan + 1;  
endwhile
```

Latihan 4.2.1. *Tuliskan algoritma Contoh 4.1.2 dengan menggunakan while...do.*

4.3 Perulangan repeat...until

Perulangan **repeat...until** dapat digambarkan dalam bentuk flowchart berikut.



Gambar 4.3: Flowchart repeat...until

Apabila dituliskan dalam bentuk deskripsi maka akan menjadi seperti berikut.

```
repeat  
    AKSI  
until (SYARAT)
```

Contoh 4.3.1. Berikut adalah modifikasi Contoh 4.1.1 dengan menggunakan *repeat...until*.

Algoritma tampil_bilangan

Deklarasi

```
integer: k, bilangan;
```

Deskripsi

```
read(k);  
bilangan <- 1;  
repeat  
  write(bilangan);  
  bilangan <- bilangan + 1;  
until (bilangan > k)
```

Latihan 4.3.1. *Modifikasilah Contoh 4.1,2 dengan menggunakan repeat...until.*

Latihan 4.3.2. *Tuliskan algoritma berikut dengan menggunakan for, while...do, dan repeat...until sesuai dengan keperluan.*

1. *Buatlah algoritma untuk menampilkan tulisan HELLO WORLD sebanyak 5 kali.*
2. *Buatlah algoritma untuk membaca integer tak negatif n dan menghitung faktorialnya.*
3. *Buatlah algoritma untuk menghitung $1+2+3+\dots+100$.*
4. *Buatlah algoritma untuk menghitung $1+3+5+\dots+99$.*
5. *Buatlah algoritma untuk menghitung $1 \times 2 \times 4 \times 7 \times 11 \times \dots \times n$, dengan $n < 100$.*
6. *Diketahui deret Fibonacci $S_1 = 1, S_2 = 1, S_n = S_{n-1} + S_{n-2}, \forall n \geq 3$. Buat algoritma untuk membaca nilai n , kemudian menghitung nilai S_n .*

7. *Buatlah algoritma dari syair lagu ANAK AYAM.*
8. *Buatlah algoritma untuk menghitung deret $1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots + \frac{1}{2^N}$ dengan masukan N .*
9. *Buatlah algoritma dari sebuah perpangkatan.*

TRIJA FAYELDI, M.Si

Bab V

Praktikum Matlab 1

Matlab merupakan sebuah singkatan dari Matrix Laboratory. Matlab dikenalkan untuk kali pertama oleh University of New Mexico dan University of Stanford. Matlab awalnya hanya digunakan untuk keperluan analisis numerik, aljabar linear, dan matriks. Akan tetapi, saat ini kemampuan dan fitur yang dimiliki oleh Matlab sudah jauh lebih lengkap dengan ditambahkannya beragam toolbox. Beberapa manfaat yang didapatkan dari Matlab antara lain sebagai berikut.

1. Perhitungan matematika.
2. Komputasi numerik.
3. Simulasi dan pemodelan.
4. Visualisasi dan analisis data.
5. Pembuatan grafik untuk keperluan sains dan teknik.
6. Pengembangan aplikasi berbasis General User Interface.

5.1 Tipe Data Pada Matlab

Tipe data yang dikenal dalam pemrograman Matlab hanya dua, yaitu *Numeric* dan *String*. Pada pemrogram-

an Matlab, tidak diperlukan deklarasi untuk menyatakan tipe data karena Matlab memiliki kemampuan tersendiri untuk mengenali tipe data yang dimasukkan oleh program pada setiap variabelnya. Beberapa hal penting yang harus Anda perhatikan dalam penulisan instruksi pada Matlab, yaitu sebagai berikut.

1. Variabel bersifat case sensitive, artinya Matlab akan membedakan adanya huruf besar dan kecil dalam penamaan variabel tersebut.
2. Panjang nama variabel maksimum 31 karakter.
3. Penamaan variabel harus selalu diawali dengan huruf, tidak boleh dengan bilangan, ataupun simbol.

5.2 Desktop Matlab

Saat Matlab dibuka, akan tampak desktop Matlab seperti gambar berikut.



Gambar 5.1: Tampilan Matlab.

Berbeda dengan Pascal, Matlab tidak memiliki struktur yang harus diikuti seperti pada struktur penulisan algoritma. Matlab tidak perlu menuliskan header dan deklarasi variabel di awal program. Untuk lebih jelasnya, kerjakan praktikum berikut.

Praktikum 5.2.1. Ketiklah perintah berikut pada Matlab dan perhatikan hal yang terjadi.

```
>> x = 'a'  
>> y = 100;  
>> z = 'makanan';
```

```
>> whos
```

Berikut ini disajikan tabel beberapa perbedaan dasar antara struktur algoritma dan struktur pemrograman Matlab.

Tabel 5.1: Perbedaan struktur penulisan.

Algoritma	Matlab
header	tak ada header
deklarasi	tak ada deklarasi
read(x)	$x = \text{input}(\dots)$
write(x)	disp(x)

5.3 M-files

Program-program yang ada kebanyakan tidak hanya terdiri atas beberapa baris saja. Banyak sekali program yang terdiri atas puluhan, bahkan ratusan baris. Untuk itu, pengetikkan program perlu dilakukan di sebuah tempat khusus di Matlab yang bernama M-files.

Praktikum 5.3.1. *Kerjakan program berikut pada M-files.*

```
r = input('Masukkan jari-jari lingkaran: ');  
L = pi*r^2;  
disp('Luas lingkaran adalah ');  
disp(L)
```

Praktikum 5.3.2. *Buatlah program Matlab dari permasalahan berikut.*

1. *Buatlah program untuk menghitung nilai rata-rata dari tiga nilai ujian seorang mahasiswa.*
2. *Buatlah program untuk menghitung luas segitiga dengan masukan panjang sisi a , b , dan c menggunakan Teorema Heron, yaitu*

$$L = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \frac{1}{2} \cdot \text{keliling.}$$

3. *Buatlah program untuk menghitung keliling suatu persegi panjang dengan masukan panjang dan lebar persegi panjang tersebut.*
4. *Buatlah program untuk membaca dua bilangan x dan y , kemudian pertukarkanlah isinya TANPA menggunakan variabel tambahan.*
5. *Buatlah program untuk membaca tiga bilangan x , y , dan z . Kemudian, pertukarkan isinya dengan aturan isi x pindah ke y ; isi y pindah ke z ; dan isi*

z pindah ke *x* dengan menggunakan satu variabel tambahan.

6. Buatlah program untuk menentukan nilai rata-rata \bar{x} dan standar deviasi σ dari lima buah data.

Rumus:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

TRIJA FAYELDI, M.Si

Bab VI

Praktikum Matlab 2

6.1 Perintah disp Versus fprintf

`disp(x)` akan mengeluarkan nilai dari suatu variabel x tanpa mencetak nama variabel tersebut.

```
>> x = 2;
>> y = 'Hello World';
>> disp(x);
>> disp(y);
```

Apabila nilai variabel yang akan ditampilkan lebih dari satu, maka perlu dilakukan modifikasi pada perintah tadi. Sebagai contoh, ketikkan program berikut.

```
>> nama = 'Tedi';
>> umur = 18;
>> tampil1 = [nama, ' saat ini berumur ',num2str(umur),' tahun.'];
>> disp(tampil1);
>> tampil2 = sprintf('%s saat ini berumur %d tahun.',nama,umur);
>> disp(tampil2);
>> fprintf('%s saat ini berumur %d tahun.',nama,umur)
```

Tabel 6.1: Format option pada **fprintf**

Perintah	Fungsi
%d	cetak sebagai bilangan bulat
%f	cetak sebagai bilangan desimal
%s	cetak sebagai string
%c	cetak sebagai karakter tunggal
\n	pindah baris
\%	cetak simbol persen

Sekarang, cobalah Anda ketikkan perintah-perintah berikut pada command window dan amati perbedaannya.

```
>> fprintf('x adalah %d',x);
>> fprintf('x adalah %f',x);
>> fprintf('x adalah %.2f',x);
>> fprintf('x adalah %s',x);
>> fprintf('x adalah %c',x);
>> fprintf('x adalah %d',x);fprintf('selesai');
>> fprintf('x adalah %d\n',x);fprintf('selesai');
```

Praktikum 6.1.1. *Perhatikan output berikut.*

```
=====
PROGRAM MENGHITUNG RATA-RATA
=====
Masukkan nilai 1: 2
Masukkan nilai 2: 5
Masukkan nilai 3: 7
Nilai rata-rata mahasiswa adalah 4.667
```

Buat listing program untuk menampilkan output tersebut.

Praktikum 6.1.2. *Perhatikan output berikut.*

```
=====
PROGRAM MENGHITUNG LUAS SEGITIGA
=====
Masukkan panjang sisi a: 3
Masukkan panjang sisi b: 4
Masukkan panjang sisi c: 5
-----
Luas 6.00 dan keliling 12.00
```

Buat listing program untuk menampilkan output tersebut.

6.2 Instruksi Pemilihan

Sebelum memahami implementasi instruksi pemilihan pada Matlab, ada baiknya Anda mengenal berbagai operasi relasional pada Matlab, antara lain sebagai berikut.

Tabel 6.2: Ekspresi Relasional Pada Matlab.

Notasi	Makna
<	kurang dari
>	lebih dari
<=	kurang dari atau sama dengan
>=	lebih dari atau sama dengan
==	sama dengan
~=	tidak sama dengan

Instruksi pemilihan dapat berbentuk:

1. `if (SYARAT)`
 `(AKSI)`
`end;`

- ```
2. if (SYARAT 1)
 (AKSI 1)
 else
 (AKSI 2)
 end;

3. if (SYARAT 1)
 (AKSI 1)
 elseif (SYARAT 2)
 (AKSI 2)
 else
 (AKSI 3)
 end;

4. switch (VARIABEL YANG AKAN DIPERIKSA NILAINYA)
 case (SYARAT KASUS 1)
 (AKSI 1)
 case (SYARAT KASUS 2)
 (AKSI 2)
 .
 .
 otherwise
 (AKSI JIKA TIDAK ADA
 SYARAT KASUS YANG TERPENUHI)
 end
```

## 6.2.1 if...elseif...else...end

**Contoh 6.2.1.** Berikut ini adalah contoh program untuk menampilkan kata *LULUS* jika nilai seorang mahasiswa lebih dari 60.

```
clc;
clear all;
x = input('Masukkan nilai: ');

if x > 60
 disp('LULUS')
end;
```

**Praktikum 6.2.1.** Modifikasilah Praktikum 6.2.1 agar dapat pula menampilkan kata *TIDAK LULUS* jika nilai yang diraih kurang dari atau sama dengan 60.

**Praktikum 6.2.2.** Modifikasilah Praktikum 6.2.1 agar dapat pula menampilkan kata *PAS LULUS* jika nilai yang diraih sama dengan 60.

**Latihan 6.2.1.** Kerjakan soal-soal berikut.

1. Buatlah program untuk menentukan semua akar real dari persamaan kuadrat  $ax^2 + bx + c = 0$ . Jika tidak ditemukan akar real, tampilkan output berupa kalimat *Tidak memiliki akar real*.



2. *Buatlah program untuk menentukan apakah suatu bilangan bulat  $x$  adalah bilangan genap ataukah bilangan ganjil.*
3. *Menentukan jenis dari suatu segitiga dengan masukan panjang sisi  $a, b, c$  dengan  $c$  adalah sisi yang terpanjang dengan ketentuan:  
Jika  $c \geq a + b$  maka segitiga tidak mungkin terbentuk.  
Jika  $c^2 = a^2 + b^2$  maka jenisnya segitiga siku-siku.  
Jika  $c^2 > a^2 + b^2$  maka jenisnya segitiga tumpul.  
Jika  $c^2 < a^2 + b^2$  maka jenisnya segitiga lancip.*

Terkadang, Anda perlu mengevaluasi dua syarat sekaligus untuk memeriksa suatu kondisi. Untuk itu, diperlukan operator logika. Operator logika yang umumnya digunakan adalah AND dan OR. Pada Matlab, operator logika AND dinotasikan dengan  $\&$  dan operator logika OR dinotasikan dengan  $|$ . Berikut ini beberapa contoh penggunaannya.

**Contoh 6.2.2.** *Salinlah program berikut.*

```
clc;
clear all;

x = input('Masukkan x: ');
y = input('Masukkan y: ');
```

```
if x ~= 2 & y == 3
 disp('Sesuai dengan syarat');
else
 disp('Tidak sesuai dengan syarat')
end;
```

*Coba Anda beri masukan berupa bilangan, karakter, dan string. Gunakan tanda ' ' apabila masukan berupa karakter atau string. Kemudian, amati perbedaannya. Lakukan hal yang sama untuk operator logika |.*

**Praktikum 6.2.3.** *Buatlah program untuk menuliskan nilai huruf seorang mahasiswa dengan ketentuan sebagai berikut.*

Tabel 6.3: Aturan Penilaian.

| Nilai Angka $x$  | Nilai Huruf |
|------------------|-------------|
| $0 \leq x < 55$  | E           |
| $55 \leq x < 65$ | D           |
| $65 \leq x < 75$ | C           |
| $75 \leq x < 85$ | B           |
| $x \geq 85$      | A           |

## 6.2.2 switch...case...otherwise...end

Pada beberapa kasus, bentuk pemilihan dapat disederhanakan penulisannya dalam bentuk **switch...case...end**. Perhatikan contoh berikut.

**Contoh 6.2.3.** *Program berikut akan menampilkan nama asli tokoh pemeran Pandawa.*

```
clear all;
clc;

nama = input('Masukkan nama tokoh Pandawa: ');

switch nama
 case 'Yudhistira'
 fprintf('%s: Rohit Bharadwaj',nama);
 case 'Bima'
 fprintf('%s: Saurav Gurjar',nama);
 case 'Arjuna'
 fprintf('%s: Shaheer Sheikh',nama);
 case 'Nakula'
 fprintf('%s: Vin Rana',nama);
 case 'Sadewa'
 fprintf('%s: Lavanya Bhardwaj',nama);
 otherwise
 fprintf('%s bukan tokoh Pandawa',nama)
end
```

**Praktikum 6.2.4.** *Berikut disajikan tabel stok barang di sebuah toko.*

Tabel 6.4: Daftar Harga.

| Kode | Nama        | Harga        | Diskon |
|------|-------------|--------------|--------|
| 100  | Baju anak   | Rp50.000,00  | 10%    |
| 110  | Baju dewasa | Rp120.000,00 | 20%    |
| 120  | Jaket       | Rp300.000,00 | 15%    |

*Seorang pembeli memilih salah satu barang dari toko tersebut dengan memilih kode barang. Tampilkan nama barang, harga awal barang sebelum diskon, harga diskon, dan harga barang yang harus dibayar setelah didiskon dengan menggunakan instruksi switch case. Jika kode barang yang dimasukkan tidak terdaftar maka tampilkan informasi bahwa kode tersebut tidak terdaftar, sehingga harga dan diskonnya bernilai 0.*

## 6.3 Instruksi Pengulangan

Pada Matlab, terdapat dua instruksi pengulangan yang dapat digunakan, yaitu instruksi **for...end** dan **while...end**.

### 6.3.1 for...end

Bentuk **for...end** pada Matlab dengan variabel awal  $a$ , step  $h$ , dan variabel akhir  $b$  adalah

```
for a:h:b
 (AKSI)
end;
```

Apabila nilai  $h = 1$  maka  $h$  tidak perlu ditulis.

**Praktikum 6.3.1.** Program berikut akan menampilkan 5 bilangan asli pertama.

```
clc;
clear all;
for r = 1:5
 disp(r);
end;
```

**Praktikum 6.3.2.** Kerjakan soal-soal berikut.

1. Buatlah program untuk menampilkan 5 bilangan asli pertama secara menurun.
2. Buatlah program untuk menampilkan bilangan ganjil mulai dari 1 hingga 15.
3. Buatlah program untuk menampilkan bilangan mulai dari  $-4,25$  hingga  $2,8$  dengan kenaikan  $0,25$
4. Buatlah program untuk menampilkan bilangan 1 hingga 10 dan tentukan jenis bilangan-bilangan tersebut apakah genap atau ganjil.

## 6.3.2 while..end

Bentuk instruksi **while...end** dapat ditulis seperti berikut.

```
while (SYARAT)
 (AKSI)
end
```

**Contoh 6.3.1.** Program berikut akan menampilkan 5 bilangan asli pertama.

```
clc;
clear all;
bil = 1;
while bil <= 5
 disp(bil)
 bil = bil+1;
end
```

Coba Anda hapus baris `bil = bil + 1`. Apakah yang terjadi?

**Praktikum 6.3.3.** Kerjakan soal-soal berikut.

1. Buatlah program untuk menampilkan 5 bilangan asli pertama secara menurun dengan menggunakan **while..end**.
2. Buatlah program untuk menampilkan bilangan ganjil mulai dari 1 hingga 15 dengan menggunakan **while..end**.

3. *Buatlah program untuk menampilkan bilangan mulai dari  $-4,25$  hingga  $2,8$  dengan kenaikan  $0,25$  dengan menggunakan **while..end***
4. *Buatlah algoritma untuk menghitung  $1+2+3+\dots+100$ . Kemudian, tampilkan di layar tulisan berupa  $1+2+3+\dots+100=5050$  dengan memanfaatkan instruksi pengulangan dan pemilihan.*
5. *Buatlah algoritma untuk menghitung  $1+3+5+\dots+99$ .*
6. *Buatlah algoritma untuk membaca integer tak negatif  $n$  dan menghitung faktorialnya. Kemudian, tampilkan tulisan misalnya  $3! = 1 \times 2 \times 3 = 6$ .*
7. *Buatlah algoritma untuk menghitung  $1 \times 2 \times 4 \times 7 \times 11 \times \dots \times n$ . Kemudian, tampilkan tulisan misalnya  $1 \times 2 \times 4 = 8$ .*
8. *Diketahui deret Fibonacci  $S_1 = 1, S_2 = 1, S_n = S_{n-1} + S_{n-2}, \forall n \geq 3$ . Buat algoritma untuk membaca nilai  $n$ , kemudian menghitung nilai  $S_n$ . Kemudian, tampilkan barisan Fibonaccinya, misalkan 1123.*

# Bab VII

---

## Praktikum Matlab 3

---

### 7.1 Fungsi dan Grafik

Sebagai orang matematika, terkadang Anda perlu untuk melakukan pemetaan dari suatu domain  $x$  ke sebuah fungsi  $f$ . Matlab dapat membantu Anda melakukan hal tersebut dengan perintah *inline*. Secara umum, perintah itu berbentuk

```
f = inline('FUNGSI','VARIABEL 1',... 'VARIABEL n');
```

**Contoh 7.1.1.** Fungsi  $f(x) = x^2 + 2x - 1$  dapat Anda definisikan dengan menggunakan perintah berikut.

```
f = inline('x.^2 + 2*x - 1')
```

Untuk mengevaluasi nilai dari  $f$  di suatu titik  $x = x_0$  cukup dituliskan  $f(x_0)$ . Cobalah Anda ketikkan  $f(0)$ ,  $f(-1)$ , dan  $f(5)$ . Kemudian perhatikan hasilnya.

Anda dapat pula mengevaluasi nilai dari  $f$  pada suatu selang  $[a, b]$  dengan cara membentuk vektor  $x$  dengan step  $h$  tertentu yang kecil dengan perintah

```
x = a:h:b
```



Adapun untuk melakukan plot fungsi dengan sumbu datar  $x$  dan sumbu tegak  $y = f(x)$ , dapat digunakan perintah

```
plot(x,y)
```

**Contoh 7.1.2.** Untuk mengevaluasi dan membuat grafik nilai-nilai fungsi  $f(x) = x^2 + 2x - 1$  pada  $x \in [-5, 5]$  dapat digunakan perintah berikut.

```
clc;
clear all;
f = inline('x.^2 + 2*x - 1','x');

x = -5:0.1:5;
y = f(x);
end
plot(x,y)
title('Grafik Fungsi y = f(x)')
xlabel('x')
ylabel('y')
```

**Latihan 7.1.1.** Gambarkan grafik dari fungsi  $y = f(x)$  berikut.

1.  $f(x) = x$  untuk  $x \in [0, 7]$ .
2.  $f(x) = \sin(x)$  untuk  $x \in [0, 2\pi]$ .
3.  $f(x) = \sqrt{x}$  untuk  $x \in [0, 8]$ .

$$4. f(x) = \frac{1}{x} \text{ untuk } x \in [-3, 3].$$

$$5. (f(x) = e^{-x^2} \text{ untuk } x \in [-2, 2])$$

## 7.2 Animasi Grafik

Persamaan garis lurus dapat ditulis sebagai  $y = mx$ , dengan  $m$  adalah gradien garis. Anda dapat membuat sebuah animasi yang menggambarkan perubahan yang terjadi pada grafik apabila  $m$  diubah-ubah nilainya mulai dari 0 hingga 5. Caranya adalah sebagai berikut.

```
clc;
clear all;
clf;

f = inline('x','x');

x = 0:0.01:5;
m = 1:0.1:4;
nm = length(m);

for r = 1:nm
y = m(r)*f(x);
plot(x,y);
axis([0 5 0 20])
title('Grafik y = mx')
```

```
xlabel('x')
ylabel('y')
drawnow;
pause(0.05)
end
```

**Latihan 7.2.1.** *Salinlah program berikut.*

```
clc;
clear all;
clf;

f = inline('x.^2','x');
x = -3:0.1:3;
a = -5:0.1:5;
na = length(a);;
for r = 1:na
 a(r)
 y = a(r)*f(x);
 plot(x,y);
 axis([-3 3 -10 10])
 drawnow;
 pause(0.05)
end
```

## Bab VIII

---

### Soal Aneka Ragam

---

Buatlah program Matlab dari algoritma berikut.

- ```
1. read(angka1);
   read(angka2);
   if (angka1 > angka2)
       then write('Angka yang lebih besar adalah ');
           write (angka1);
       else
           write('Angka yang lebih besar adalah ');
           write(angka2);
       endif
```
- ```
2. write('Masukkan panjang alas: ');
 read(alas);
 write('Masukkan tinggi: ');
 read(tinggi);
 luas <- alas*tinggi/2;
 write('Luas segitiga = ');
 write(luas)
```
- ```
3. write('Masukkan nama pegawai: ');
   read(nama);
   write('Masukkan gaji pokok: ');
```

```
read(gajipokok)
```

```
tunjangan <- 0.25*gajipokok;
```

```
pajak <- 0.15*(gajipokok + tunjangan);
```

```
gajibersih <- gajipokok + tunjangan - pajak
```

```
write('Gaji ');
```

```
write(nama);
```

```
write('adalah ');
```

```
write(gajibersih)
```

```
4. write('Masukkan bilangan 1: ');
```

```
read(A);
```

```
write('Masukkan bilangan 2: ');
```

```
read(B);
```

```
write('Masukkan bilangan 3: ');
```

```
read(C);
```

```
if (A > B)
```

```
  then maks <- A;
```

```
  else maks <- B;
```

```
endif
```

```
if (C > maks)
```

```
  then maks <- C;
```

```
endif
```

Buatlah program Matlab berdasarkan pola bilangan berikut dengan masukan n , yaitu banyaknya pola yang akan dimunculkan.

1. Pola: 1 22 333 4444 55555 666666....
2. Pola: 666666 55555 4444 333 22 1....
3. Pola: 1 12 123 1234 12345 123456....
4. Pola: ... 654321 54321 4321 321 21 1
5. Pola: 1 22 123 4444 12345 666666....
6. Pola: 1 5 3 7 5 9 7 11 9 13 11 15,....
7. Pola: 3 9 4 12 7 21 16 48 43 129 ,
8. Pola: 1 2 4 7 8 10 13 14 16 19 20 22 25,...
9. Pola: 1 123 33 1234 55555 123456
10. Pola: 2 12 7 17 12 22 17 27 22 32 , ...

Buatlah grafik dari fungsi berikut pada Matlab.

1. $f(x) = x + 1, \forall x \in [-5, 5], x \in \mathbb{R}$.
2. $f(x) = x^2 - 8, \forall x \in [-3, 7], x \in \mathbb{R}$.
3. $f(x) = x^3 + 2x^2 - 1, \forall x \in [-10, 10], x \in \mathbb{R}$.
4. $f(x) = \sqrt{x^2 - 5} * x + 8, \forall x \in [-4, 7], x \in \mathbb{R}$.
5. $f(x) = |x|, \forall x \in [-6, 6], x \in \mathbb{R}$.

Buatlah program Matlab dari soal-soal berikut.

1. Program untuk menghitung $1 \times 2 \times 4 \times 7 \times 11 \times \dots$.
Kemudian, tampilkan tulisan misalnya $1 \times 2 \times 4 = 8$.
2. Diketahui deret Fibonacci $S_1 = 1, S_2 = 1, S_n = S_{n-1} + S_{n-2}, \forall n \geq 3$. Buat program untuk mem-

baca nilai n , kemudian menghitung nilai S_n . Kemudian, tampilkan barisan Fibonacci-nya, misalkan 1123.

3. INPUT:

Masukkan banyak anak ayam: 3

OUTPUT:

Anak ayam turunlah 3, mati satu tinggal 2

Anak ayam turunlah 2, mati satu tinggal 1

Anak ayam tinggalah 1, mati 1 tinggal induknya

4. INPUT:

=====

PROGRAM BANYAK BILANGAN

masukkan bilangan: 2

masukkan bilangan: 5

masukkan bilangan: 1

masukkan bilangan: 0

=====

OUTPUT:

Banyak bilangannya ada 4

5. INPUT:

=====

PROGRAM MENJUMLAHKAN BILANGAN

```
-----  
masukkan bilangan: 1  
masukkan bilangan: 10  
masukkan bilangan: 3  
masukkan bilangan: 0  
=====
```

OUTPUT:

Ada 4 bilangan dengan jumlah 14

6. INPUT:

PROGRAM MENJUMLAHKAN BILANGAN

```
-----  
masukkan bilangan: 2  
masukkan bilangan: 6  
masukkan bilangan: 4  
masukkan bilangan: 0  
=====
```

OUTPUT:

Banyak bilangannya 4
Jumlahnya 12
Rata-ratanya 3.000

7. Program Menghitung a^b

Dengan a,b Bilangan Bulat

Masukkan a: 2

Masukkan b: 5

$2^5 = 32$

8. INPUT:

=====

Masukkan absis titik pertama: 3

Masukkan ordinat titik pertama: 0

Masukkan absis titik kedua: 0

Masukkan ordinat titik kedua: 4

=====

OUTPUT:

Jarak titik (3,0) dan (0,4) adalah 5.00

9. INPUT:

PROGRAM Mencari Warna Sekunder

=====

KODE WARNA	WARNA SEKUNDER
------------	----------------

100	Merah	Merah + Kuning = Jingga
-----	-------	-------------------------

200	Biru	Biru + Kuning = Hijau
-----	------	-----------------------

300	Kuning	Merah + Biru = Ungu
-----	--------	---------------------

=====

Masukkan kode warna primer 1: 200

Masukkan kode warna primer 2: 100

OUTPUT:

Campurannya warna ungu

10. INPUT:

=====

Masukkan jam: 2

Masukkan menit: 35

Masukkan detik: 42

=====

OUTPUT:

2 jam + 35 menit + 42 detik = 9342 detik

11. INPUT:

Masukkan detik: 5243

OUTPUT:

5243 detik = 1 jam + 27 menit + 23 detik

12. INPUT:

=====

Waktu Keberangkatan

Jam: 1

Menit: 43

Detik: 23

Waktu Kedatangan

Jam: 5

Menit: 12

Detik: 1

=====

OUTPUT:

Lama Perjalanan: 3 jam 28 menit 38 detik