

BAB 1

Pengantar Algoritma dan Pemrograman

Tujuan Pembelajaran

1. Mahasiswa mampu menjelaskan pengertian program komputer
2. Mahasiswa mampu menjelaskan pengertian algoritma
3. Mahasiswa mampu menjelaskan perbedaan penyajian algoritma dengan bahasa dan penyajian algoritma dengan *flowchart*
4. Mahasiswa mampu menyusun algoritma dan menyajikannya dalam bentuk bahasa maupun *flowchart*
5. Mahasiswa mampu menjelaskan langkah-langkah penyusunan program komputer
6. Mahasiswa mampu menjelaskan kriteria program komputer yang baik

BAB 2

Struktur Dasar Penyajian Algoritma

Tujuan Pembelajaran

1. Mahasiswa mampu menjelaskan struktur dasar penyajian algoritma
2. Mahasiswa mampu merancang algoritma dalam bentuk struktur dasar urut (*sequence*)
3. Mahasiswa mampu merancang algoritma dalam bentuk struktur dasar pemilihan (*selection*)
4. Mahasiswa mampu merancang algoritma dalam bentuk struktur dasar pengulangan (*repetition*)

Algoritma yang merupakan langkah-langkah penyelesaian masalah dapat dibedakan menjadi beberapa struktur dasar, yaitu Struktur Urut (*Sequence*), Struktur Pemilihan (*Selection*), dan Struktur Pengulangan (*Repetition*).

A. Struktur Dasar Urut (*Sequence*)

Struktur urut adalah struktur yang digunakan untuk mengerjakan jenis program yang pernyataannya sequential atau berurutan. Pada struktur ini, perintah yang diberikan secara beruntun atau berurutan baris per baris mulai dari awal hingga akhir. Struktur urut tidak memuat lompatan atau pengulangan didalamnya.

Intruksi dalam struktur urut memiliki karakteristik seperti:

1. Tiap perintah dikerjakan satu per satu sebanyak sekali
2. Pelaksanaan perintah dilakukan secara berurutan
3. Perintah terakhir merupakan akhir dari algoritma
4. Perubahan urutan dapat menyebabkan hasil yang berbeda

Struktur urut dalam sains biasanya digunakan untuk melakukan perhitungan pada kasus yang melibatkan rumus-rumus sederhana dengan melibatkan operator penjumlahan, pengurangan, dan perkalian. Beberapa contoh kasus yang dapat menerapkan logika dengan struktur urut adalah perhitungan suatu besaran dengan rumus sederhana seperti jarak tempuh, luas persegi panjang, luas lingkaran, perhitungan upah pegawai, dan sejenisnya.

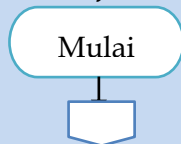
CONTOH 2.1

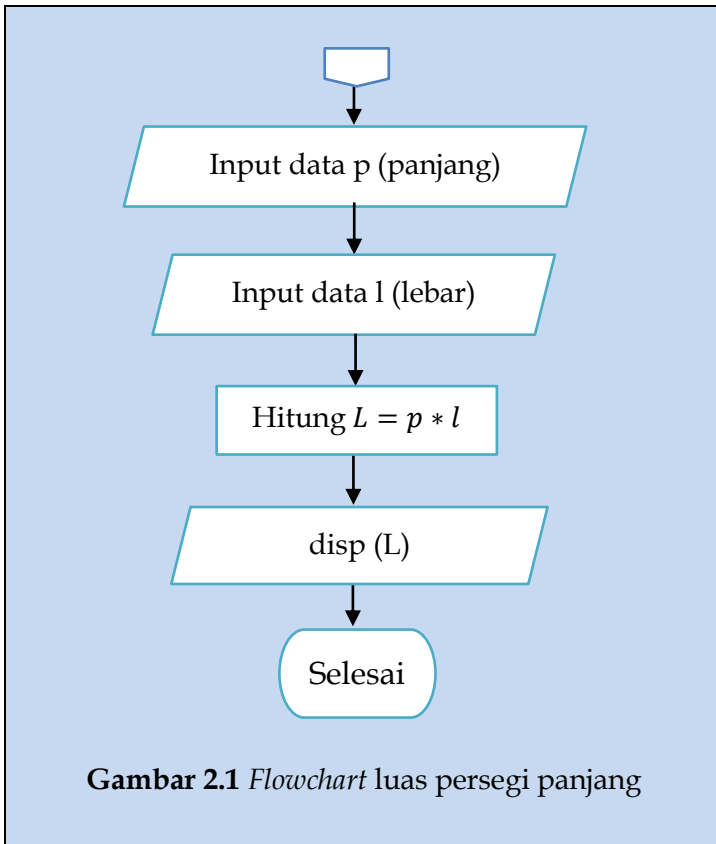
Algoritma untuk menentukan luas persegi panjang,

Algoritma dalam struktur bahasa :

- Step 1. Mulai
- Step 2. Masukkan data panjang
- Step 3. Masukkan data lebar
- Step 4. Hitung luas = panjang * lebar
- Step 5. Tampilkan luas
- Step 6. Selesai

Algoritma dalam bentuk *flowchart* :





B. Struktur Dasar Pemilihan (*Selection*)

Struktur Pemilihan adalah struktur yang digunakan pada program yang memerlukan proses pengujian kondisi untuk mengambil suatu keputusan apakah suatu baris perintah akan diproses atau tidak. Pengujian kondisi ini dilakukan untuk memilih salah satu dari beberapa alternatif yang tersedia. Tidak semua baris program akan dikerjakan pada struktur ini, melainkan hanya baris yang memenuhi syarat saja.

Secara umum, perintah dalam struktur ini berjalan secara runtun atau berururut mulai dari perintah pertama hingga akhir, namun perintah dapat

dibuat berpindah ke perintah lain atau berhenti jika syarat yang ditentukan terpenuhi.

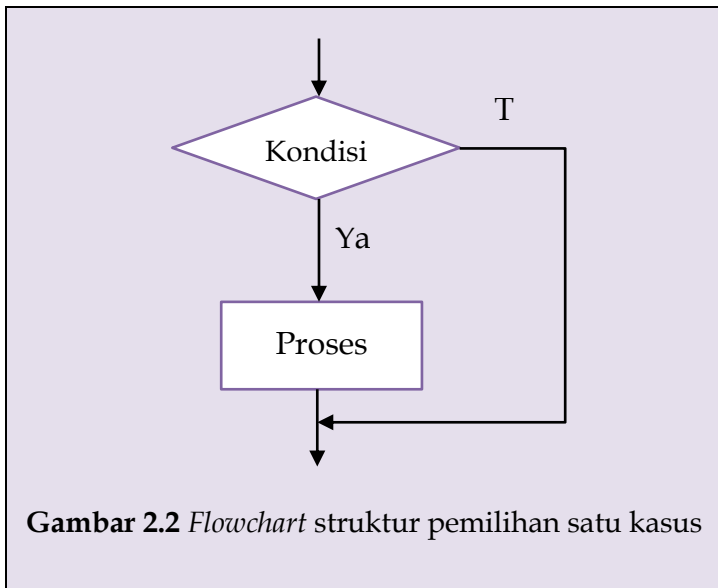
Dalam implementasi MATLAB, beberapa intruksi atau perintah yang dapat digunakan untuk membuat program dengan struktur pemilihan, antara lain perintah

1. `if..then..end`
2. `if..then..else..end`
3. `if..then..elseif..else..end`
4. `switch..case..otherwise`

Perintah-perintah ini digunakan sesuai dengan jenis struktur pemilihan yang akan dibuat.

1. Struktur Pemilihan Satu Kasus

Perintah `if..then..end` digunakan jika struktur pemilihan hanya memuat satu pilihan. Bentuk ini juga sering disebut dengan istilah struktur pemilihan satu kasus. Bentuk umum struktur pemilihan satu kasus dinyatakan sebagai berikut,



Gambar 2.2 Flowchart struktur pemilihan satu kasus

Dalam hal ini proses akan dijalankan jika memenuhi kondisi yang ditetapkan.

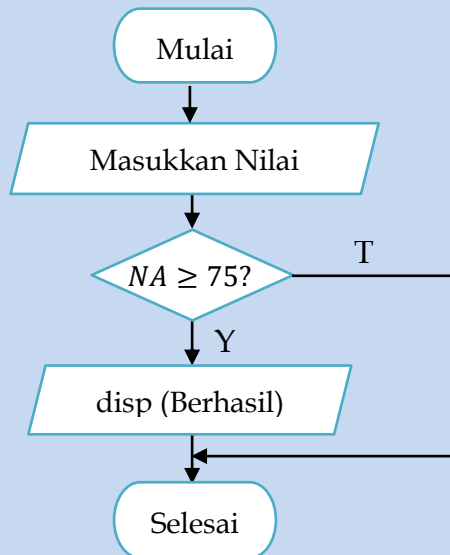
CONTOH 2.2

Algoritma untuk menentukan kriteria kelulusan dengan syarat Nilai Akhir (NA) lebih besar dari 75:

Algoritma dalam struktur bahasa :

- Step 1. Mulai
- Step 2. Masukkan data Nilai Akhir (NA)
- Step 3. Baca Nilai Akhir
- Step 4. Periksa $NA > 75$? jika benar, maka
- Step 5. Tulis Berhasil
- Step 6. Akhiri program

Algoritma dalam bentuk *flowchart* :



Gambar 2.3 *Flowchart* kriteria kelulusan dengan satu pilihan

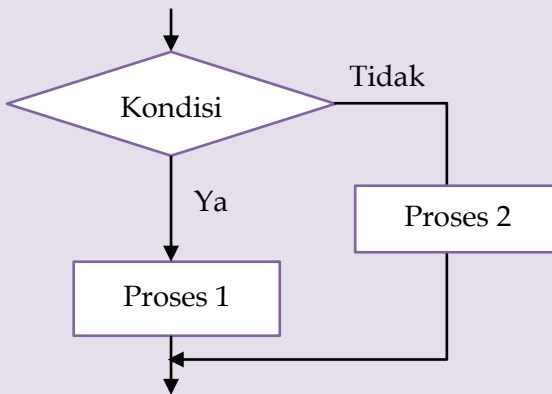
2. Struktur Pemilihan Dua Kasus

Struktur pemilihan dua kasus menggunakan perintah `if..then..else..end`, artinya perintah ini digunakan jika kasus yang dihadapi memuat 2 pilihan.

Bentuk umum struktur pemilihan dua kasus dinyatakan sebagai berikut,

```
if (kondisi)
then proses 1
else proses 2
end
```

Dalam hal ini proses akan dijalankan jika memenuhi kondisi yang ditetapkan. Dalam bentuk *flowchart* dinyatakan pada Gambar 2.4 berikut.



Gambar 2.4 Flowchart struktur pemilihan dua kasus

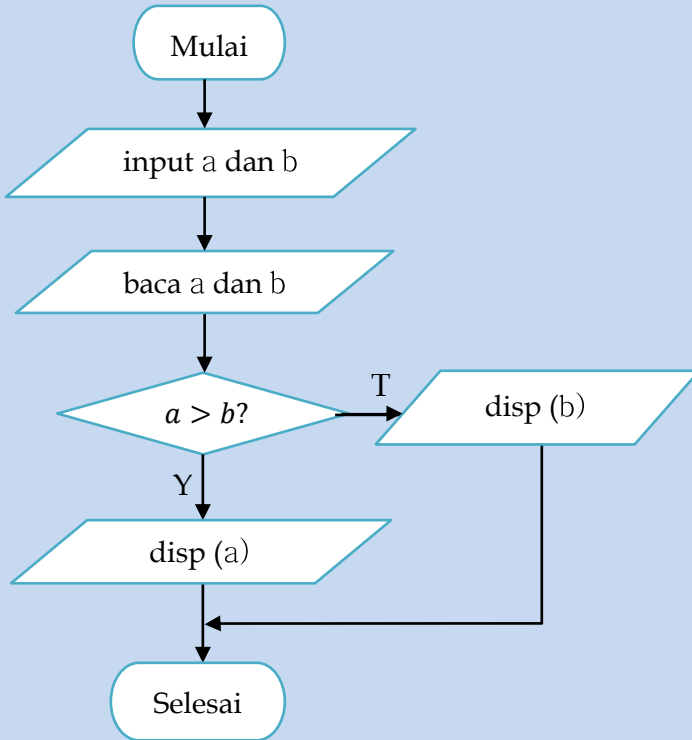
CONTOH 2.3

Algoritma untuk untuk menentukan bilangan yang lebih besar diantara dua buah bilangan:

Algoritma dalam struktur bahasa :

- Step 1. Mulai
- Step 2. Masukkan data a dan b
- Step 3. Baca nilai dari a dan b
- Step 4. Periksa apakah $a > b$, jika benar, maka
 - Step 5. Tulis a
- Step 6. Kondisi lainnya yakni ingkaran dari step 4 yaitu $b > a$
 - Step 7. Tulis b
- Step 8. Akhiri program

Algoritma dalam bentuk *flowchart* :



Gambar 2.5 *Flowchart* perbandingan dua bilangan

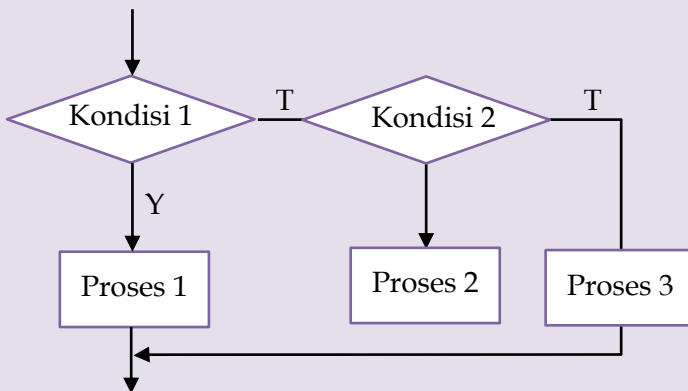
3. Struktur Pemilihan Tiga Kasus atau Lebih

Struktur pemilihan dengan tiga kasus atau lebih menggunakan perintah `if .. then .. elseif .. then .. else .. end`, artinya perintah ini digunakan jika masalah yang dihadapi memuat tiga pilihan atau lebih.

Bentuk umum struktur pemilihan tiga kasus atau lebih dinyatakan sebagai berikut,

```
if (kondisi 1)
then proses 1
elseif (kondisi 2)
then proses 2
else proses 3
end
```

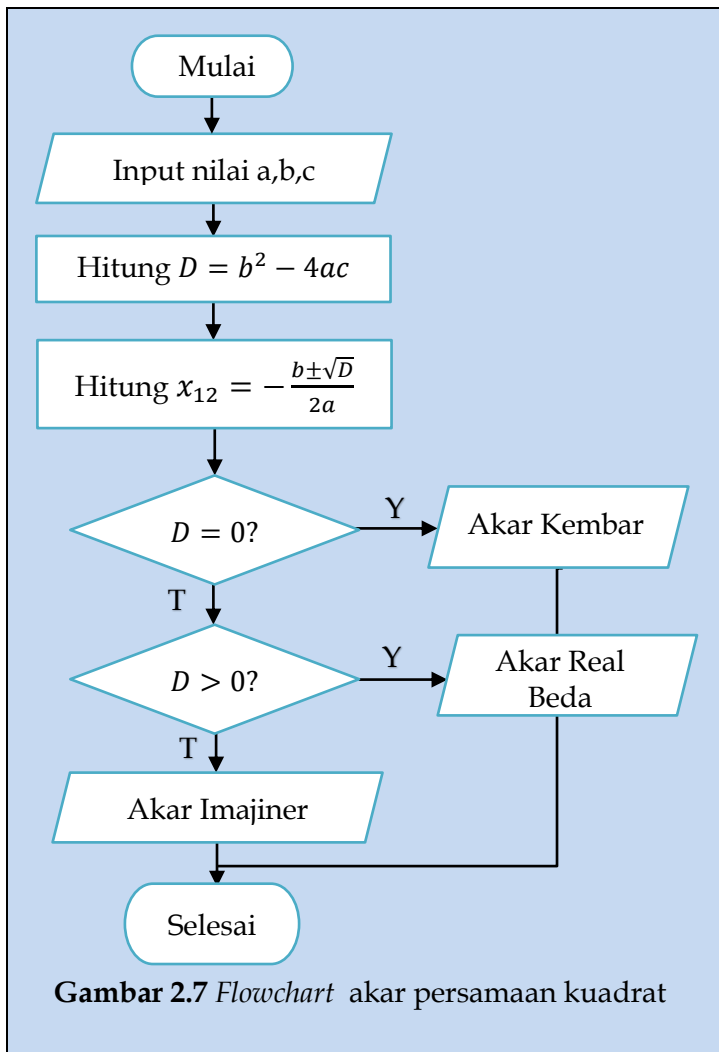
Dalam hal ini proses akan dijalankan jika memenuhi kondisi yang ditetapkan. Dalam bentuk *flowchart* dinyatakan pada Gambar 2.6 berikut,



Gambar 2.6 *Flowchart* struktur pemilihan dua kasus

CONTOH 2.4

Algoritma penentuan akar persamaan kuadrat dengan tiga kasus,



Struktur algoritma yang dinyatakan dalam bentuk *flowchart* pada Gambar 2.7 tersebut bersesuaian dengan algoritma dengan struktur bahasa berikut:

1. Mulai
2. Tentukan nilai a, b, c
3. Hitung nilai dari $D = b^2 - 4ac$
4. Hitung $x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$
5. Periksa apakah $D = 0$, jika benar, maka
 6. Tampilkan “Akar Real Kembar”
7. Jika tidak, periksa apakah $D > 0$? Jika benar,
 8. Tampilkan “Akar Real Beda”
9. Kondisi lainnya ($D < 0$)
 10. Tampilkan “Akar Imajiner”
11. Akhiri program

4. Struktur Pemilihan dengan Perintah `switch..case..otherwise`

Perintah `switch..case..otherwise` dapat digunakan sebagai alternatif perintah selain `if..elseif..else..end`. Lebih khusus, perintah ini digunakan untuk memilih suatu proses dari beberapa kemungkinan proses berdasarkan nilai dari kondisi variabel kontrol. Kondisi pada perintah `switch..case..otherwise` ditentukan oleh nilai variabel yang menjadi pilihan, artinya program akan mengeksekusi suatu proses berdasarkan nilai variabel pilihan.

Bentuk umum struktur pemilihan dengan perintah switch .. case .. otherwise dinyatakan sebagai berikut,

```
switch kondisi
    case kondisi 1
        proses 1
    case {kondisi 2, kondisi 3, ...}
        proses 2/proses 3/...
    case {...}
    ...
    otherwise
        proses n
end
```

Dalam hal ini pencocokan nilai ekspresi switch dengan nilai ekspresi case dilakukan secara bertingkat dimulai dari yang paling atas. Misalnya, jika nilai kondisi 1 terpenuhi, maka hanya proses 1 yang akan dijalankan. Jika tidak cocok, maka proses akan dilakukan pada case selanjutnya. Apabila tak satupun ekspresi case yang cocok dengan ekspresi switch, maka pernyataan otherwise yang akan dijalankan.

Hal menarik pada statement switch adalah bagian ekspresi case dapat melibatkan lebih dari satu ekspresi yang di buat dalam tanda { } dengan menggunakan tanda koma sebagai pemisah antar ekspresi.

CONTOH 2.5

Algoritma penentuan gaji karyawan,

```
1. Mulai
2. Masukkan data
   input (nama)
   input (golongan)
3. Proses
   case golongan
     case 1
       disp (gaji golongan 1)
     case 2
       disp (gaji golongan 2)
     case 3
       disp (gaji golongan 3)
     case 4
       disp (gaji golongan 4)
     otherwise
       disp (hanya tersedia golongan 1 s.d 4)
   end case
4. Output Data
   write (nama)
   write (golongan)
   write (gaji)
5. Akhiri program
   end
```

C. Struktur Dasar Pengulangan (*Repetition*)

Struktur Pengulangan adalah struktur yang melakukan pengulangan beberapa kali terhadap satu baris atau satu blok baris program. Pengulangan akan dilakukan sesuai dengan persyaratan yang diberikan.

Beberapa intruksi atau perintah yang dapat digunakan untuk membuat program dengan struktur pengulangan, antara lain perintah `for`, `while` atau `do..while`. Perintah ini digunakan sesuai dengan jenis struktur pengulangan yang akan dibuat.

1. Struktur Pengulangan dengan Perintah `for`

Struktur pengulangan dengan perintah `for` digunakan untuk mengulang suatu proses dengan variabel kendali. Variabel kendali yang dimaksud memuat nilai awal dan nilai akhir. Dengan nilai awal dan nilai akhir ini, satu baris perintah akan berulang sampai jumlah perulangan yang disyaratkan terpenuhi.

Berikut karakteristik pengulangan dengan perintah `for` :

- ✓ Terdapat nilai awal dan nilai akhir yang menunjukkan banyaknya pengulangan yang akan dilakukan.
- ✓ Proses pengulangan akan berhenti jika nilai variabel kendali mencapai batas nilai akhir.
- ✓ Nilai variabel kendali berubah secara teratur dengan interval tertentu.
- ✓ Jumlah pengulangan dapat diketahui.

Bentuk umum dan *flowchart* pada Gambar 2.8 berikut mendeskripsikan bagaimana struktur pengulangan dengan perintah `for` dijalankan. Pada *flowchart* ini dijelaskan bahwa perintah 1, perintah 2, dan perintah 3 akan dikerjakan secara berulang dimulai dari nilai_awal dan berakhir pada nilai_akhir yang diberikan. Pengulangan akan berhenti apabila pengulangan sudah sampai pada kondisi nilai_akhir yang diberikan.

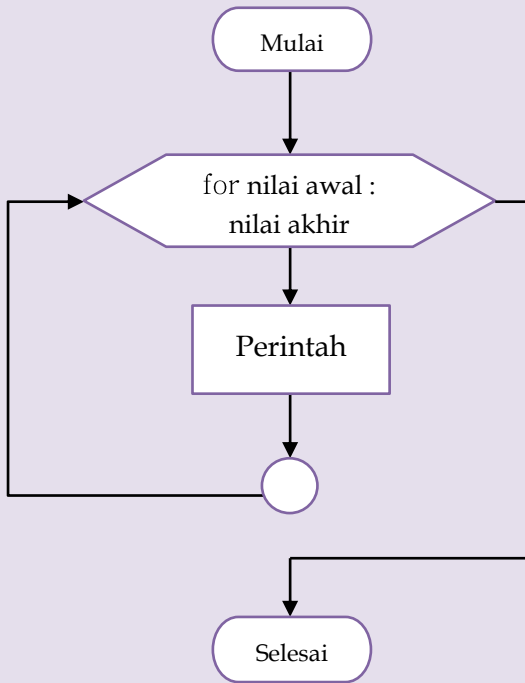
Bentuk umum pengulangan for :

```

for var = nilai awal : step : nilai akhir
...
perintah-perintah
...
end

```

Dalam bentuk *flowchart* dinyatakan sebagai berikut,



Gambar 2.8 *Flowchart* struktur pengulangan dengan perintah for

Keterangan:

- ✓ Variabel perulangan (var) harus bertipe dasar: integer, real, atau char.
- ✓ Nilai awal < nilai akhir jika step positif.

- ✓ Nilai awal > nilai akhir jika step negatif.
- ✓ Variabel dengan nilai awal bertambah sebesar 1 step setiap satu kali putaran.
- ✓ Pengulangan akan berhenti setelah nilai variabel mencapai batas akhir atau nilai akhir.

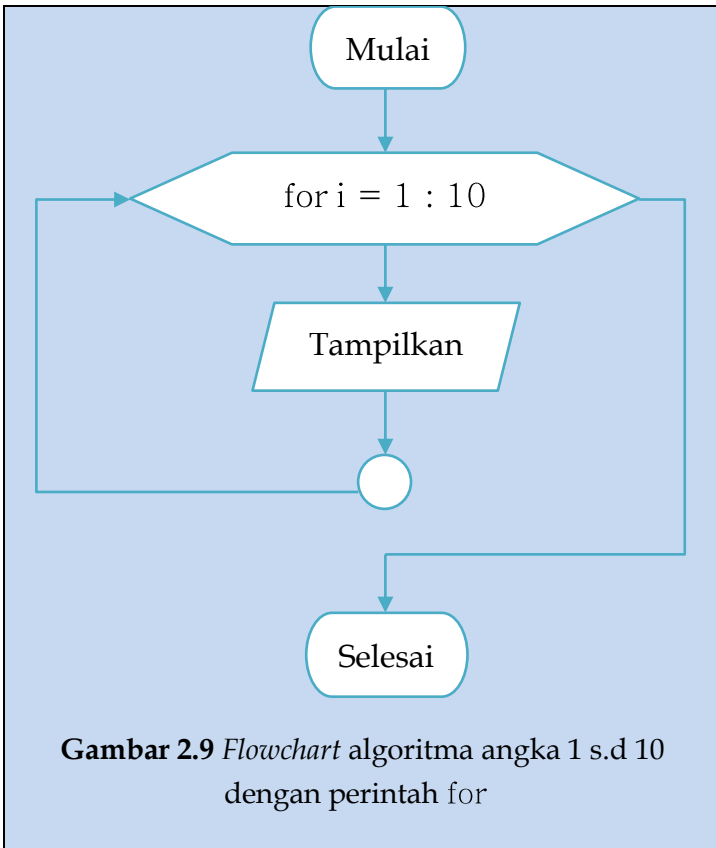
CONTOH 2.6

Algoritma untuk menampilkan angka 1 sampai 10,

Algoritma dalam struktur bahasa :

1. Mulai
2. Proses
for i = 1 : 10
3. Output
disp (i)
4. Selesai

Dalam bentuk *flowchart* :



2. Struktur Pengulangan dengan Perintah while

Struktur pengulangan dengan perintah while digunakan untuk mengulang satu baris perintah selama syarat yang diberikan masih terpenuhi. Perintah while bekerja dengan cara melakukan pengecekan nilai kondisi benar atau salah.

Berikut karakteristik pengulangan dengan perintah while :

- ✓ Syarat akan uji terlebih dahulu sebelum perintah pengulangan dikerjakan, sehingga ada kemungkinan baris perintah yang akan diulang tidak dikerjakan sama sekali.
- ✓ Proses pengulangan terjadi jika syarat terpenuhi dan berhenti jika syarat tidak terpenuhi.

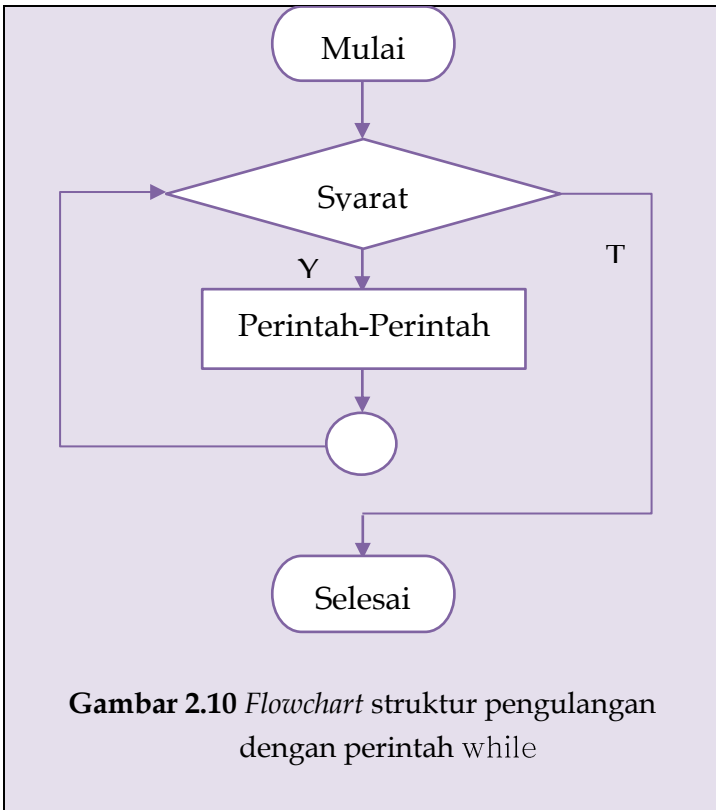
- ✓ Nilai variabel kondisi dikendalikan pada proses pengulangan.
- ✓ Jumlah pengulangan tidak penting untuk diketahui.

Bentuk umum dan *flowchart* pada Gambar 2.10 berikut mendeskripsikan bagaimana struktur pengulangan dengan perintah `while` dijalankan.

Bentuk umum struktur pengulangan dengan perintah `while` :

```
while kondisi
    ...
    perintah-perintah
    ...
end
```

Dalam bentuk *flowchat* dinyatakan sebagai berikut,



Pada *flowchart* ini dijelaskan bahwa syarat akan terlebih dahulu diuji sebelum masuk pada blok yang diulang. Jika syarat yang diuji bernilai benar maka perintah akan dikerjakan, kemudian syarat akan diuji kembali. Jika syarat yang diuji bernilai benar maka perintah akan dikerjakan lagi. Pengulangan akan terus berlanjut sampai syarat yang diuji bernilai salah (tidak terpenuhi).

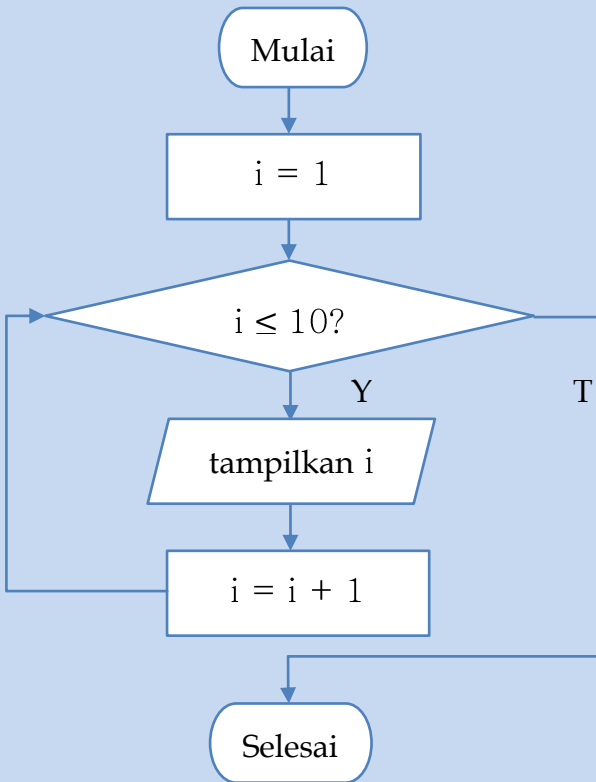
CONTOH 2.7

Algoritma untuk menampilkan angka 1 sampai 10 dengan perintah while,

Algoritma dalam struktur kalimat :

1. Mulai
2. Identifikasi nilai awal
 $i = 1$
3. Proses
selama $i \leq 10$, kerjakan langkah 4 dan 5
4. disp (i)
5. $i = i + 1$
4. Selesai

Dalam bentuk *flowchart* :

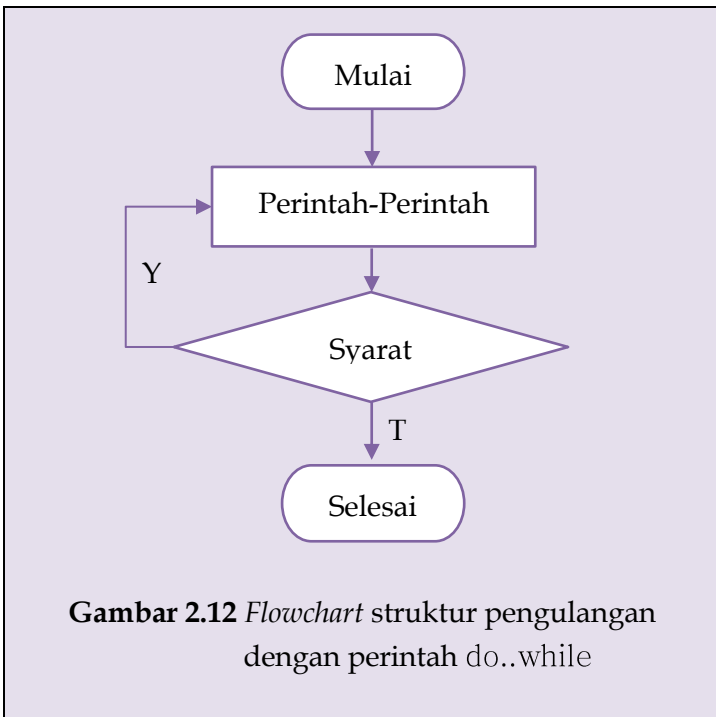


Gambar 2.11 Flowchart algoritma angka 1 s.d 10 dengan perintah while

3. Struktur Pengulangan dengan Perintah do..while

Struktur pengulangan dengan perintah `do..while` digunakan untuk mengulang satu baris perintah sampai syarat tidak terpenuhi. Ciri utama pengulangan `do..while` adalah syarat akan uji setelah perintah yang akan diulang dikerjakan, sehingga akan ada minimal satu kali baris perintah yang dikerjakan masuk dalam blok `do..while`.

Flowchart pada Gambar 2.12 berikut mendeskripsikan bagaimana struktur pengulangan dengan perintah `do..while` dijalankan.

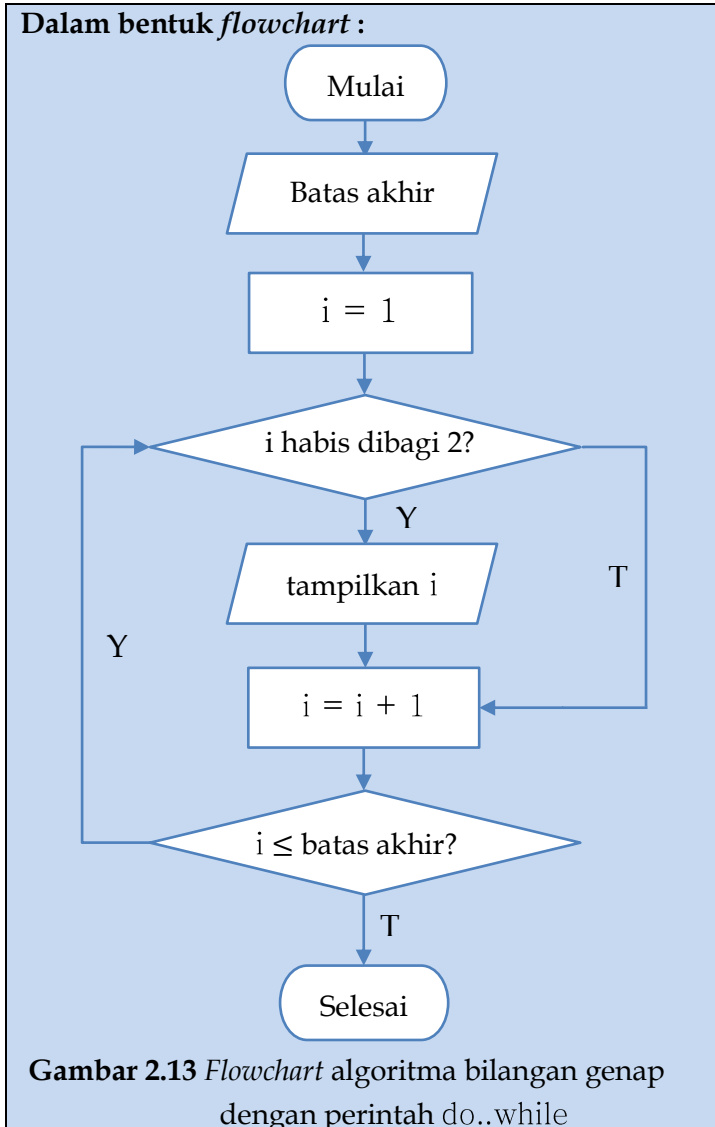


Pada *flowchart* ini dijelaskan bahwa syarat diuji setelah perintah dijalankan. Jika syarat yang diuji bernilai benar maka pengulangan perintah akan dikerjakan, kemudian syarat akan diuji kembali.

Pengulangan akan terus berlanjut sampai syarat yang diuji bernilai salah (tidak terpenuhi).

CONTOH 2.8

Algoritma untuk menampilkan bilangan genap,



Algoritma pada **Gambar 2.13** diatas dapat dinyatakan dalam struktur kalimat sebagai berikut:

1. Mulai
2. Identifikasi batas akhir
3. Identifikasi nilai awal
 $i = 1$
3. Proses
Selama $i \leq$ batas akhir, kerjakan step 4, 5, 6
4. Jika i habis dibagi 2, maka kerjakan step 5
5. disp (i)
6. $i = i + 1$
7. Selesai

LATIHAN 2

Buatlah algoritma dalam bentuk kalimat dan *flowchart* dari kasus-kasus berikut:

1. Algoritma untuk menghitung besarnya gaji bersih jika diketahui : besarnya gaji pokok, tunjangan, uang lebur, pajak, dan potongan kredit.
2. Sifat trikotomi bilangan real menyatakan bahwa, jika X dan Y adalah sembarang bilangan real, maka terdapat 3 kemungkinan hubungan yang terjadi antara X dan Y , yaitu $X=Y$, $X > Y$ atau $X < Y$. Buatlah algoritma untuk menyimpulkan hubungan X dan Y .
3. Buatlah algoritma untuk menentukan nilai terbesar diantara tiga buah bilangan.
4. Algoritma untuk mencetak angka 1 sampai 10 dengan instruksi `do..while`.

5. Algoritma konversi nilai angka menjadi huruf dengan ketentuan:

No	Nilai Angka	Nilai Huruf
1	90-100	A (Sangat Baik)
2	80-89	B (Baik)
3	70-79	C (Cukup Baik)
4	0-69	D (Tidak Baik)

DAFTAR REFERENSI

- Anonim. (2018, September). *Perbedaan Hardware, Software, dan Brainware*. Retrieved Januari 2019, from www.indoworx.com: <https://www.indoworx.com/perbedaan-hardware-software-dan-brainware/>
- Attaway, S. (2009). *MATLAB : A Practical Introduction to Programming and Problem Solving*. Boston: Boston University.
- Brassard, G., & Bratley, P. (1988). *Algorithm, Theory and Practice*. New Jersey: Prentice Hall.
- Brian, R., Hunt, R., Lipsman, J., & Rosenberg. (2001). *A Guide to MATLAB for Beginners and Experienced Users*. Cambridge: Cambridge University Press.
- Chabert, J. (1999). *A History of Algorithm*. Berlin: Springer-Verlag Heidelberg.
- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2002). *Intoduction to Algorithms*. London: The MIT Press.
- Gerritsen, M. (2006). *A Brief Introduction to MATLAB*. CME200 Handout.
- Munir, R. (2007). *Algoritma dan Pemrograman, Jilid I*. Bandung: Informatika.
- Praptomo, Y. (-----). *Algoritma dan Pemrograman*. Yogyakarta: STIMIK El Rahma.

- Sahyar. (2016). *Algoritma dan Pemrograman Menggunakan MATLAB*. Medan: Kencana.
- Suarga. (2004). *Algoritma Pemrograman*. Yogyakarta: Penerbit Andi.
- Suryadi, H., & Sumin, A. (1997). *Algoritma dan Pemrograman*. Jakarta: Gunadarma.
- Sutedjo, B., & Michael, A. (2000). *Algoritma dan Teknik Pemrograman*. Yogyakarta: Penerbit Andi.
- Xenophontos, C. (1999). *A Beginners Guide to MATLAB*. New York: Clarkson University.
- Yuhendra. (2013). *Algoritma dan Pemrograman dalam Bahasa Pascal*. Padang: Institut Teknologi Padang.