

DASAR-DASAR PEMROGRAMAN

“Pengantar Algoritma dan Pemrograman”



Indra Hermawan, S.Kom, M.Kom

indrah13@gmail.com / indra@nurulfikri.ac.id

No. 085217987034

Tujuan Pembelajaran

Dalam pertemuan ini mahasiswa mampu:

1. Menjelaskan pengertian algoritma
2. Menjelaskan ciri-ciri suatu algoritma
3. Menyebutkan dan menjelaskan proses-proses yang ada pada suatu algoritma
4. Menyebutkan dan menjelaskan cara menyajikan suatu algoritma

- Bahasa Pemrograman
- Definisi Algoritma
- Ciri Algoritma
- Jenis Proses dalam Algoritma
- Notasi Algoritma
- Penerapan Algoritma

- Bahasa atau dalam bahasa inggris *language* adalah suatu sistim untuk berkomunikasi.
- Bahasa tertulis menggunakan simbol (yaitu huruf) untuk membentuk kata.
- Dalam ilmu komputer, bahasa manusia disebut bahasa alamiah (natural languages), dimana komputer tidak bisa memahaminya, sehingga diperlukan suatu bahasa komputer.
- Komputer mengerjakan transformasi data berdasarkan kumpulan perintah - program - yang telah dibuat oleh pemrogram.

Bahasa Pemrograman

- Kumpulan perintah ini harus dimengerti oleh komputer, berstruktur tertentu (syntax) dan bermakna.
- Bahasa pemrograman merupakan notasi untuk memberikan secara tepat program komputer.
- Berbeda dengan bahasa alamiah, mis. Bahasa Indonesia, Inggris dsb. yang merupakan bahasa alamiah(natural language), sintaks dan semantik bahasa pemrograman (komputer) ditentukan secara kaku, sehingga bahasa pemrograman juga disebut sebagai bahasa formal (formal language).

- Jadi, dalam bahasa pemrograman yang digunakan sebagai alat komunikasi untuk memberikan perintah kepada komputer, tidak berlaku kebebasan berekspresi seperti lainnya dalam bahasa alamiah.
- Dalam pengertian yang lebih sempit, pemrograman merupakan **pengkodean (coding atau program writing = penulisan program) dan pengujiannya (testing)** berdasarkan rancangan tertentu.
- Pemrograman dalam pengertian luas meliputi seluruh kegiatan yang tercakup dalam pembuatan **program, termasuk analisis kebutuhan (requirement's analysis) dan keseluruhan tahapan dalam perencanaan (planning), perancangan (design) dan pewujudannya (implementation).**

Bahasa Pemrograman

JENIS PROGRAM	BAHASA TERBAIK	BAHASA TERBURUK
Data terstruktur	ADA, C /C++, PASCAL	Assembler, BASIC
Proyek cepat	BASIC	PASCAL, ADA, Assembler
Eksekusi cepat	Assembler, C	BASIC, Interpreter Language
Kalkulasi matematika	FORTTRAN	PASCAL
Menggunakan memori dinamis	PASCAL, C	BASIC
Lingkungan memori terbatas	BASIC, Assembler, C	FORTTRAN
Program real-time	ADA, Assembler, C	BASIC, FORTTRAN
Manipulasi string	BASIC, PASCAL	C
Program mudah dikelola	PASCAL, ADA	C, FORTTRAN

- Pemrograman komputer dan algoritma pemrograman adalah dua hal yang tidak dapat dipisahkan karena pembuatan program komputer akan lebih sulit dan lama tanpa mengetahui dengan pasti bagaimana algoritma penyelesaian masalahnya.
- Definisi program komputer adalah sederetan perintah-perintah (instruksi) yang harus dikerjakan oleh komputer untuk menyelesaikan masalah.
- menurut Microsoft Press Computer and Internet Dictionary (1998)
 - “urutan langkah logis tertentu untuk memecahkan suatu masalah”.

- Yang ditekankan adalah urutan **langkah logis**, yang berarti algoritma harus mengikuti suatu urutan tertentu, tidak boleh melompat-lompat dan disusun secara sistematis.
- Sedangkan yang dimaksud dengan **langkah-langkah logis** adalah kita harus dapat mengetahui dengan pasti setiap langkah yang kita buat.
- Menurut Sjukani (2005),
 - “algoritma adalah alur pemikiran dalam menyelesaikan suatu pekerjaan yang dituangkan secara tertulis”.

Algoritma

- Yang ditekankan pertama adalah alur pikiran, sehingga algoritma seseorang dapat berbeda dari algoritma orang lain.
- Sedangkan penekanan kedua adalah tertulis, yang artinya dapat berupa kalimat, gambar, atau tabel tertentu.
- dapat disimpulkan bahwa algoritma lebih merupakan alur pemikiran untuk menyelesaikan suatu pekerjaan atau suatu masalah daripada pembuatan program komputer.

Ciri Algoritma

- Input: algoritma dapat memiliki nol atau lebih inputan dari luar.
- Output: algoritma harus memiliki minimal satu buah output keluaran.
- *Definiteness (pasti): algoritma memiliki instruksi-instruksi yang jelas dan tidak ambigu.*
- *Finiteness (ada batas): algoritma harus memiliki titik berhenti (stopping role).*
- *Effectiveness (tepat dan efisien): algoritma sebisa mungkin harus dapat dilaksanakan dan efektif. Contoh instruksi yang tidak efektif adalah: $A = A + 0$ atau $A = A * 1$*

- Empat jenis proses yang dikenal dalam algoritma:
 1. *Sequence Process*, merupakan instruksi yang dikerjakan secara sekuensial (berurutan).
 2. *Selection Process*, adalah instruksi yang dikerjakan jika memenuhi suatu kriteria atau kondisi (keadaan) tertentu.
 3. *Iteration Process*, yaitu instruksi yang dikerjakan selama memenuhi suatu kriteria atau kondisi (keadaan) tertentu.
 4. *Concurrent Process*, beberapa instruksi dikerjakan secara bersama.

Menyajikan Algoritama

- Algoritma mempunyai aturan penulisan sendiri yang disebut dengan notasi algoritma.
- Notasi algoritma merupakan bahasa universal yang dapat diterima oleh semua bahasa pemrograman yang ada. Oleh sebab itu algoritma yang baik harus dapat diterjemahkan ke dalam bentuk source code dari semua bahasa pemrograman yang ada.
- Terdapat tiga buah notasi algoritma yang dikenal yaitu,
 - uraian kalimat deskriptif,
 - *flow chart*,
 - *pseudo code*.

- Algoritma dengan uraian kalimat deskriptif adalah notasi algoritma yang paling sederhana karena algoritma ini menggunakan bahasa sehari-hari.

1. Masukkan sembarang bilangan sebanyak tiga buah.
2. Ambil bilangan pertama dan set maksimum-nya sama dengan bilangan pertama.
3. Ambil bilangan kedua dan bandingkan dengan maksimum.
4. Apabila bilangan kedua lebih besar dari maksimum maka ubah maksimum-nya menjadi sama dengan bilangan kedua.
5. Ambil bilangan ketiga dan bandingkan dengan maksimum.
6. Apabila bilangan ketiga lebih besar dari maksimum maka ubah lagi maksimum-nya menjadi sama dengan bilangan ketiga.
7. Variabel maksimum akan berisi bilangan yang terbesar dan tampilkan hasilnya.

Contoh Penggunaan Deskriptif Algoritma

Menyajikan Algoritma (Deskriptif Algoritma)

1. Masukkan a, b, dan c.
2. $\text{mak} \leftarrow a$.
3. Jika $b > \text{mak}$, kerjakan langkah ke-4. Jika tidak, kerjakan langkah ke-5.
4. $\text{mak} \leftarrow b$.
5. Jika $c > \text{mak}$, kerjakan langkah ke-6. Jika tidak, kerjakan langkah ke-7.
6. $\text{mak} \leftarrow c$.
7. Tulis mak.

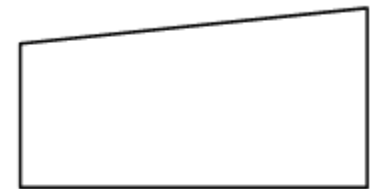
Contoh Penggunaan Deskriptif Algoritma yang lebih sederhana

Menyajikan Algoritama (Flow Chart)

- Notasi algoritma yang paling banyak digunakan adalah flow chart karena bentuknya yang sederhana dan mudah dipahami.
- Flow chart (diagram alir) adalah penggambaran secara grafik dari langkah-langkah pemecahan masalah yang harus diikuti oleh pemroses.
- Flow chart terdiri atas sekumpulan simbol, dimana masing-masing simbol menggambarkan suatu kegiatan tertentu.
- Flow chart diawali dengan penerimaan masukan (input), pemrosesan masukan, dan diakhiri dengan menampilkan hasilnya (output).

Menyajikan Algoritama (Flow Chart)

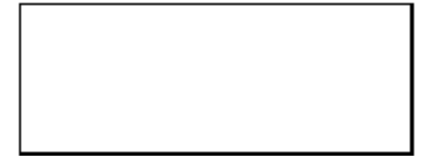
- Adapun simbol-simbol yang sering digunakan untuk menyusun flow chart (dalam microsoft visio) adalah sebagai berikut :
- Masukan
 - Masukan merupakan kegiatan penerimaan data yang disimbolkan dengan jajaran genjang.
- Masukan Manual
 - Untuk masukan secara manual yang dimasukkan melalui keyboard, atau perangkat input lainnya seperti barcode reader, kita dapat menggunakan simbol masukan secara manual.



Menyajikan Algoritama (Flow Chart)

- Proses

- Data yang dimasukan kemudian diproses untuk menghasilkan jawaban atas persoalan yang ingin dipecahkan. Masukan Manual



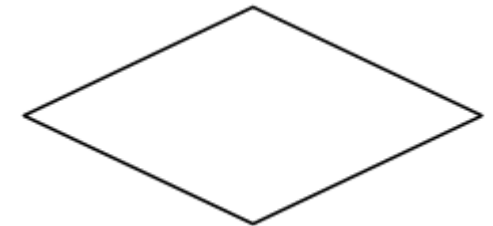
- Keluaran

- Keluaran adalah hasil dari pemrosesan data dan merupakan jawaban atas permasalahan yang ada.

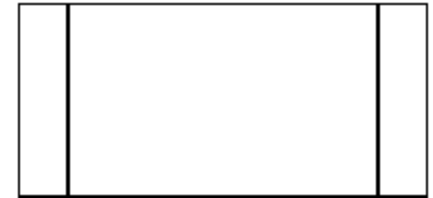


- Percabangan

- Yang dimaksud dengan percabangan disini adalah suatu kegiatan untuk mengecek atau memeriksa suatu keadaan apakah memenuhi suatu kondisi tertentu atau tidak.

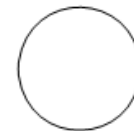
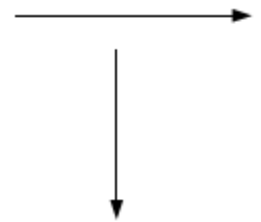


- Sub rutin
 - Sub rutin adalah suatu bagian dalam program yang dapat melakukan (atau diberi) tugas tertentu.
 - Jadi sub rutin merupakan “program kecil” yang menjadi bagian dari suatu program yang besar.
 - Sub rutin ada dua macam, yaitu prosedur (procedure) dan fungsi (function).
 - Perbedaan antara keduanya adalah setelah dipanggil prosedur tidak mengembalikan suatu nilai sedangkan fungsi selalumengembalikan suatu nilai.

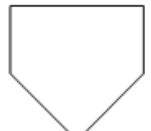


Menyajikan Algoritama (Flow Chart)

- Arah aliran
 - Arah aliran merupakan jalur yang harus diikuti dan merupakan garis penghubung yang menghubungkan setiap langkah pemecahan masalah yang ada dalam flow chart.
- Terminator
 - Terminator berfungsi untuk menandai titik awal dan titik akhir dari suatu flow chart.
- Konektor
 - Konektor berfungsi untuk menghubungkan suatu langkah dengan langkah lain dalam sebuah flow chart dengan keadaan on page atau off page.



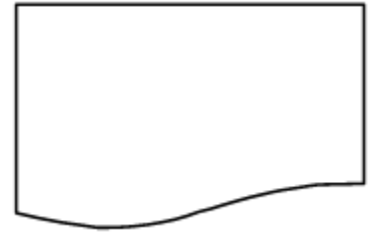
(a)



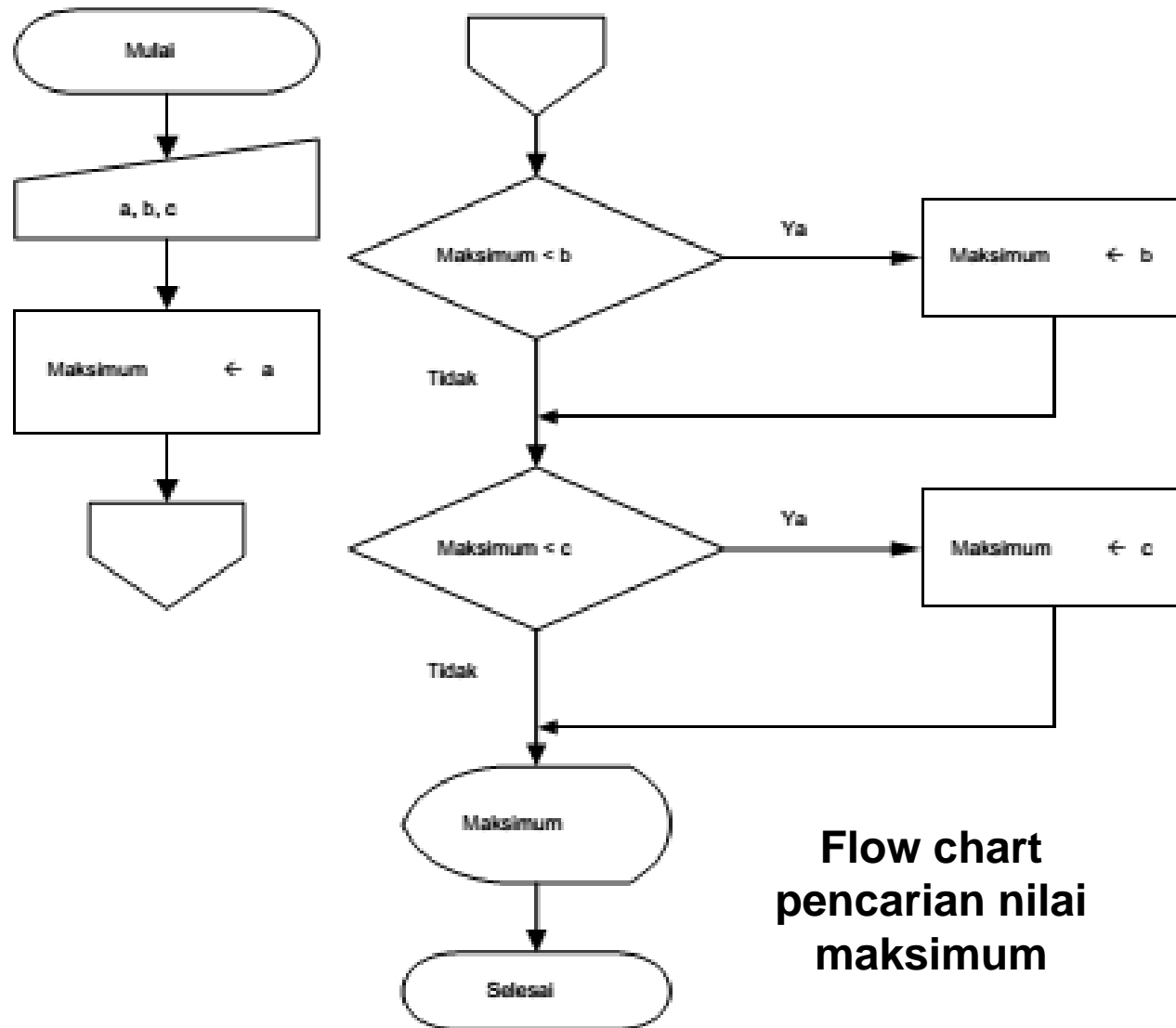
(b)

(a) On page
(b) off page

- Dokumen
 - Dokumen merupakan tampilan data secara fisik yang dapat dibaca oleh manusia. Data ini biasanya merupakan hasil pemecahan masalah (informasi) yang telah dicetak (print out).



Menyajikan Algoritama (Contoh Flow Chart)



Flow chart
pencarian nilai
maksimum

Menyajikan Algoritma (Psoudo code)

- Pseudo code adalah algoritma yang bentuknya (strukturnya) sangat mirip dengan bahasa pemrograman khususnya bahasa pemrograman terstruktur seperti pascal.
- Kemiripan ini merupakan keuntungan dari pseudo code karena implementasi atau penerjemahan algoritma ke dalam source code suatu bahasa pemrograman sangatlah mudah meskipun penggunaannya tidak sepopuler flow chart.

Menyajikan Algoritama (Psoudo code)

- Dalam penulisannya, pseudo code harus terdiri dari tiga bagian, yaitu :
 - Judul algoritma
 - Bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) dari algoritma tersebut.
 - Deklarasi
 - Bagian untuk mendefinisikan semua nama yang digunakan di dalam program.
 - Deskripsi
 - Bagian ini berisi uraian langkahlangkah penyelesaian masalah yang ditulis dengan menggunakan aturan-aturan yang akan dijelaskan selanjutnya.

Menyajikan Algoritama (Psoudo code)

{algoritma ini mencari bilangan terbesar dari tiga bilangan yang dimasukkan}
deklarasi

a,b,c,mak : integer

deskripsi

read(a,b,c)

mak \leftarrow a

if (mak<b)

mak \leftarrow b

else if(mak<c)

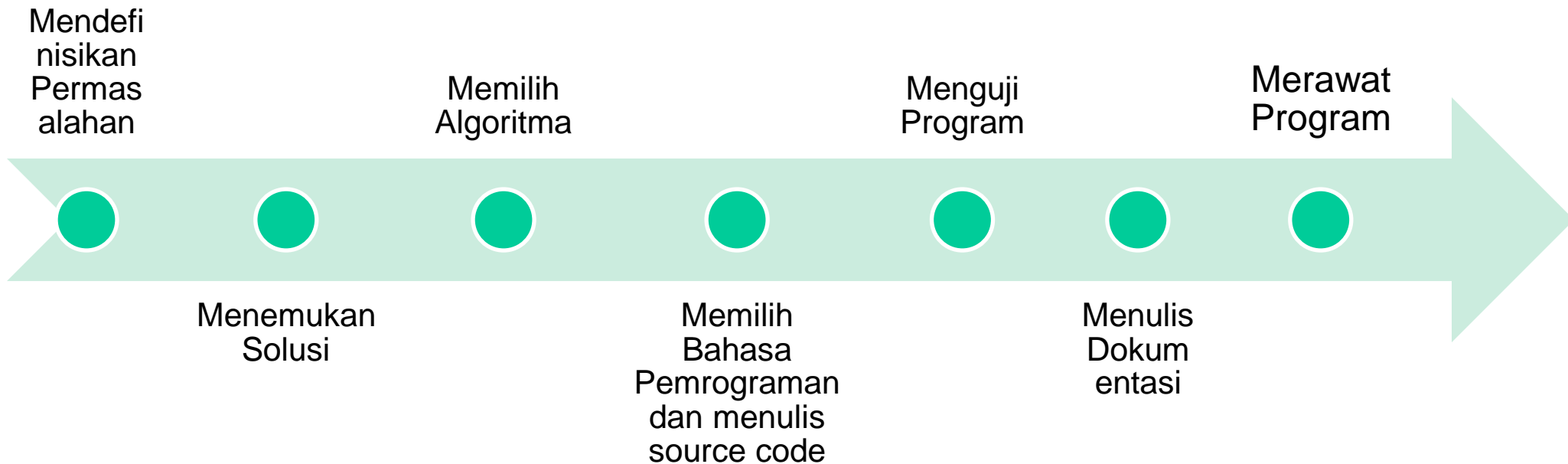
mak \leftarrow c

end if

write(mak)

Pseudo code pencarian nilai maksimum

Penerjemahan ke Kode Sumber



Tahapan pembuatan program komputer

Contoh permasalahan

1. Masalah penukaran isi ember (contoh sequence process).
 - Masalah penukaran isi ember dapat kita aplikasikan untuk masalah berikut ini. Mula-mula A bernilai 19 dan B bernilai 33.

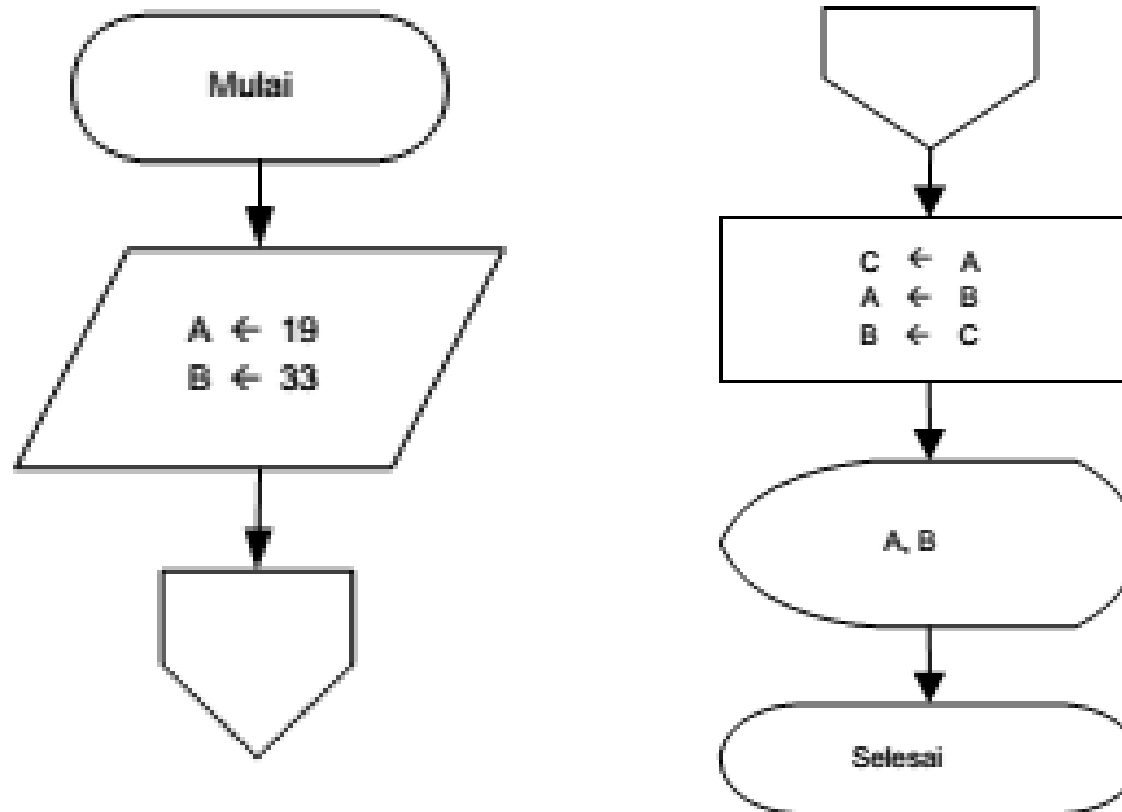
Menggunakan uraian kalimat deskriptif

- 1) Set nilai variabel A \leftarrow 19 dan B \leftarrow 33
- 2) Set nilai dari variabel C menjadi sama dengan variabel A.
- 3) Set nilai variabel A menjadi sama dengan B.
- 4) Set nilai variabel B menjadi sama dengan C.
- 5) Tampilkan nilai variabel A dan B.

Algoritma diatas dapat juga ditulis seperti dibawah ini:

- 1) A \leftarrow 19 dan B \leftarrow 33
- 2) C \leftarrow A
- 3) A \leftarrow B
- 4) B \leftarrow C
- 5) Tulis A dan B

Contoh permasalahan



Flow chart

Contoh permasalahan

Algoritma tukar_data

{algoritma ini digunakan untuk menukarkan dua buah data }

deklarasi

a,b,c : integer

deskripsi

a \leftarrow 19

b \leftarrow 33

c \leftarrow a

a \leftarrow b

b \leftarrow c

write(a,b)

Pseudo code

Contoh permasalahan

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     int a = 19, b = 33;
5.     int c = a;
6.     a = b;
7.     b = c;
8.     cout << "a = " << a << endl;
9.     cout << "b = " << b;
10.    return 0;
11.}
```

Source code

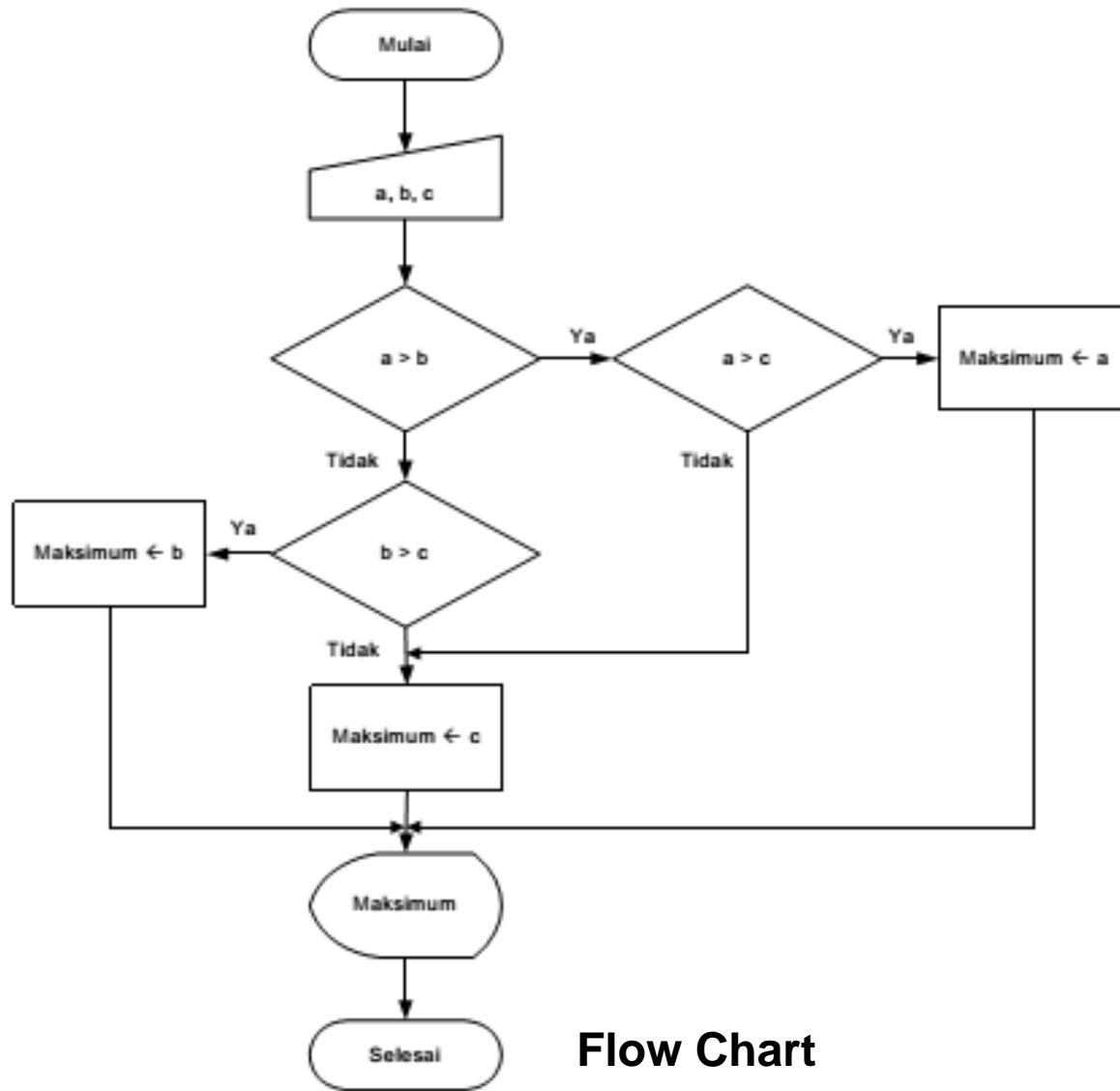
Contoh permasalahan

2. Masalah mencari bilangan terbesar dari tiga bilangan yang dimasukkan (contoh selection process).

- 1) Masukkan a, b, dan c.
- 2) Jika $a > b$, maka kerjakan langkah ke-3. Jika tidak, kerjakan langkah ke-5.
- 3) Jika $a > c$, maka kerjakan langkah ke-4. Jika tidak, kerjakan langkah ke-7.
- 4) $\text{mak} \leftarrow a$.
- 5) Jika $b > c$, kerjakan langkah ke-6. Jika tidak, kerjakan langkah ke-7.
- 6) $\text{mak} \leftarrow b$.
- 7) $\text{mak} \leftarrow c$.
- 8) Tulis mak.

Menggunakan kalimat deskriptif

Contoh permasalahan



Flow Chart

Contoh permasalahan

Algoritma bilangan_terbesar

{algoritma ini mencari bilangan terbesar dari tiga bilangan yang dimasukkan secara acak}

deklarasi

a,b,c,mak : integer

deskripsi

read(a,b,c)

if (a > b)

if (a > c)

mak \leftarrow a

else mak \leftarrow c

end if

else

if(b > c)

mak \leftarrow b

else mak \leftarrow c

end if

end if

write(mak

Pseudo code

Contoh permasalahan

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     int a,b,c,mak;
5.     cout << "Bilangan pertama = ";
6.     cin >> a;
7.     cout << "Bilangan kedua  = ";
8.     cin >> b;
9.     cout << "Bilangan ketiga = ";
10.    cin >> c;
11.    if (a > b) {
12.        if (a > c)
13.            mak = a;
14.        else mak = c;
15.    }
16.    else {
17.        if (b > c)
18.            mak = b;
19.        else mak = c;
20.    }
21.    cout << "Bilangan terbesarnya = " << mak;
22.    return 0;
23. }
```

Source code

Contoh permasalahan

3. Diminta suatu program dimana jika dimasukkan suatu bilangan (1 sampai dengan 7) maka akan menampilkan nama hari, yaitu :

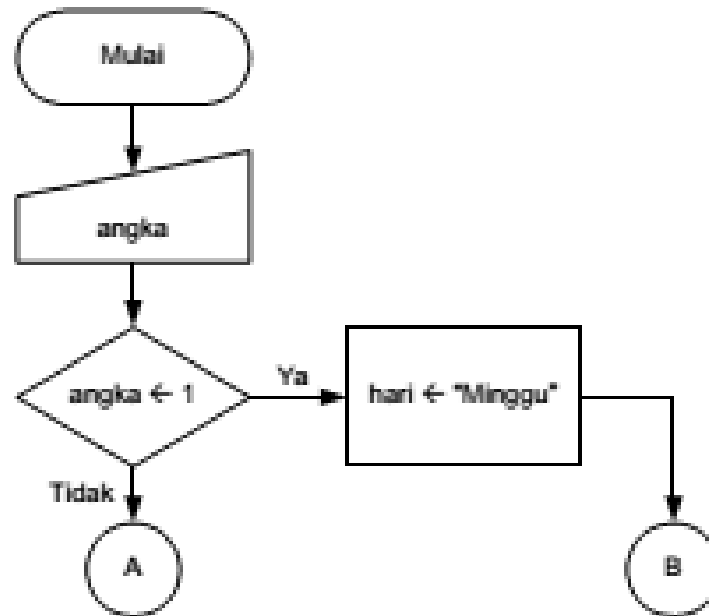
- Bilangan 1 untuk hari Minggu.
- Bilangan 2 untuk hari Senin.
- Bilangan 3 untuk hari Selasa.
- Bilangan 4 untuk hari Rabu.
- Bilangan 5 untuk hari Kamis.
- Bilangan 6 untuk hari Jumat.
- Bilangan 7 untuk hari Sabtu.

Contoh permasalahan

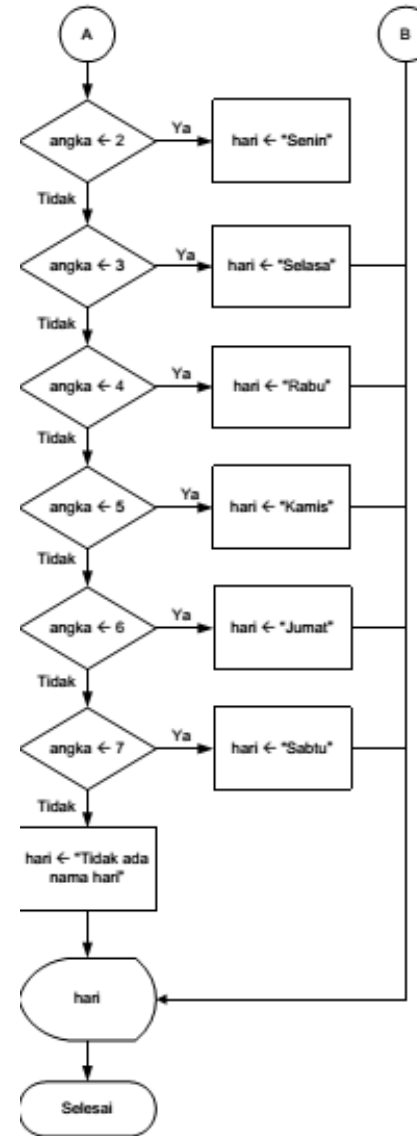
Menggunakan uraian kalimat deskriptif

- 1) Masukan bilangan sebuah bilangan bulat (angka).
- 2) Jika angka \leftarrow 1, maka kerjakan langkah ke-3. Jika tidak, kerjakan langkah ke-4.
- 3) hari \leftarrow "Minggu".
- 4) Jika angka \leftarrow 2, maka kerjakan langkah ke-5. Jika tidak, kerjakan langkah ke-6.
- 5) hari \leftarrow "Senin".
- 6) Jika angka \leftarrow 3, maka kerjakan langkah ke-7. Jika tidak, kerjakan langkah ke-8.
- 7) hari \leftarrow "Selasa".
- 8) Jika angka \leftarrow 4, maka kerjakan langkah ke-9. Jika tidak, kerjakan langkah ke-10.
- 9) hari \leftarrow "Rabu".
- 10) Jika angka \leftarrow 5, maka kerjakan langkah ke-11. Jika tidak, kerjakan langkah ke-12.
- 11) hari \leftarrow "Kamis".
- 12) Jika angka \leftarrow 6, maka kerjakan langkah ke-13. Jika tidak, kerjakan langkah ke-14.
- 13) hari \leftarrow "Jumat".
- 14) Jika angka \leftarrow 7, maka kerjakan langkah ke-15. Jika tidak, kerjakan langkah ke-16.
- 15) hari \leftarrow "Sabtu".
- 16) hari \leftarrow "Tidak ada nama hari untuk angka tersebut"
- 17) Tampilkan hari.

Contoh permasalahan



Flow chart



Contoh permasalahan

Algoritma nama_hari

{algoritma ini digunakan untuk mengetahui nama hari dari bilangan bulat yang dimasukkan}

deklarasi

angka : integer

hari : char

deskripsi

read(angka)

if(angka \leftarrow 1)

hari \leftarrow "Minggu"

else if(angka \leftarrow 2)

hari \leftarrow "Senin"

else if(angka \leftarrow 3)

hari \leftarrow "Selasa"

else if(angka \leftarrow 4)

hari \leftarrow "Rabu"

else if(angka \leftarrow 5)

hari \leftarrow "Kamis"

else if(angka \leftarrow 6)

hari \leftarrow "Jumat"

else if(angka \leftarrow 7)

hari \leftarrow "Sabtu"

else

hari \leftarrow "Tidak ada nama hari"

end if

write(hari)

Algoritma nama_hari

{algoritma ini digunakan untuk mengetahui nama hari dari bilangan bulat yang dimasukkan}

deklarasi

angka : integer

hari : char

deskripsi

read(angka)

switch(angka)

case 1 : hari \leftarrow "Minggu"

case 2 : hari \leftarrow "Senin"

case 3 : hari \leftarrow "Selasa"

case 4 : hari \leftarrow "Rabu"

case 5 : hari \leftarrow "Kamis"

case 6 : hari \leftarrow "Jumat"

case 7 : hari \leftarrow "Sabtu"

default : hari \leftarrow "Tidak ada nama hari"

end switch

write(hari)

Pseudo code

Contoh permasalahan

```
1. #include <iostream>
2. #include <cstring>
3. using namespace std;
4. int main() {
5.     int angka;
6.     string hari;
7.     cout << "Bilangan = ";
8.     cin >> angka;
9.     if (angka==1) hari="Minggu";
10.    else if (angka==2) hari="Senin";
11.    else if (angka==3) hari="Selasa";
12.    else if (angka==4) hari="Rabu";
13.    else if (angka==5) hari="Kamis";
14.    else if (angka==6) hari="Jumat";
15.    else if (angka==7) hari="Sabtu";
16.    else hari="tidak ada";
17.    cout << "Nama hari ke-" << angka << " adalah " << hari;
18.    return 0;
19. }
```

Source code

Contoh permasalahan

```
1. #include <iostream>
2. #include <cstring>
3. using namespace std;
4. int main() {
5.     int angka;
6.     string hari;
7.     cout << "Bilangan = ";
8.     cin >> angka;
9.     switch(angka) {
10.        case 1 : hari = "Minggu";
11.            break;
12.        case 2 : hari = "Senin";
13.            break;
14.        case 3 : hari = "Selasa";
15.            break;
16.        case 4 : hari = "Rabu";
17.            break;
18.        case 5 : hari = "Kamis";
19.            break;
20.        case 6 : hari = "Jumat";
21.            break;
22.        case 7 : hari = "Sabtu";
23.            break;
24.        default : hari = "tidak ada";
25.    }
26.    cout << "Nama hari ke-" << angka << " adalah = " << hari;
27.    return 0;
28. }
```

Source code

Contoh permasalahan

4. Masalah menghitung rata-rata dari sejumlah data yang dimasukkan secara manual melalui keyboard (contoh iteration process). Algoritmanya adalah:

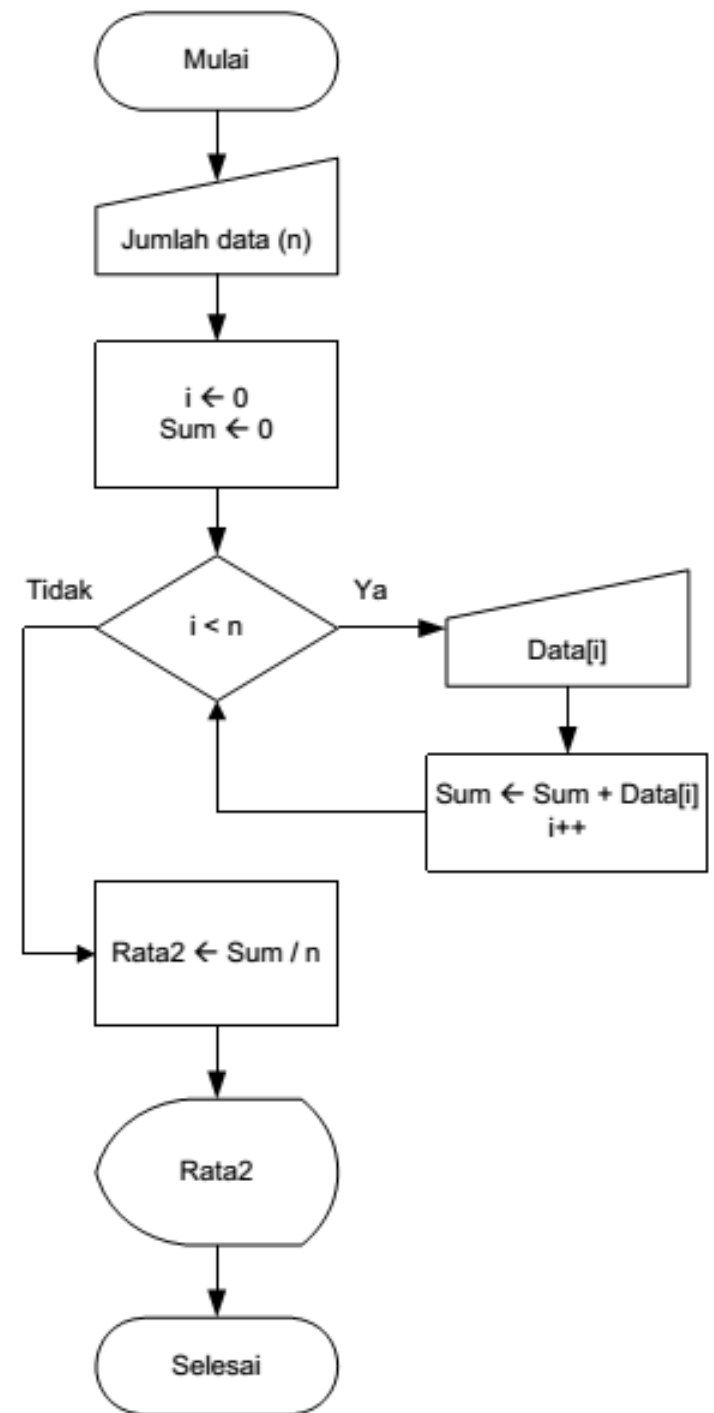
- 1) Masukan jumlah (n) data yang diinginkan.
- 2) Selama jumlah data yang dimasukkan kurang dari n, kerjakan langkah ke-3 dan ke-4. Jika tidak kerjakan langkah ke-5.
- 3) Masukan data.
- 4) Tambahkan data yang dimasukkan dengan data sebelumnya.
- 5) Hitung rata-rata data.
- 6) Tampilkan besar rata-ratanya.

Menggunakan kalimat deskriptif

Uraian diatas dapat juga ditulis seperti dibawah ini:

- 1) Masukkan n
- 2) $i \leftarrow 0$
- 3) $Sum \leftarrow 0$
- 4) Selama $i < n$, kerjakan langkah ke-5, 6, dan 7.
- 5) Masukkan data[i]
- 6) $Sum \leftarrow Sum + data[i]$
- 7) $i++$
- 8) $Rata2 \leftarrow Sum / n$
- 9) Tulis Rata2

Flow chart



Contoh permasalahan

Pseudo code

Algoritma hitung_rata2

{algoritma ini digunakan untuk menghitung rata-rata dari sejumlah data yang dimasukkan melalui keyboard}

deklarasi

i, n, sum, data[20] : integer
rata2 : float

deskripsi

read(n)
 $i \leftarrow 0$
 $sum \leftarrow 0$
while(i < n)
 read(data[i])
 $sum += data[i]$
 i++
end while
 $rata2 \leftarrow sum / n$
write(rata2)

Algoritma hitung_rata2

{algoritma ini digunakan untuk menghitung rata-rata dari sejumlah data yang dimasukkan melalui keyboard}

deklarasi

i, n, sum, data[20] : integer
rata2 : float

deskripsi

read(n)
 $sum \leftarrow 0$
for(i=0; i<n; i++)
 read(data[i])
 $sum += data[i]$
end for
 $rata2 \leftarrow sum / n$
write(rata2)

Contoh permasalahan

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     int n, i=0, sum=0, data[20];
5.     float rata2;
6.     cout << "Jumlah data (maksimum 20 data) = ";
7.     cin >> n;
8.     while(i < n) {
9.         int j = i + 1;
10.        cout << "Masukkan data ke-" << j << " : ";
11.        cin >> data[i];
12.        sum += data[i];
13.        i++;
14.    }
15.    rata2 = sum / n;
16.    cout << "Rata-ratanya = " << rata2;
17.    return 0;
18. }
```

Source code

Contoh permasalahan

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     int n, i, sum=0, data[20];
5.     float rata2;
6.     cout << "Jumlah data (maksimum 20) = ";
7.     cin >> n;
8.     for(i=0; i < n; i++) {
9.         int j = i + 1;
10.        cout << "Masukkan data ke-" << j << " : ";
11.        cin >> data[i];
12.        sum += data[i];
13.    }
14.    rata2 = sum / n;
15.    cout << "Rata-ratanya = " << rata2;
16.    return 0;
17. }
```

Source code

5. Masalah menentukan Faktor Persekutuan Besar (FPB).

- Untuk memecahkan persoalan FPB di atas, kita dapat menggunakan algoritma Euclid yang langkah-langkahnya adalah sebagai berikut :
 - 1) Diberikan dua buah bilangan bulat positif, misalnya m dan n dimana syaratnya adalah $m > n$.
 - 2) Bagilah m dengan n , misalnya hasilnya adalah r .
 - 3) Apabila $r = 0$, maka hasilnya adalah n itu sendiri (merupakan FPB dari m dan n).
 - 4) Jika $r \neq 0$ maka proses pencarian dilanjutkan yaitu
 - 5) ganti m dengan n dan n dengan r dan ulangi langkah ke-2, demikian selanjutnya.

Contoh permasalahan

Menggunakan uraian kalimat deskriptif

- 1) Masukkan m dan n .
- 2) Selama $m < n$ ulangi kembali langkah pertama.
- 3) $r \leftarrow m \% n$
- 4) Jika $r = 0$ maka $\text{fpb} \leftarrow n$. Jika tidak kerjakan langkah ke-5.
- 5) Selama $r \neq 0$ kerjakan langkah ke-6 sampai ke-8.
- 6) $m \leftarrow n$ dan $n \leftarrow r$
- 7) $r \leftarrow m \% n$
- 8) $\text{fpb} \leftarrow n$
- 9) Tulis fpb .

Contoh permasalahan

Menggunakan pseudo code

{algoritma ini digunakan untuk mencari faktor persekutuan besar dari dua buah bilangan yang dimasukkan}

deklarasi

m, n, r, fpb : integer

deskripsi

do read(m,n)

while(m<n)

end do while

$r \leftarrow m \% n$

if (r == 0) fpb \leftarrow n

else

while(r != 0)

$m \leftarrow n$

$n \leftarrow r$

$r \leftarrow m \% n$

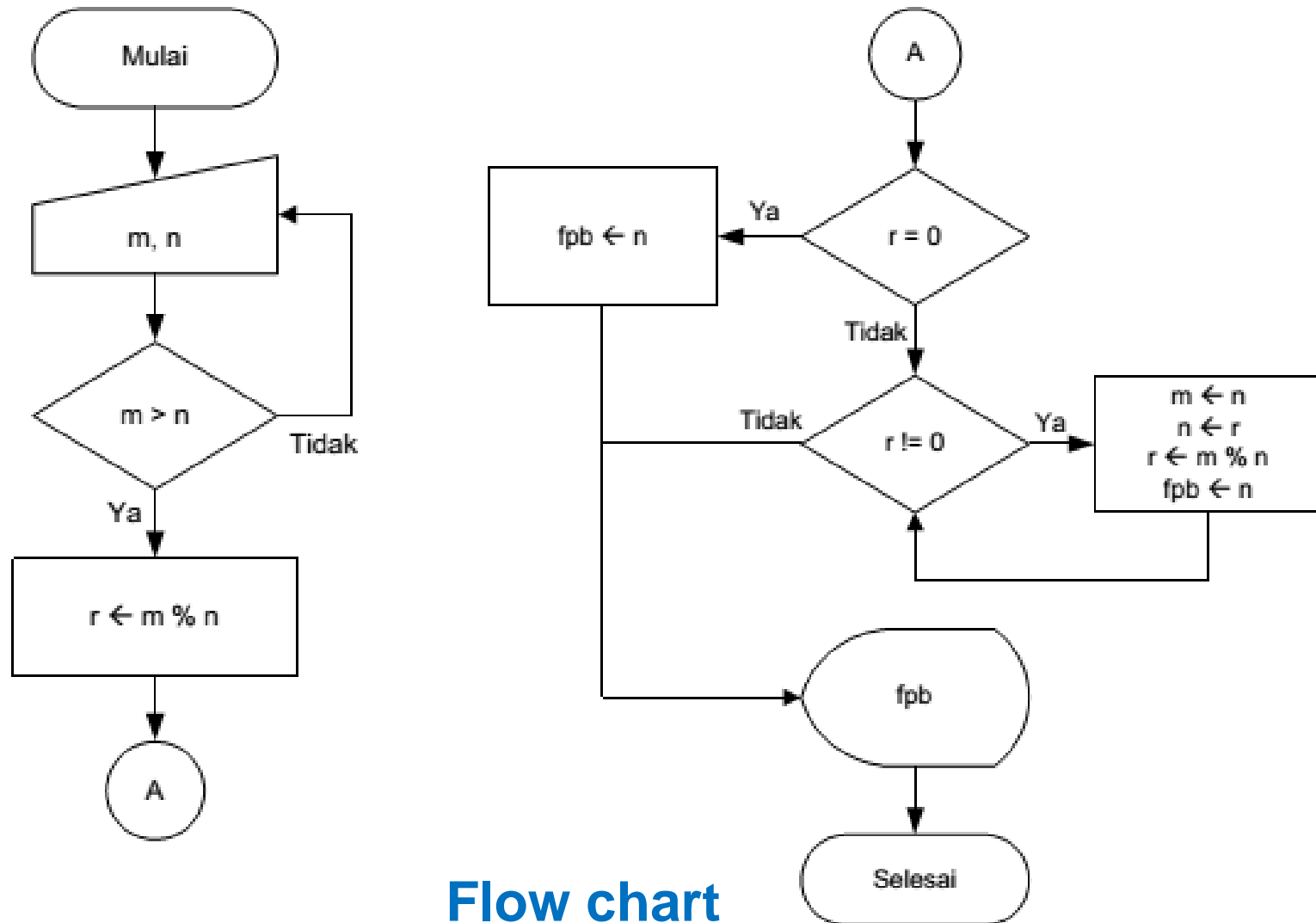
fpb \leftarrow n

end while

end if

write(fpb)

Contoh permasalahan



Flow chart

Contoh permasalahan

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     int m,n,r,fpb;
5.     do {
6.         cout << "Masukkan bilangan pertama = ";
7.         cin >> m;
8.         cout << "Masukkan bilangan kedua = ";
9.         cin >> n;
10.    } while (m < n);
11.    r = m % n;
12.    if (r==0) fpb = n;
13.    else {
18.        fpb = n;
19.    }
20. }
21. cout << "FPB-nya = " << fpb;
22. return 0;
23. }
```

Source code

Latihan

Pada pertemuan ini kita telah belajar mengenai:

1. Pengertian algoritma
2. Ciri suatu algoritma
3. Proses-proses yang terdapat pada suatu algoritma
4. Metode untuk menyajikan suatu algoritma

Tugas (Opsional)

Pre-Test (Optional)
