

MODUL MATA KULIAH

ANALISIS DAN DESAIN ALGORITMA

PG167 – 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
SEPTEMBER 2019**

TIM PENYUSUN

Atik Ariesta, S.Kom., M.Kom
Ita Novita, S.Kom., M.T.I
Dr. Achmad Solichin, S.Kom., M.T.I



MODUL PERKULIAHAN #5

STRUKTUR KONTROL

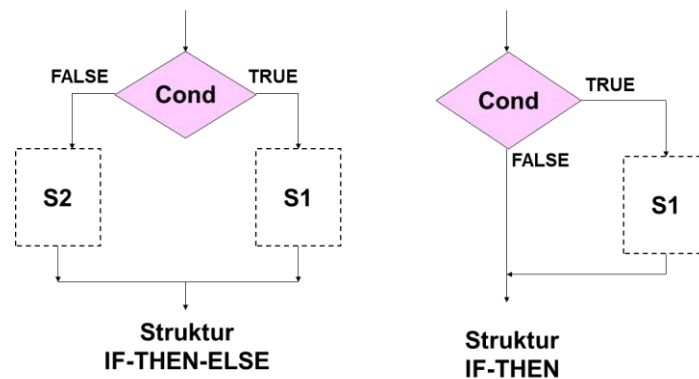
PERCABANGAN LANJUTAN

Capaian Pembelajaran	:	Mahasiswa memahami bentuk umum dan penggunaan struktur kondisi IF bertingkat dan Switch Case.
Sub Pokok Bahasan	:	1.1. Nested If 1.2. Bentuk Nested If 1.3. Multi Condition dan Logical Operator 1.4. Jenis Operator Logika 1.5. Konversi Multi Condition Menjadi Nested-If 1.6. Contoh Program Sederhana Menggunakan Nested If dan Multi Condition 1.7. Seleksi Menggunakan Switch-Case 1.8. Swtich-Case Berjenjang
Daftar Pustaka	:	1. Gaddis, nd.2011. Starting Out with C++ from Control Structures through Objects .8th. Boston: Addison-Wesley. 2. Institue of Distance & Open Learning, n.d. UNIT I Algorithms, Flowcharts & Program Design in: INTRODUCTION TO C++. p. 205 3. Sjukani,Moh .2014. Algoritma (Algoritma & Struktur Data 1) Dengan C, C++, dan Java Edisi 9", Mitra Wacana Media.

STRUKTUR KONTROL PERCABANGAN LANJUTAN

1.1. NESTED IF

Perhatikan kembali struktur IF-THEN dan IF-THEN-ELSE statement seperti di bawah ini:



Dari ilustrasi di atas, terlihat S adalah satu atau sekelompok statement. Didalam kelompok S mungkin terdapat statement IF sehingga terjadi IF secara berjenjang atau secara tersarang yang biasa disebut Nested If (nest=sarang)

1.2. BENTUK NESTED IF

Bentuk-bentuk NESTED-IF (If bersarang) dapat dilihat pada contoh-contoh di bawah ini. Contoh-contoh tersebut merupakan salah satu diantara beberapa cara yang bisa dilakukan dalam membuat Nested-If selama mengikuti aturan penggambaran Flowchart pada IF-THEN dan IF-THEN-ELSE.



CONTOH 1

Penggalan Program	Pseudocode	Flowchart
<pre> if (cond1) { if (cond2) { - - S1 - } } else { - - S2 - } </pre>	<pre> IF (cond1) THEN IF(cond2) THEN EndIF ELSE EndIF </pre>	<pre> graph TD Start(()) --> Cond1{Cond1} Cond1 -- F --> S2[S2] S2 --> EndIF1((EndIF)) Cond1 -- T --> Cond2{Cond2} Cond2 -- T --> S1[S1] S1 --> EndIF2((EndIF)) Cond2 -- F --> EndIF2 EndIF1 --> Exit(()) EndIF2 --> Exit </pre>

CONTOH 2

Penggalan Program	Pseudocode	Flowchart
<pre> if (cond1) { if (cond2) { - - S1 - } else { - - S2 - } } else { - - S3 - } </pre>	<pre> IF (cond1) THEN IF(cond2) THEN ELSE EndIF ELSE EndIF </pre>	<pre> graph TD Start(()) --> Cond1{Cond1} Cond1 -- F --> S3[S3] S3 --> EndIF1((EndIF)) Cond1 -- T --> Cond2{Cond2} Cond2 -- T --> S2[S2] S2 --> EndIF2((EndIF)) Cond2 -- F --> S1[S1] S1 --> EndIF2 EndIF1 --> Exit(()) EndIF2 --> Exit </pre>



CONTOH 3

Penggalan Program	Pseudocode	Flowchart
<pre> if cond1 { - S1 - if cond2 { - S2 } - S3 } else { if cond3 { - S4 } else { - S5 } } </pre>	<pre> IF (cond1) THEN :: S1 IF(cond2) THEN :: S2 ENDIF :: S3 ELSE IF(cond3) THEN :: S4 ELSE :: S5 ENDIF EndIF </pre>	<pre> graph TD Start(()) --> Cond1{Cond1} Cond1 -- T --> S1[S1] S1 --> Cond2{Cond2} Cond2 -- T --> S2[S2] S2 --> Join(()) Cond2 -- F --> S3[S3] S3 --> Join Cond1 -- F --> Cond3{Cond3} Cond3 -- T --> S4[S4] S4 --> Join Cond3 -- F --> S5[S5] S5 --> Join Join --> End(()) </pre> <p>Perhatikan posisi letak 'titik' Endif (akhir fungsi if) dalam flowchart. Posisi ini penting untuk menganalisa aliran terutama untuk nested IF yang kompleks atau untuk proses pengulangan yang bersifat rekursif.</p>



CONTOH 4

Penggalan Program	Pseudocode	Flowchart
<pre> if (cond1) { - S1 - if (cond2) { - S2 - } - else { - S3 - } } else { if (cond3) { - S4 - } } else { - S5 - } - S6 } </pre>	<pre> IF (cond1) THEN S1 IF(cond2) THEN S2 ELSE S3 EndIF ELSE IF(cond3) THEN S4 ELSE S5 EndIF S6 EndIF </pre>	<pre> graph TD Start(()) --> Con1{Con1} Con1 -- T --> S1[S1] Con1 -- F --> Con3{Con3} Con3 -- T --> S4[S4] Con3 -- F --> S5[S5] S4 --> S6[S6] S5 --> S6 S1 --> Con2{Con2} Con2 -- T --> S2[S2] Con2 -- F --> S3[S3] S2 --> End(()) S3 --> End S6 --> End </pre>

CONTOH 5

Penggalan Program	Pseudocode	Flowchart
<pre> if (cond1) {if (cond2) {if (cond3) {if (cond4) { - S1 - } } } } } </pre>	<pre> IF (cond1) THEN IF(cond2) THEN IF(cond3) THEN IF(cond4) THEN S1 EndIF EndIF EndIF </pre>	<pre> graph TD Start(()) --> Cond1{Cond1} Cond1 -- True --> Cond2{Cond2} Cond1 -- False --> endif1[endif] Cond2 -- True --> Cond3{Cond3} Cond2 -- False --> endif2[endif] Cond3 -- True --> Cond4{Cond4} Cond3 -- False --> endif3[endif] Cond4 -- True --> S1[S1] Cond4 -- False --> endif4[endif] S1 --> End(()) endif1 --> End endif2 --> End endif3 --> End endif4 --> End </pre> <p>S1 dilaksanakan, hanya bila keempat kondisi nilainya TRUE</p>



CONTOH 6

Penggalan Program	Pseudocode	Flowchart
<pre> if(cond1) { - S1 - } else {if(cond2) { - S2 - } else {if(cond3) { - S3 - } else {if(cond3) { - S4 - } else {if(cond3) { - S4 - } else { - S5 } } } } </pre>	<pre> IF (cond1) THEN :: S1 ELSE IF(cond2) THEN :: S2 ELSE IF(cond3) THEN :: S3 ELSE IF(cond4) THEN :: S4 ELSE :: S5 ENDIF ENDIF ENDIF </pre>	

1.3. MULTI CONDITION DAN LOGICAL OPERATOR

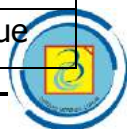
Kadang-kadang satu kondisi saja tidak cukup untuk menentukan suatu syarat, sehingga diperlukan dua atau lebih kondisi. Untuk menggabungkan kondisi-kondisi tersebut gunakan operator logika yang disebut logical operator.

1.4. JENIS OPERATOR LOGIKA

OPERATOR NOT

Operator NOT, bukan digunakan untuk menggabungkan dua buah kondisi tapi bekerja sebagai pembalik nilai logika TRUE menjadi FALSE, FALSE menjadi TRUE sehingga sering disebut Unary Operator.

No	A	B	Condition	Nilai	Not (Condition)	Sama Maksudnya	Nilai
1.	5	2	A == B	False	! (A == B)	A != B	True
2.	5	2	A > B	True	! (A > B)	A <= B	False
3.	5	2	A < B	False	! (A < B)	A >= B	True



No	A	B	Condition	Nilai	Not (Condition)	Sama Maksudnya	Nilai
4.	5	2	$A \geq B$	True	$!(A \geq B)$	$A < B$	False
5.	5	2	$A \leq B$	False	$!(A \leq B)$	$A > B$	True
6.	5	2	$A \neq B$	True	$!(A \neq B)$	$A == B$	False

OPERATOR AND

Operator AND menggabungkan dua buah kondisi menjadi satu nilai sedemikian rupa akan bernilai TRUE hanya bila kedua kondisi yang digabungkan bernilai TRUE. Dengan istilah lain, kedua syarat harus dipenuhi.

Syntax Operator AND pada kondisi: `if (cond1 && cond2)` atau `if ((cond1) && (cond2))`

Contoh: `if (Kode==1 && Umur<=25)` atau `if(Nilai >=60 && Nilai<70)`

Tabel Kebenaran untuk operator AND

No	Cond1	Cond2	Cond1 AND Cond2
1.	True	True	True
2.	True	False	False
3.	False	True	False
4.	False	False	False

Perhatikan tabel kebenaran No. 1 hanya bila kedua kondisi bernilai TRUE, yang akan menghasilkan nilai gabungan = TRUE

Contoh Tabel kebenaran untuk melihat nilai gabungan dua buah kondisi yang digabung dengan operator AND

No	A	B	C	D	Cond1	Cond2	Cond1 && Cond2	Nilai Akhir
1.	5	2	6	4	$A > B$ (True)	$C > D$ (True)	True && True	True
2.	5	2	6	4	$A > B$ (True)	$B > D$ (False)	True && False	False
3.	5	2	6	4	$A > C$ (False)	$B > D$ (False)	False && False	False



No	A	B	C	D	Cond1	Cond2	Cond1 && Cond2	Nilai Akhir
4.	5	2	6	4	A>B (True)	!(B>D) (True)	True && True	True

OPERATOR OR

Operator OR menggabungkan dua buah kondisi menjadi satu nilai sedemikian rupa akan bernilai TRUE cukup bila salah satu saja dari kedua kondisi yang digabungkan bernilai TRUE. Hanya bila kedua kondisi bernilai FALSE, maka nilai gabungannya bernilai FALSE. Dengan perkataan lain cukup satu syarat saja yang harus dipenuhi.

Syntax Operator OR pada kondisi: if (cond1 || cond2) atau if ((cond1) || (cond2))

Contoh: if (Status==1 || Umur>=17) atau if(Nil1 >=60 || Nil2>=65)

Tabel Kebenaran untuk operator OR

No	Cond1	Cond2	Cond1 OR Cond2
1.	True	True	True
2.	True	False	True
3.	False	True	True
4.	False	False	False

Perhatikan tabel kebenaran No. 1, 2, dan 3 hanya salah satu kondisi bernilai TRUE, maka akan menghasilkan nilai gabungan = TRUE

Contoh Tabel kebenaran untuk melihat nilai gabungan dua buah kondisi yang digabung dengan operator AND

No	A	B	C	D	Cond1	Cond2	Cond1 Cond2	Nilai Akhir
1.	5	2	6	4	A>B (True)	C>D (True)	True True	True
2.	5	2	6	4	A>B (True)	B>D (False)	True False	True
3.	5	2	6	4	A>C (False)	B>D (False)	False False	False



No	A	B	C	D	Cond1	Cond2	Cond1 Cond2	Nilai Akhir
4.	5	2	6	4	!(A>B) (False)	!(C>D) (False)	True True	False

1.5. KONVERSI MULTI CONDITION MENJADI NESTED IF

Multi Condition dapat dikonversikan menjadi bentuk Nested If. Contoh-contoh konversi di bawah ini bukan bermaksud menerangkan TRUE atau FALSE suatu kondisi tetapi contoh ini diperlukan jika harus menulis dalam bentuk Nested If.

CONTOH 1

Multi Kondisi	Nested If	Flowchart
if (cond1 && cond2) { S1 } else { S2 }	if (cond1) { if (cond2) { S1 } else { S2 } } else { S2 }	<pre> graph TD Start(()) --> Cond1{Cond1} Cond1 -- F --> S2_1[S2] Cond1 -- T --> Cond2{Cond2} Cond2 -- F --> S2_2[S2] Cond2 -- T --> S1[S1] S2_1 --> EndIF1[EndIF] S2_2 --> EndIF1 S1 --> EndIF1 EndIF1 --> Exit(()) </pre>
	Penggalan perintah nested if di atas, merupakan salah satu cara (bukan satu-satunya)	



CONTOH 2

Multi Kondisi	Nested If	Flowchart
<pre> if (cond1 cond2) { S1 } else { S2 } </pre>	<pre> if (cond1) { S1 } else { if(cond2) { S1 } else { S2 } } </pre>	

CONTOH 3

Multi Kondisi	Nested If	Flowchart
<pre> if (cond1 && cond2 && cond3) { S1 } else { S2 } </pre> <p>Yang diproses oleh komputer:</p> <pre> cond1 && cond2 && cond3 </pre>	<pre> if (cond1) { if (cond2) { if (cond3) { S1 } else { S2 } } else { S2 } } else { S2 } </pre>	



CONTOH 4

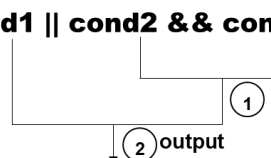
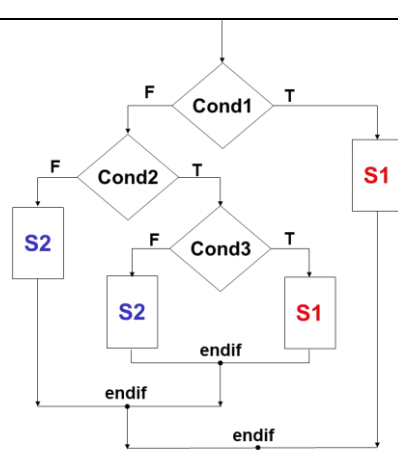
Multi Kondisi	Nested If	Flowchart
<pre>if (cond1 cond2 cond3) { S1 } else { S2 }</pre> <p>Yang diproses oleh komputer:</p> <p>cond1 cond2 cond3</p>	<pre>if (cond1) { S1 } else { if(cond2) { S1 } else { S2 } } }</pre>	

CONTOH 5

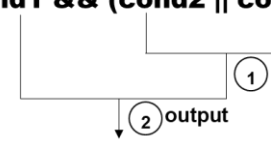
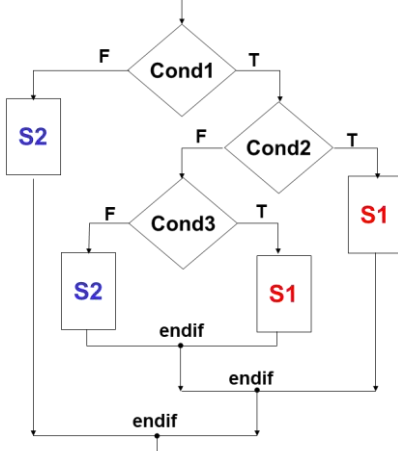
Multi Kondisi	Nested If	Flowchart
<pre>if (cond1 && cond2 cond3) { S1 } else { S2 }</pre> <p>Yang diproses oleh komputer:</p> <p>cond1 && cond2 cond3</p>	<pre>if (cond3) { S1 } else { if(cond1) { if(cond2) { S1 } else { S2 } } else { S2 } } }</pre>	



CONTOH 6

Multi Kondisi	Nested If	Flowchart
<pre>if (cond1 cond2 && cond3) { S1 } else { S2 }</pre> <p>Yang diproses oleh komputer:</p> <pre>cond1 cond2 && cond3</pre> 	<pre>if (cond1) { S1 } else { if(cond2) { if(cond3) { S1 } else { S2 } } else { S2 } }</pre>	

CONTOH 7

Multi Kondisi	Nested If	Flowchart
<pre>if (cond1 && (cond2 cond3)) { S1 } else { S2 }</pre> <p>Yang diproses oleh komputer:</p> <pre>cond1 && (cond2 cond3)</pre> 	<pre>if (cond1) { if(cond2) { S1 } else { if(cond3) { S1 } else { S2 } } } else { S2 }</pre>	



1.6. CONTOH PROGRAM SEDERHANA MENGGUNAKAN NESTED IF DAN MULTI CONDITION

Susun Algoritma untuk menginput 3 (tiga) bilangan bulat (integer), dimana ketiga buah bilangan tersebut dianggap bernilai tidak sama, kemudian mencetak salah satu bilangan yang nilainya terbesar.

JAWAB:

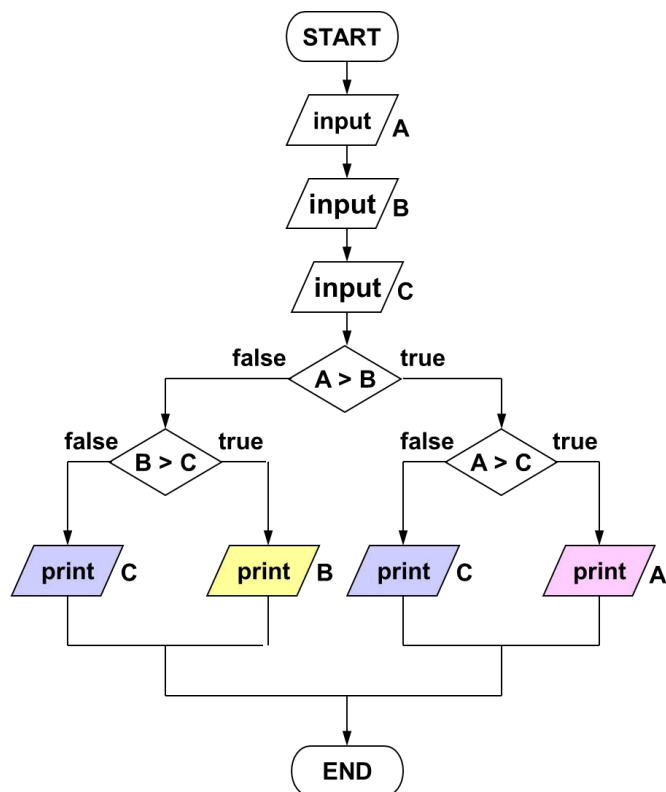
CARA – 1

Cara satu dengan menggunakan tiga variabel untuk menyimpan Bilangan 1, Bilangan 2, dan Bilangan 3. Kemudian ketiga bilangan tersebut diperiksa satu persatu sehingga mendapatkan nilai yang terbesar.

No	Pseudocode	Simbol Flowchart
1.	Inisialisasi A, B, C	Proses
2.	Input A, B, C	Input Outpt
3.	IF (A>B) THEN IF(A>C) THEN Cetak A ELSE Cetak B ENDIF ELSE IF (B>C) THEN Cetak B ELSE Cetak C ENDIF ENDIF	Kondisi, Input Output



Flowchart Cara – 1



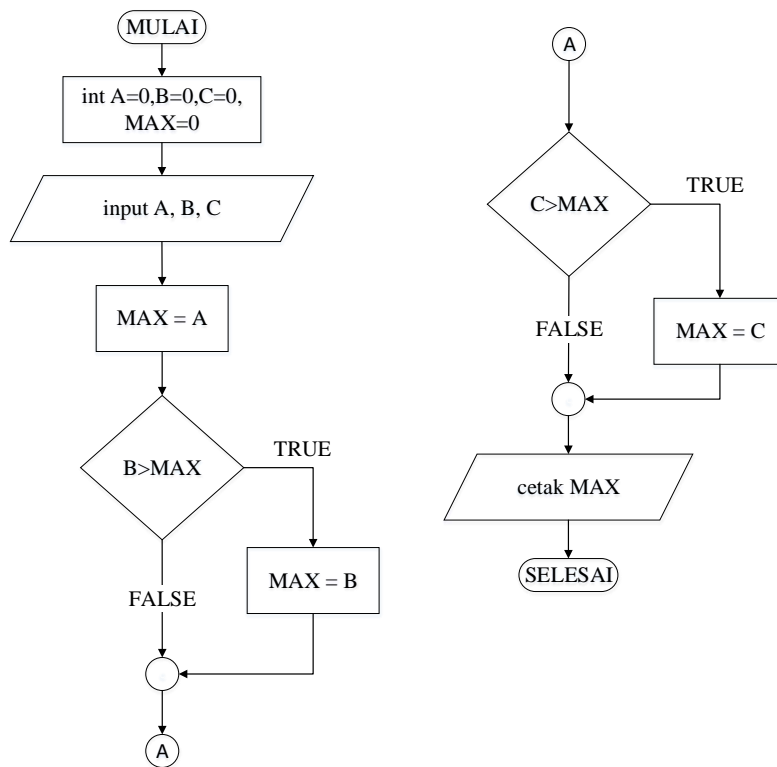
CARA – 2

Cara kedua dengan menggunakan empat variabel yaitu untuk menyimpan Bilangan 1, Bilangan 2, Bilangan 3, dan variabel MAX untuk menyimpan nilai maksimal yang di dapat. Bilangan yang diinput akan diperiksa satu-persatu jika bilangan tersebut merupakan bilangan terbesar maka akan disimpan kedalam MAX

No	Pseudocode	Simbol Flowchart
1.	Inisialisasi A, B, C, MAX	Proses
2.	Input A, B, C	Input Output
3.	Isikan MAX dengan Nilai A	Proses
4.	IF (B>MAX) THEN MAX = B ENDIF IF (C>MAX) THEN MAX = C ENDIF	Kondisi, Input Output
5.	Cetak MAX	Input Output



Flowchart Cara – 2



CARA – 3

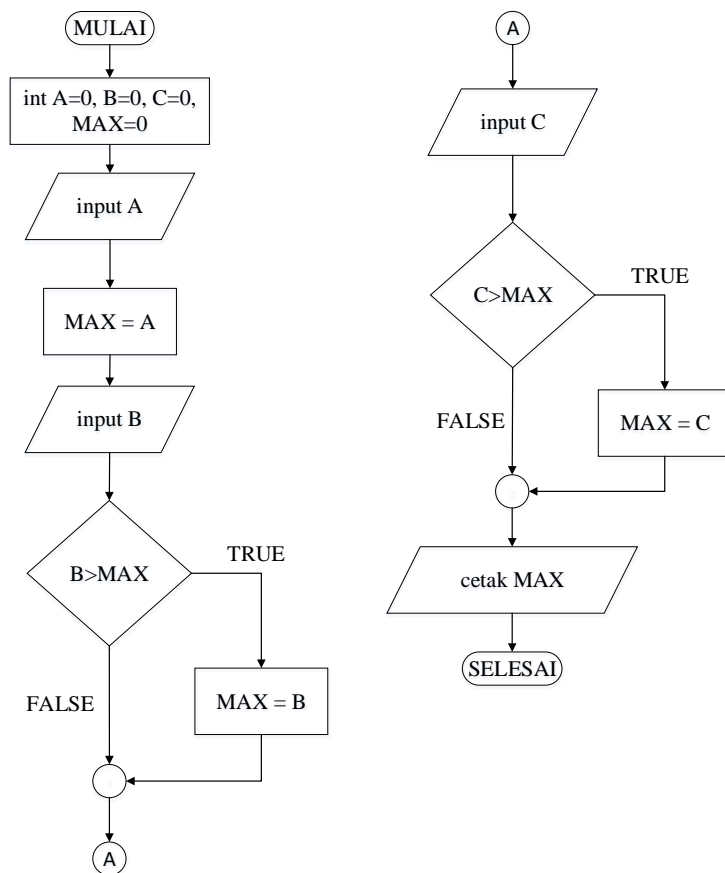
Cara ketiga hampir sama dengan Cara – 2 yaitu dengan menggunakan empat variabel yaitu untuk menyimpan Bilangan 1, Bilangan 2, Bilangan 3, dan variabel MAX untuk menyimpan nilai maksimal yang di dapat. Perbedaannya, setiap menginput Bilangan baik bilangan pertama, kedua, atau ketiga langsung dilakukan pengecekan, jika ternyata bilangan yang diinput merupakan bilangan yang terbesar maka akan disimpan kedalam variabel MAX

No	Pseudocode	Simbol Flowchart
1.	Inisialisasi A, B, C, MAX	Proses
2.	Input A	Input Output
3.	Isikan MAX dengan Nilai A	Proses
4.	Input B	Input Output
5.	IF (B>MAX) THEN MAX = B ENDIF	Kondisi, Input Output



No	Pseudocode	Simbol Flowchart
6.	Input C	Input Output
7.	IF (C>MAX) THEN MAX = C ENDIF	
8.	Cetak MAX	Input Output

Flowchart Cara – 3



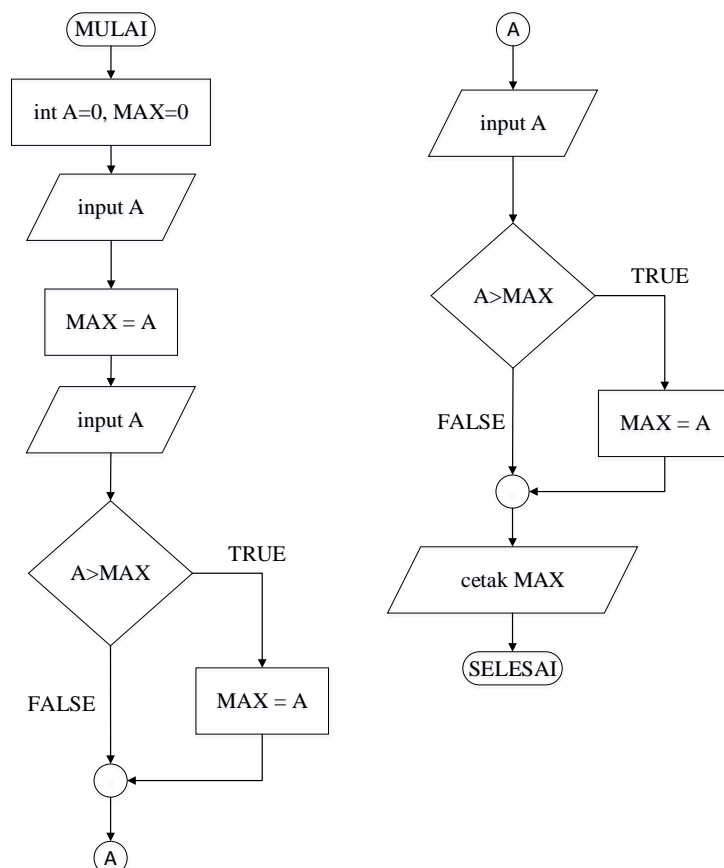
CARA – 4

Cara keempat hanya menggunakan dua variabel yaitu satu variabel yaitu A untuk menyimpan tiga bilangan yang diinput (secara bergantian), dan satu variabel lagi yaitu MAX untuk menyimpan nilai maksimal yang didapat. Setiap menginput bilangan akan di periksa yang jika ternyata nilainya lebih besar dari nilai sebelumnya, akan disimpan ke dalam variabel MAX.



No	Pseudocode	Simbol Flowchart
1.	Inisialisasi A, MAX	Proses
2.	Input A	Input Output
3.	Isikan MAX dengan Nilai A	Proses
4.	Input A	Input Output
5.	IF (A>MAX) THEN MAX = A ENDIF	Kondisi, Input Output
6.	Input A	Input Output
7.	IF (A>MAX) THEN MAX = A ENDIF	
8.	Cetak MAX	Input Output

Flowchart Cara – 4



Algoritma ini nanti yang akan menjadi dasar algoritma pencarian bilangan terbesar atau terkecil dari sejumlah bilangan yang ada atau bilangan yang diinput

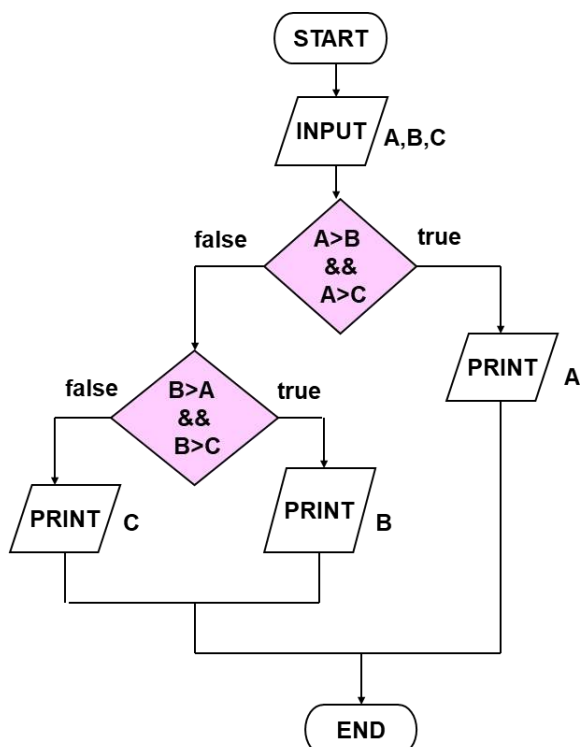


CARA – 5

Cara kelima menggunakan tiga variabel untuk menyimpan masing-masing bilangan, tetapi pemeriksaan kondisinya menggunakan Multi Condition dengan NESTED-IF

No	Pseudocode	Simbol Flowchart
1.	Inisialisasi A,B,C	Proses
2.	Input A, B, C	Input Output
3.	IF (A>B && A>C) THEN Cetak A ELSE IF (B>A && B>C) THEN Cetak B ELSE Cetak C ENDIF ENDIF	Kondisi, Input Output

Flowchart Cara – 5



CARA – 6

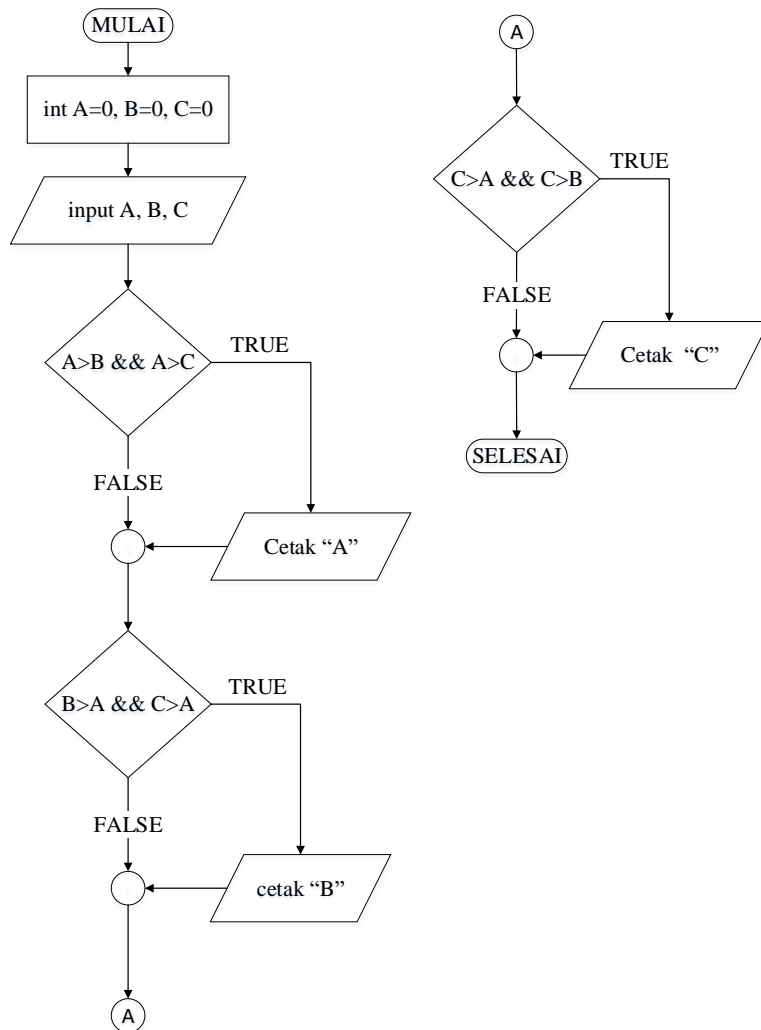
Cara keenam menggunakan 3 Variabel untuk menyimpan masing-masing bilangan, kemudian bilangan tersebut diperiksa dengan menggunakan IF-THEN dimana kondisinya menggunakan MULTI CONDITION.

Cara ini paling mudah logikanya tetapi dengan cara seperti ini computer-time bisa lebih panjang karena komputer akan melaksanakan semua instruksi if.

No	Pseudocode	Simbol Flowchart
1.	Inisialisasi A,B,C	Proses
2.	Input A, B, C	Input Output
3.	IF (A>B && A>C) THEN Cetak "A" ENDIF IF (B>A && B>C) THEN Cetak "B" ENDIF IF (C>A && C>B) THEN Cetak "C" ENDIF	Kondisi, Input Output



Flowchart Cara – 6



1.7. SELEKSI MENGGUNAKAN SWITCH-CASE

Switch-Case merupakan jenis seleksi yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternative penyelesaian.

Kondisi Switch-Case terdiri dari dua bagian, yaitu Switch dimana terdapat variabel yang akan diperiksa. Serta satu atau lebih perintah Case masing-masing untuk setiap nilai yang ingin diperiksa.

Switch-Case memiliki batasan, yaitu:

1. Data yang diperiksa harus beritpe integer atau karakter
2. Range data yang diperiksa bernilai 0 s/d 255



DEFAULT

Blok default merupakan perintah yang akan dijalankan jika semua kondisi di Case tidak ada yang True.

BREAK

Break diberikan pada Blok Case, dengan tujuan untuk keluar dari Case yang bernilai true dan lanjut pada next-instruction sehingga tidak perlu memeriksa pada perintah case selanjutnya.

CONTOH SELEKSI MENGGUNAKAN SWITCH-CASE

Susun Algoritma atau Program untuk menginput sebuah nilai integer (misal N) kemudian cetak peringkat nilai sesuai dengan nilai N sebagai berikut:

Nilai N	Peringkat yang akan dicetak
1	Kurang Sekali
2	Kurang
3	Cukup
4	Baik
5	Baik Sekali

Jawab:

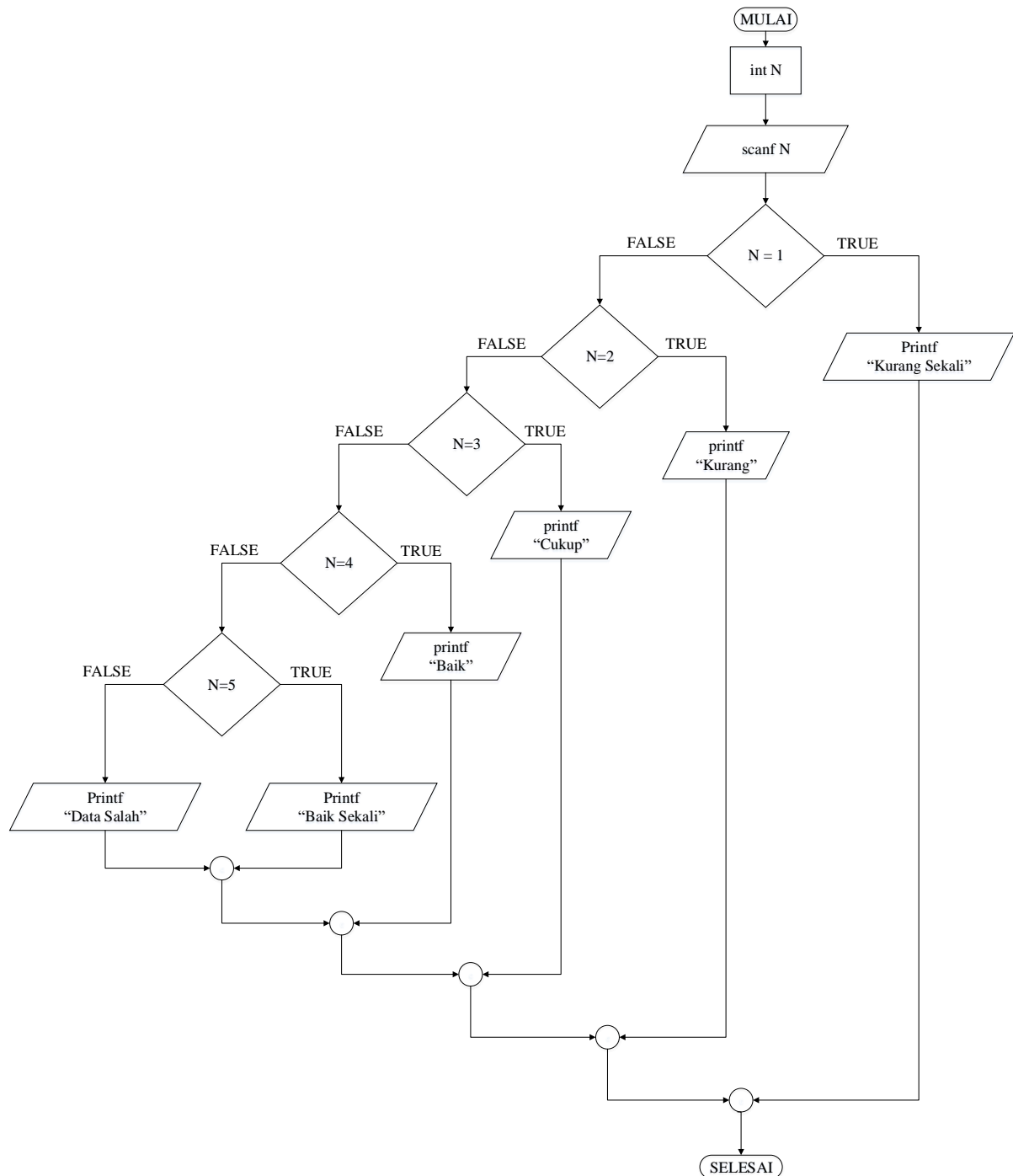
Cara – 1 (Menggunakan Nested IF)

No	Pseudocode	Simbol Flowchart
1.	Inisialisasi Nilai N	Proses
2.	Input Nilai N	Input Output
3.	IF (N=1) THEN Cetak "KURANG SEKALI" ELSE IF (N=2) THEN Cetak "Kurang" ELSE IF (N=3) THEN Cetak "Cukup" ELSE IF (N=4) THEN Cetak "Baik" ELSE IF (N=5)	Kondisi, Input Output



No	Pseudocode	Simbol Flowchart
	THEN Cetak "Baik Sekali" ELSE Cetak "Data Salah" ENDIF	

Flowchart Cara – 1

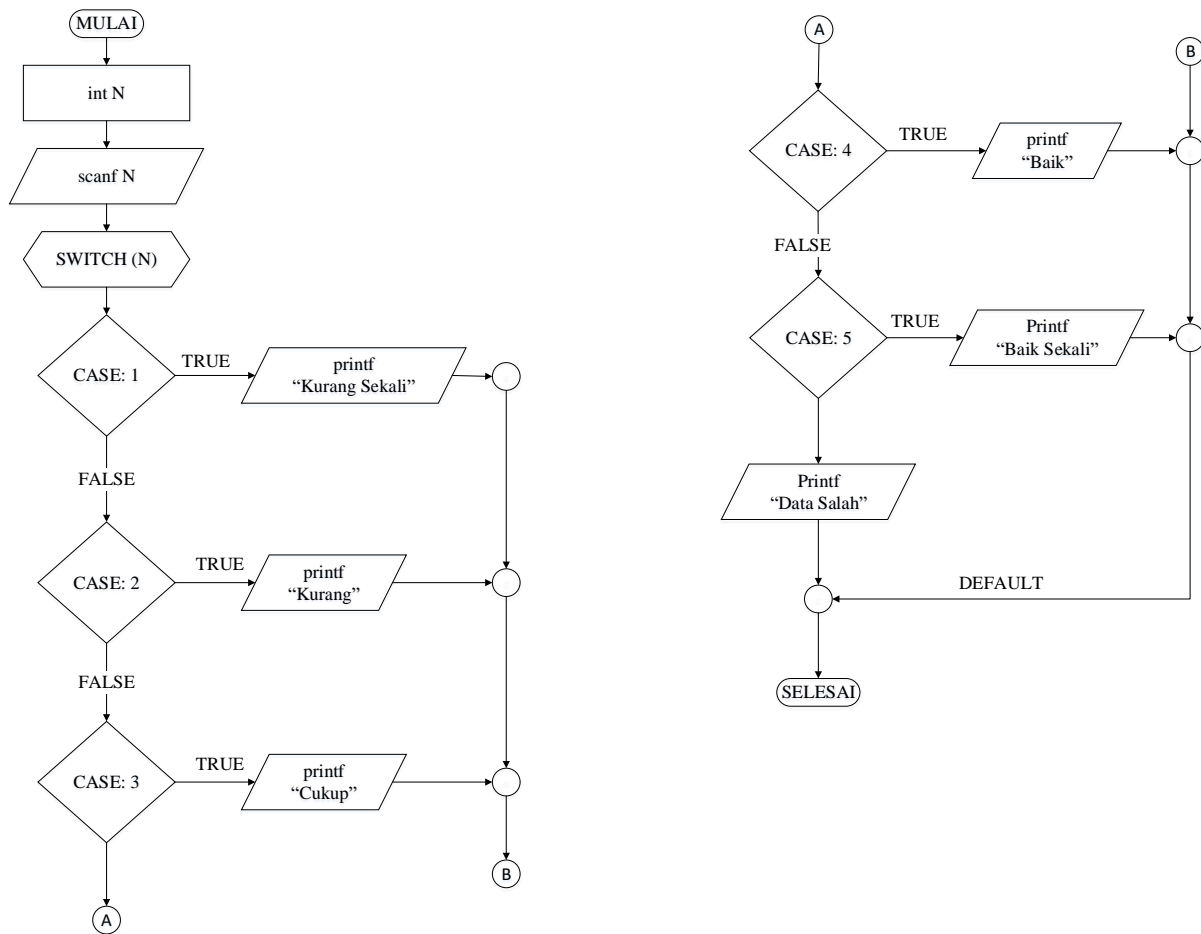


Cara – 2 (Menggunakan Switch-Case)

No	Pseudocode	Simbol Flowchart
1.	Inisialisasi Nilai N	Proses
2.	Input Nilai N	Input Output
3.	CASE 1: Cetak "Kurang Sekali" break CASE 2: Cetak "Kurang" break CASE 3: Cetak "Cukup" break CASE 4: Cetak "Baik" break CASE 5: Cetak "Baik Sekali" break Default: Cetak "Data Salah"	Kondisi, Input Output



Flowchart Cara – 2



1.8. SWITCH-CASE BERJENJANG

Switch-Case berjenjang seperti Nested-IF (If bersarang).

CONTOH SWITCH-CASE BERJENJANG

Buatlah sebuah program atau Algoritma yang digunakan untuk menginput inisial Pulau beserta dengan nama kota yang berada di pulau tersebut. Dengan data Pulau beserta dengan nama kota sebagai berikut:

Pulau		Kota	
Kode Pulau	Nama Pulau	Kode Kota	Nama Kota
J	Pulau Jawa	1	Jakarta
		2	Surabaya
		3	Bandung



		4	Semarang
		5	Yogyakarta
S	Pulau Sumatera	1	Medan
		2	Palembang
		3	Padang
K	Pulau Kalimantan	1	Banjarmasin
		2	Pontianak

Misalkan:

Bila diinput : J 2 <enter>

Maka tercetak : Pulau Jawa
Surabaya

Bilai diinput : K 2 <enter>

Maka tercetak : Pulau Kalimantan
Pontianak

Jawab:

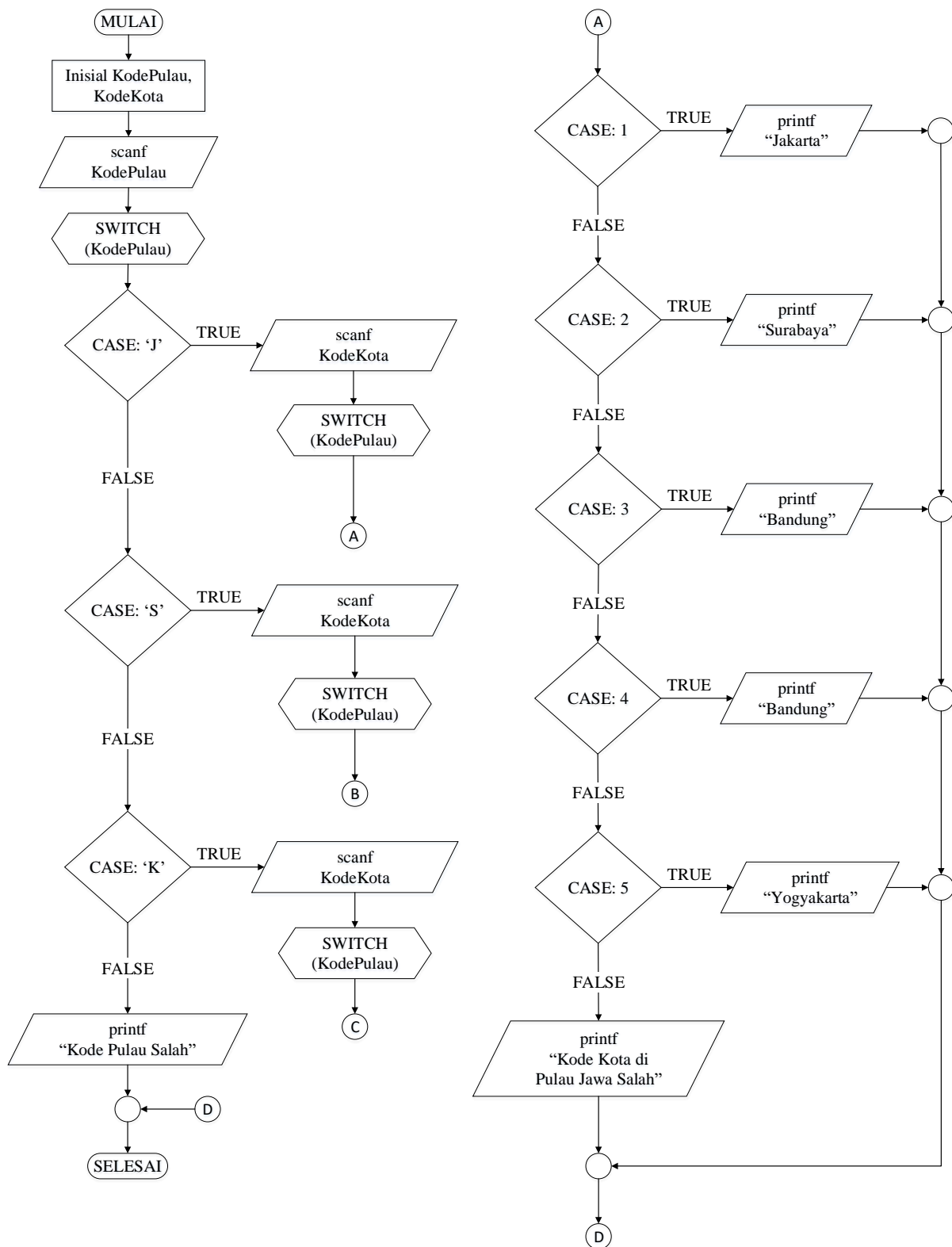
No	Pseudocode	Simbol Flowchart
1.	Inisialisasi KodePulau, KodeKota	Proses
2.	Input KodePulau	Input Output
3.	CASE 'J': Cetak "Pulau Jawa" Input KodeKota CASE 1: Cetak "Jakarta" break; CASE 2: Cetak "Surabaya" break; CASE 3: Cetak "Semarang"	Kondisi, Input Output

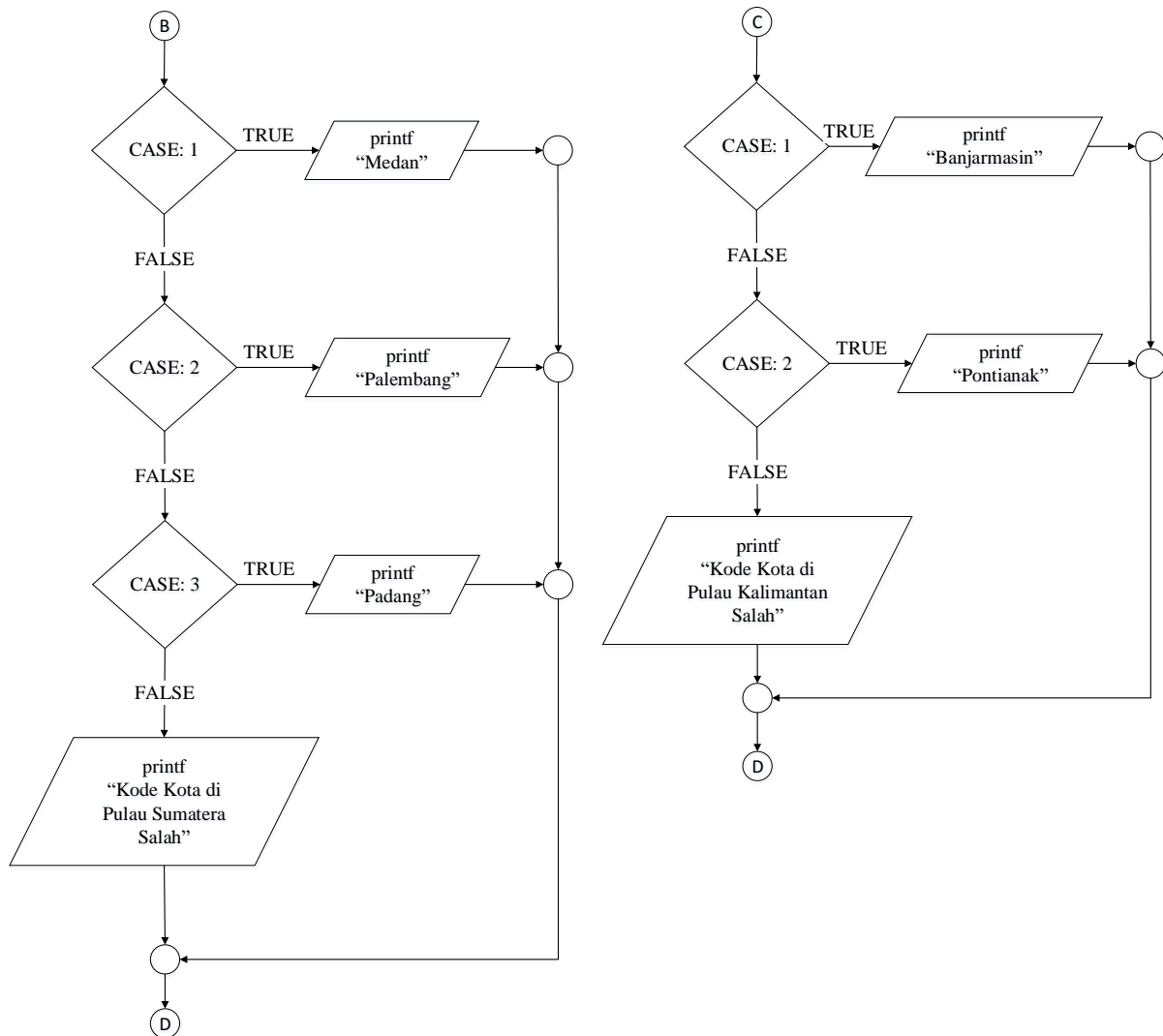


No	Pseudocode	Simbol Flowchart
	<pre> break; CASE 4: Cetak "Yogyakarta" break; default: Cetak "Kode Kota di Pulau Jawa Salah" CASE 'S': Cetak "Pulau Sumatera" Input Kode Kota Case 1: Cetak "Medan" break; Case 2: Cetak "Palembang" break; Case 3: Cetak "Padang" break; default: Cetak "Kode Kota di Sumatera Salah" Case 'K': Cetak "Pulau Kalimantan" Input Kode Kota Case 1: Cetak "Banjarmasin" break; Case 2: Cetak "Pontianak" break; default: Cetak "Kode Kota di Kalimantan Salah" default: Cetak "Kode Pulau Salah" </pre>	



Flowchart





SOAL LATIHAN

1. Susun algoritma (program) untuk menginput 3 buah bilangan yang masing-masing menyatakan panjang sisi sebuah segitiga. Kemudian periksa ketiga buah garis (sisi) tersebut.

Bila ketiga buah garis (sisi) tersebut panjangnya sama maka cetak perkataan "SAMA SISI". Bila hanya dua sisi yang sama maka cetak perkataan "SAMA KAKI". Tapi bila ketiga-tiganya tidak sama maka cetak perkataan "SEMBARANG".

Tidak boleh menggunakan logical operator AND dan OR.

2. Susun program untuk menginput tiga buah bilangan yang menyatakan nilai ujian tiga buah mata kuliah.
 - a. Cetak perkataan "TIGA" bila ketiga mata kuliah tersebut mendapat nilai lulus.



- b. Cetak perkataan "DUA", bila hanya dua dari dari ketiga mata kuliah tersebut yang mendapat nilai lulus.
 - c. Cetak perkataan "SATU" bila hanya satu mata kuliah yang mendapat nilai lulus.
 - d. Cetak perkataan "NOL" bila ketiga mata kuliah tersebut dinyatakan tidak lulus
 - e. Sebuah mata kuliah dinyatakan mendapat nilai lulus bila nilainya lebih besar atau sama dengan 60.
3. Tulis program untuk menentukan lama bekerja seorang pegawai, jika jam masuk dan jam pulang diinput. Catatan: jam berupa angka 1-12, dan seorang pegawai bekerja kurang dari 12 jam.

Contoh keluaran :

Jam masuk	Jam keluar	Keluaran/tampilan
10	11	Lama bekerja 1 jam
10	2	Lama bekerja 4 jam
10	7	Lama bekerja 9 jam

4. Buatlah program dalam bahasa C untuk menyelesaikan masalah berikut :
 Program akan menerima masukan berupa kode, jenis dan harga, dengan jenis adalah "A", "B", dan "C". Untuk setiap jenis, masing-masing akan diberikan diskon sebesar 10% untuk A, 15% untuk B, dan 20% untuk C. Program akan menghitung berapa harga setelah didiskon.

Contoh masukan :

Kode =10

Jenis =B

Harga= 10000

Contoh keluaran :

Jenis barang B mendapat diskon = 15%, Harga setelah didiskon = 8500



5. Buatlah sebuah program untuk menginput data nilai matakuliah berupa NAMA MATAKULIAH, SKS dan NILAI. Selanjutnya tentukan dan tampilkan GRADE matakuliah berdasarkan NILAI yang diinput (gunakan aturan grade baru).

Contoh masukan :

Matakuliah = LogikaMatematika

SKS = 3

Nilai = 84

Contoh keluaran :

Grade = A-

KESIMPULAN

Dalam membuat sebuah perintah algoritma dengan menggunakan struktur kontrol percabangan bisa saja kemungkinan digunakan Nested-If (If berjenjang) atau Switch-Case.

Pilihlah struktur yang sesuai dengan ketentuan Algoritma benar dan efisien





FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>