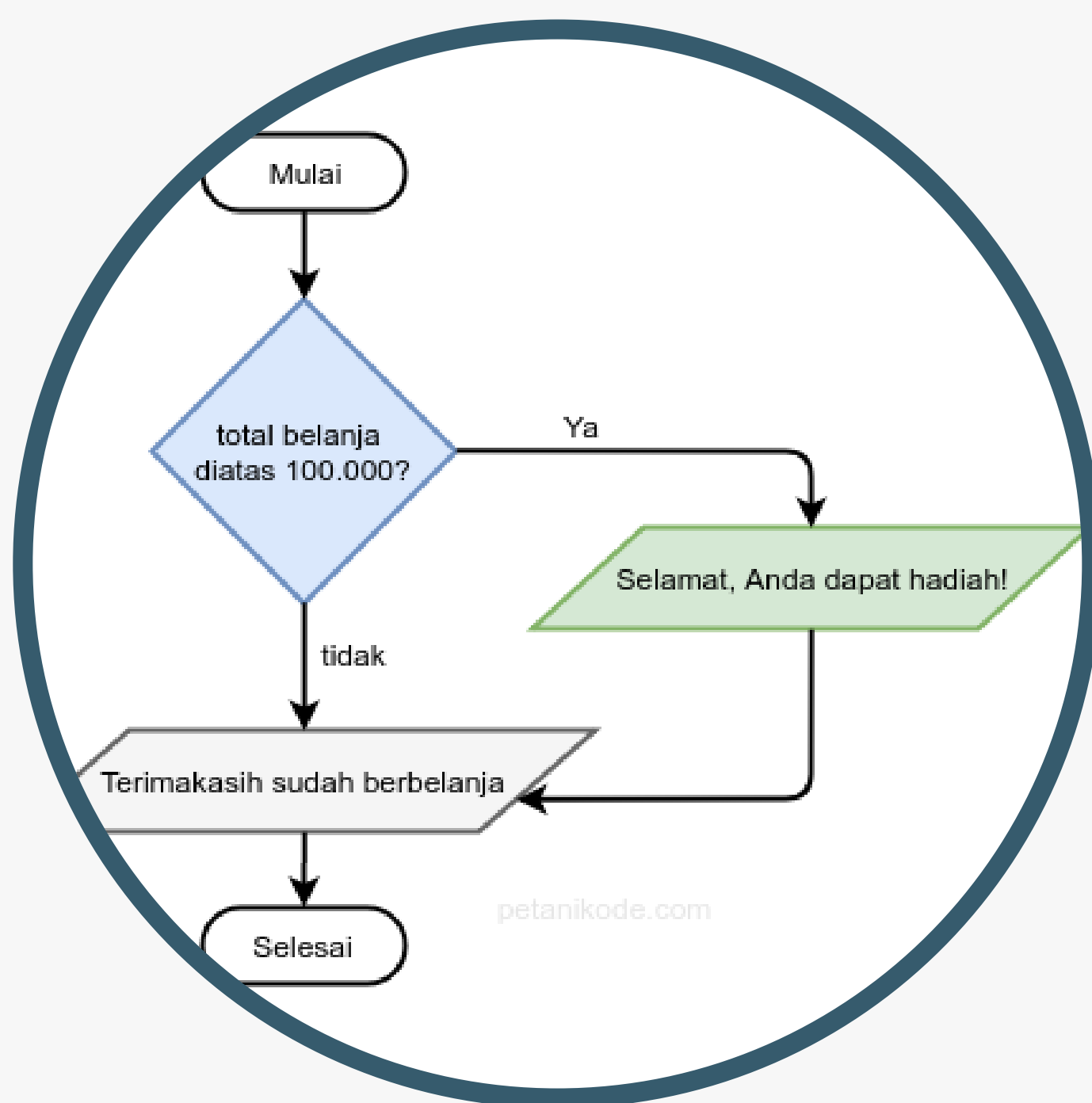


Modul Percabangan

Modul Percabangan Bagian II



Disusun Oleh :
Johannes Alexander Putra

Pengantar

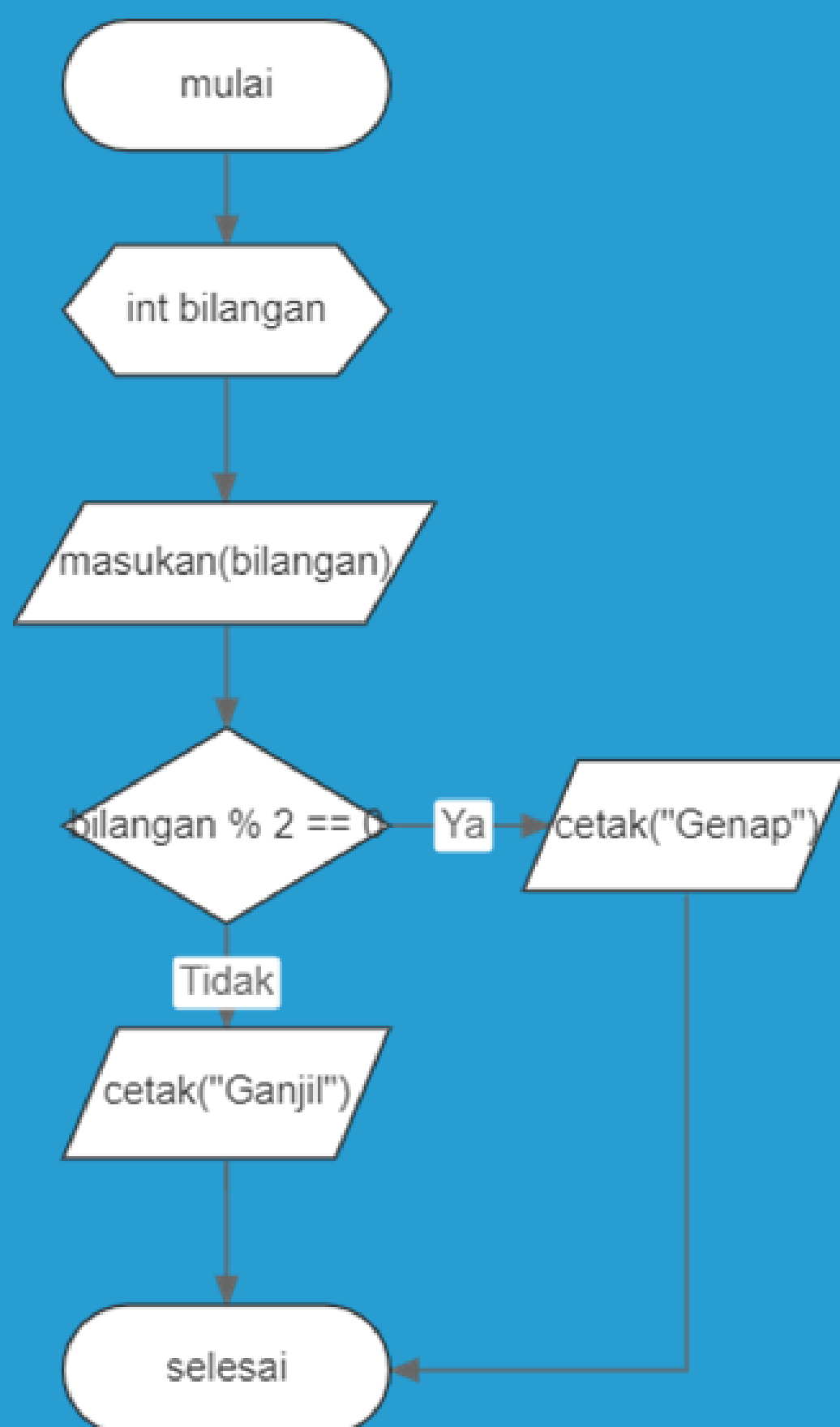
Pertemuan sebelumnya kita mempelajari mengenai percabangan satu kondisi atau percabangan tunggal. Percabangan tunggal yang kemarin hanya memfasilitasi satu kondisi saja dan tidak ada kondisi lainnya. Maka dari itu diperlukan suatu alternatif untuk percabangan tunggal agar dapat memfasilitasi lebih dari satu kondisi. Solusinya adalah dengan menggunakan percabangan dua kasus, tiga kasus atau lebih, dan percabangan switch.

Misalkan ada suatu kasus. Doni ingin pergi ke kampus. Jika cuaca hujan maka doni akan memilih menggunakan mobil sedangkan jika cuaca tidak hujan maka doni akan memilih menggunakan motor. Kasus tersebut merupakan contoh dari percabangan dua kasus.

Percabangan Dua Kasus

Percabangan Dua kasus sering juga disebut percabangan dua kondisi (if-else). Blok program if-else dipergunakan untuk menyatakan percabangan dua kondisi di mana ada dua blok aksi yang dipilih untuk dikerjakan jika syarat kondisi aksi terpenuhi. Saat pembacaan program sampai blok if else maka akan dilakukan pemeriksaan syarat kondisi percabangan yang ada pada deklarasi if. Jika syarat terpenuhi maka yang akan dijalankan adalah aksi pada blok if tetapi jika syarat tidak terpenuhi maka yang dijalankan adalah blok else.

Bentuk flowchart dari percabangan dua kasus (if-else) untuk kasus pengecekan bilangan ganjil atau genap dengan inputan bilangan dari user.



Gambar 1 Flowchart Percabangan Dua Kasus untuk Menentukan Bilangan Ganjil/Genap

Ada beberapa variasi penggunaan struktur kontrol keputusan If - Else. Bentuk umum dari pernyataan if else dan contoh penerapannya dalam C++ untuk kasus mengecek bilangan tersebut ganjil atau genap dengan inputan bilangan dari user (Musthofa, 2021).

```
if (kondisi)
<pernyataan>;
else
<pernyataan>;
```

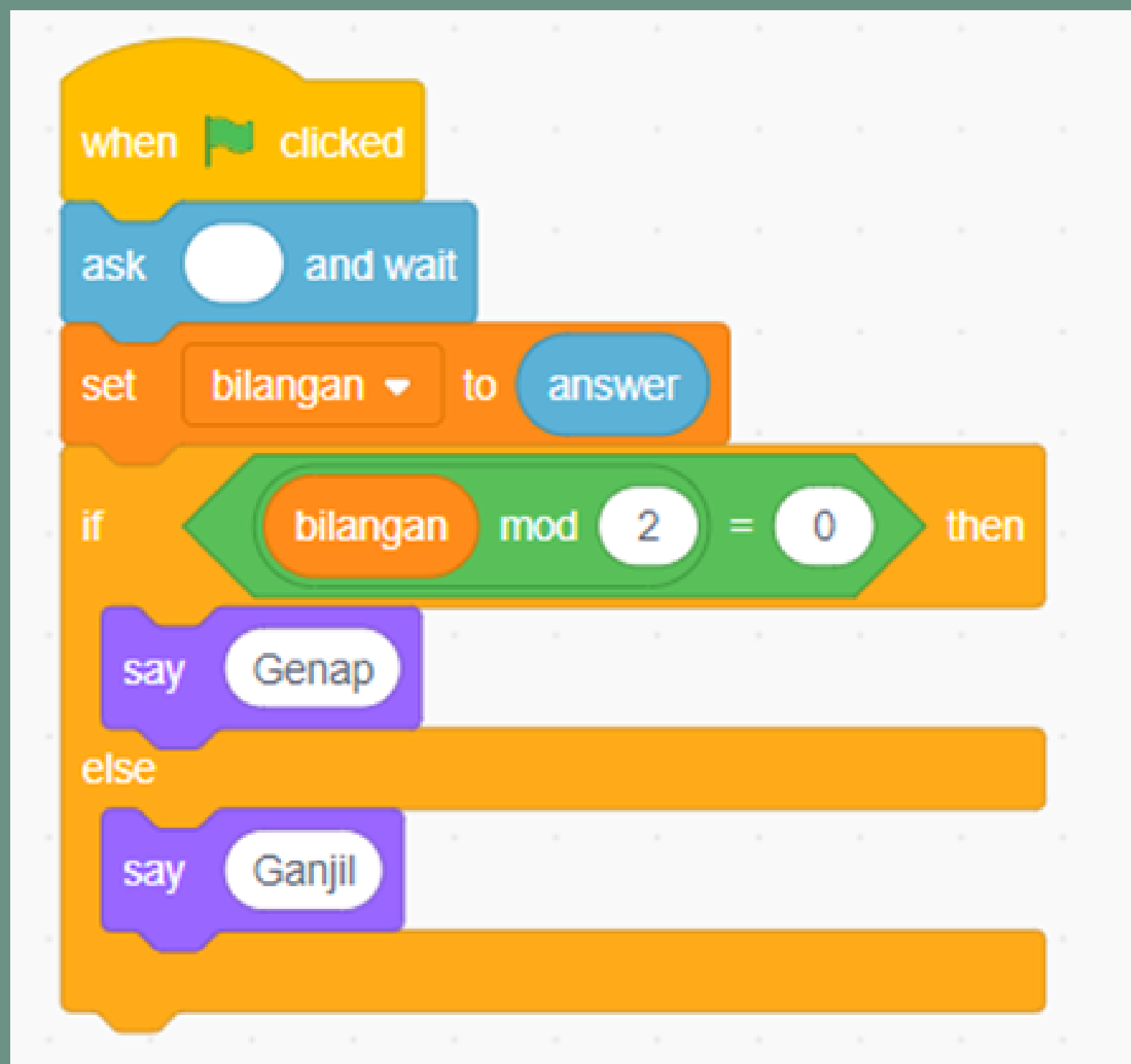
Gambar 2 Bentuk Umum If Else

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int bilangan;
5      cin>> bilangan;
6      if(bilangan % 2 == 0){
7          cout<<"Genap";
8      }else{
9          cout<<"Ganjil";
10     }
11     return 0;
12 }
```

Gambar 3 Code Percabangan 2 Kasus untuk Menentukan Ganjil atau Genap pada C++



jdoodle.com/ia/Nhk



Gambar 4 Code Percabangan 2 Kasus untuk Menentukan Ganjil/Genap pada Scratch

Contoh Soal dan Penyelesaian

Seorang guru ingin membuat program penilaian untuk siswanya. Jika siswa mendapatkan nilai 70 atau di atas 70, mereka akan diberikan predikat "Lulus". Jika nilai siswa di bawah 70, mereka akan diberikan predikat "Tidak Lulus". Inputan berasal dari user. Penamaan variable dibebaskan. Output berupa String "Lulus" atau "Tidak Lulus" tanpa baris baru. Bagaimana percabangan yang sesuai dan bagaimana codenya pada C++ maupun Scratch?

Tahap 1 (Memahami Masalah):

1. Apa saja data yang diketahui dan penting?

Jawab:

.Jika nilai siswa 70 atau di atas 70 mendapat predikat lulus.

.Bahasa pemrograman yang digunakan adalah C++ dan Scratch

.Penamaan variable dibebaskan

.Inputan berasal dari user

.Output berupa string jika siswa mendapat nilai 70 atau di atas 70 maka outputnya adalah "Lulus" Jika tidak maka outputnya adalah "Tidak Lulus" tanpa baris baru

2. Apakah data yang diberikan sudah cukup untuk menyelesaikan masalah?

Jawab: Sudah cukup, karena dengan data yang diberikan kita dapat menyelesaikan masalah. Data sudah dapat menentukan konsep pemrograman apa yang dipakai dan sudah dapat menentukan pembuatan flowchart dan code program

3. Bagaimana jika data yang diberikan diubah sehingga jika siswa mendapatkan nilai dibawah 70 maka tidak ada aksi yang dilakukan, apakah sudah cukup untuk menentukan percabangan yang digunakan adalah percabangan tunggal?

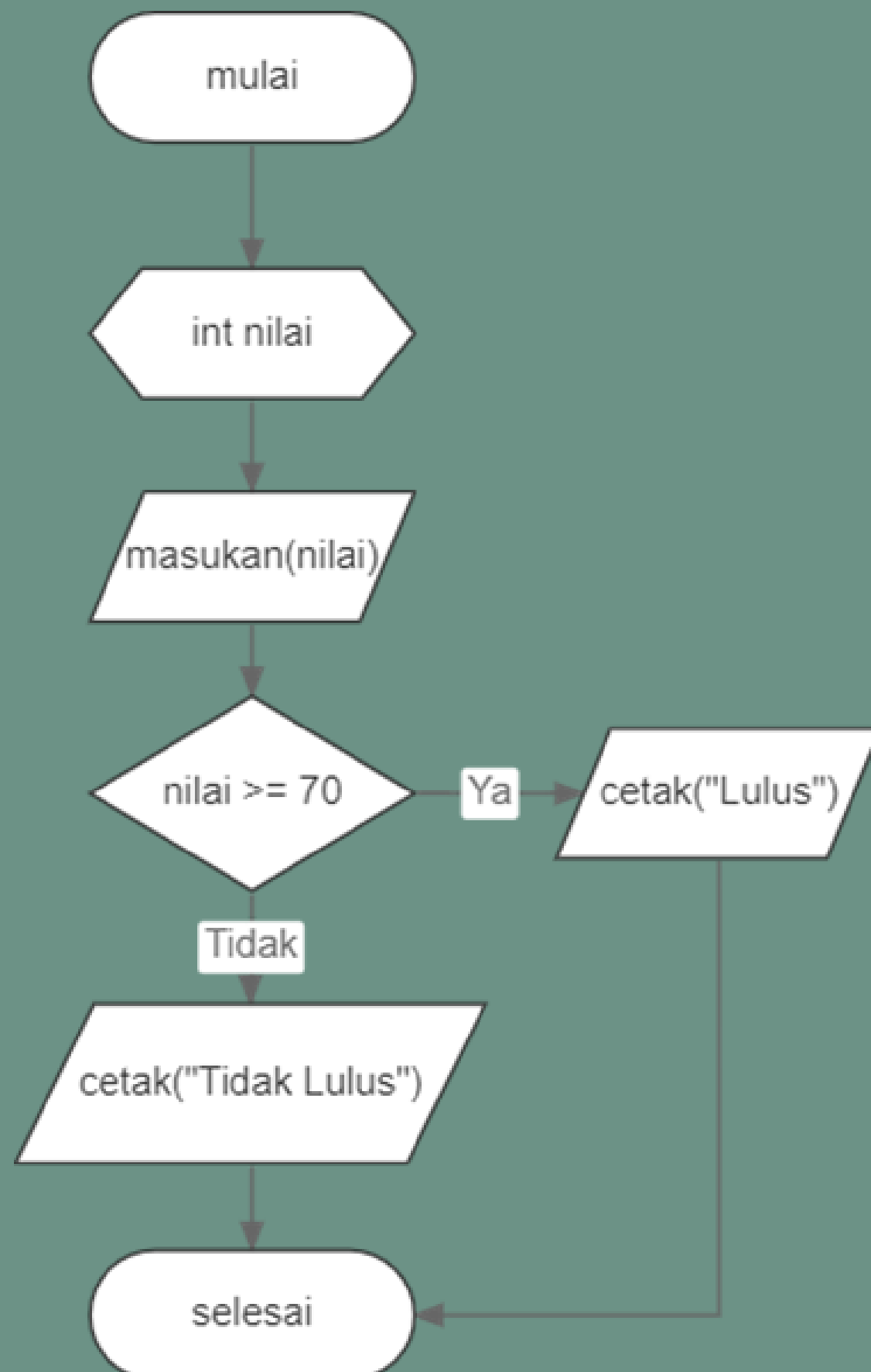
Sudah cukup, karena cukup 1 kondisi saja.

Tahap 2 (Membuat Rencana Penyelesaian Masalah):

1. Kira-kira kondisi apakah yang mirip dengan kasus tersebut?

kondisi yang mirip adalah percabangan dua kondisi/dua kasus, karena hanya ada dua kondisi yaitu ketika siswa mendapat nilai minimal 70 atau di bawah 70.

2. Bagaimana Gambaran Flowchartnya



Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

1. Bagaimana code untuk kasus tersebut?

Berikut ini adalah code untuk kasus tersebut.

Scratch	C++
	<pre>1 #include <iostream> 2 using namespace std; 3 int main(){ 4 int nilai; 5 cin>>nilai; 6 if(nilai >= 70){ 7 cout<<"Lulus"; 8 }else{ 9 cout<<"Tidak lulus"; 10 } 11 return 0; 12 }</pre>

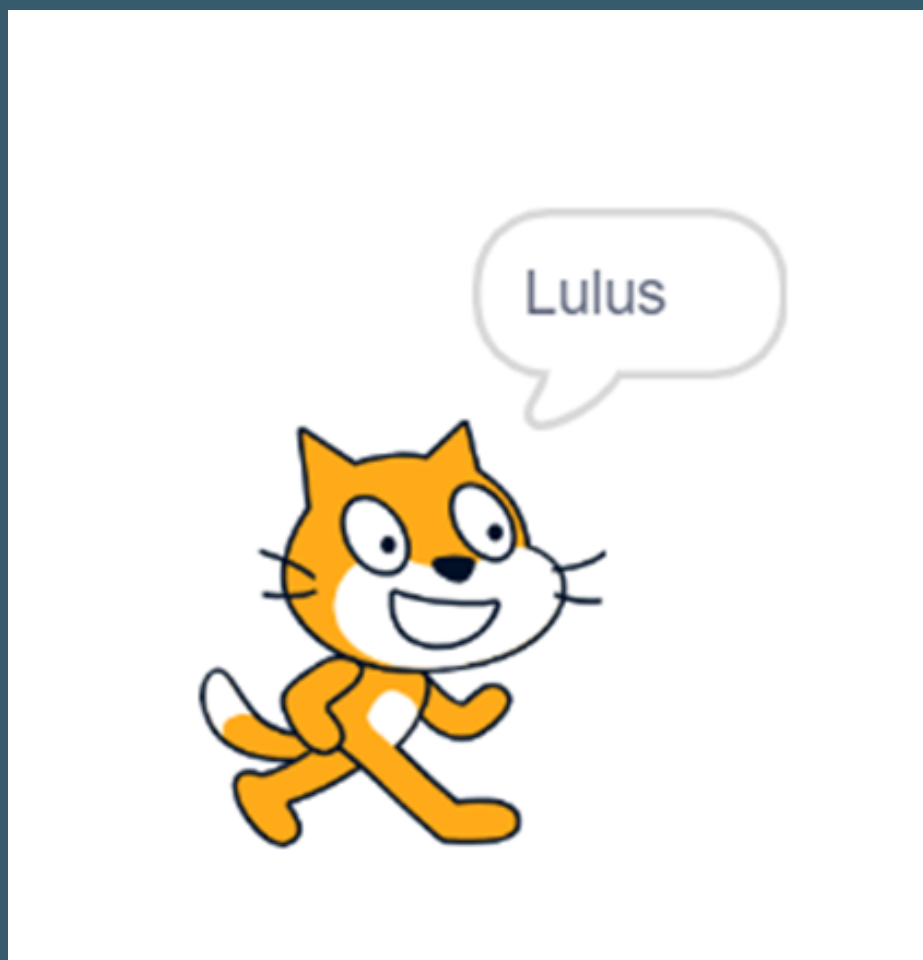
Tahap 4 (Memeriksa Kembali):

1. Apakah code yang dibuat sudah tepat dan menghasilkan output yang benar?

Kode tersebut sudah benar. Ketika dimasukan angka minimal 70 menghasilkan output Lulus sedangkan ketika dimasukan angka kurang dari 70 menghasilkan output Tidak Lulus.

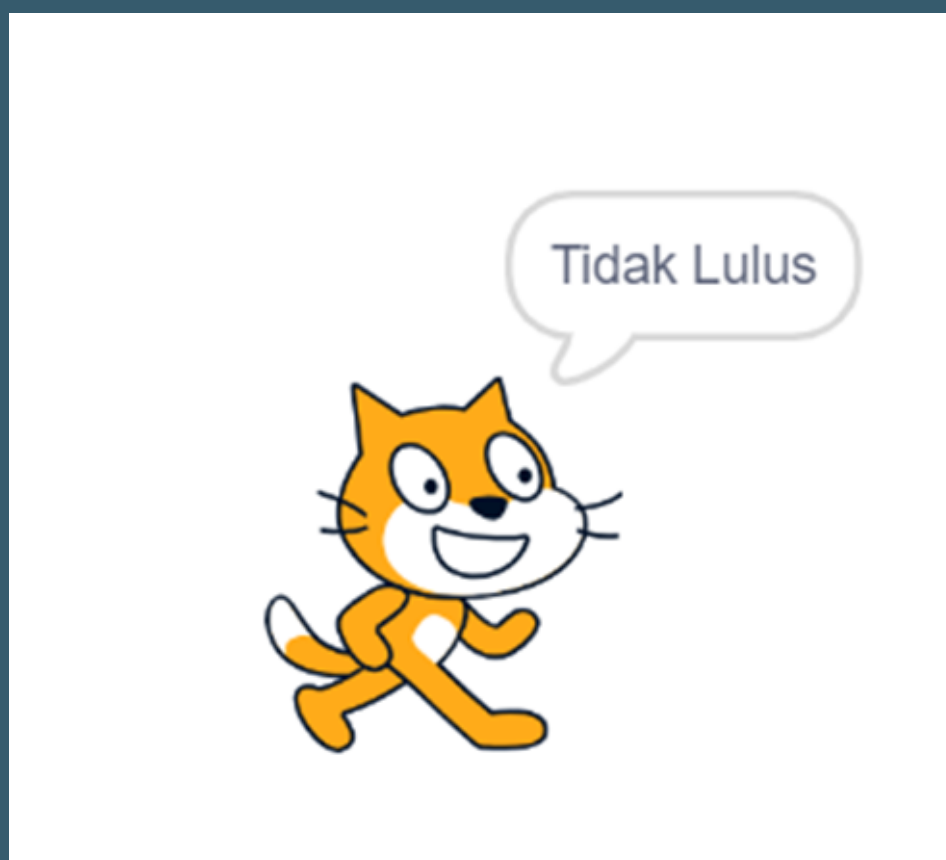
Ketika dimasukan angka 70

```
70  
Lulus
```



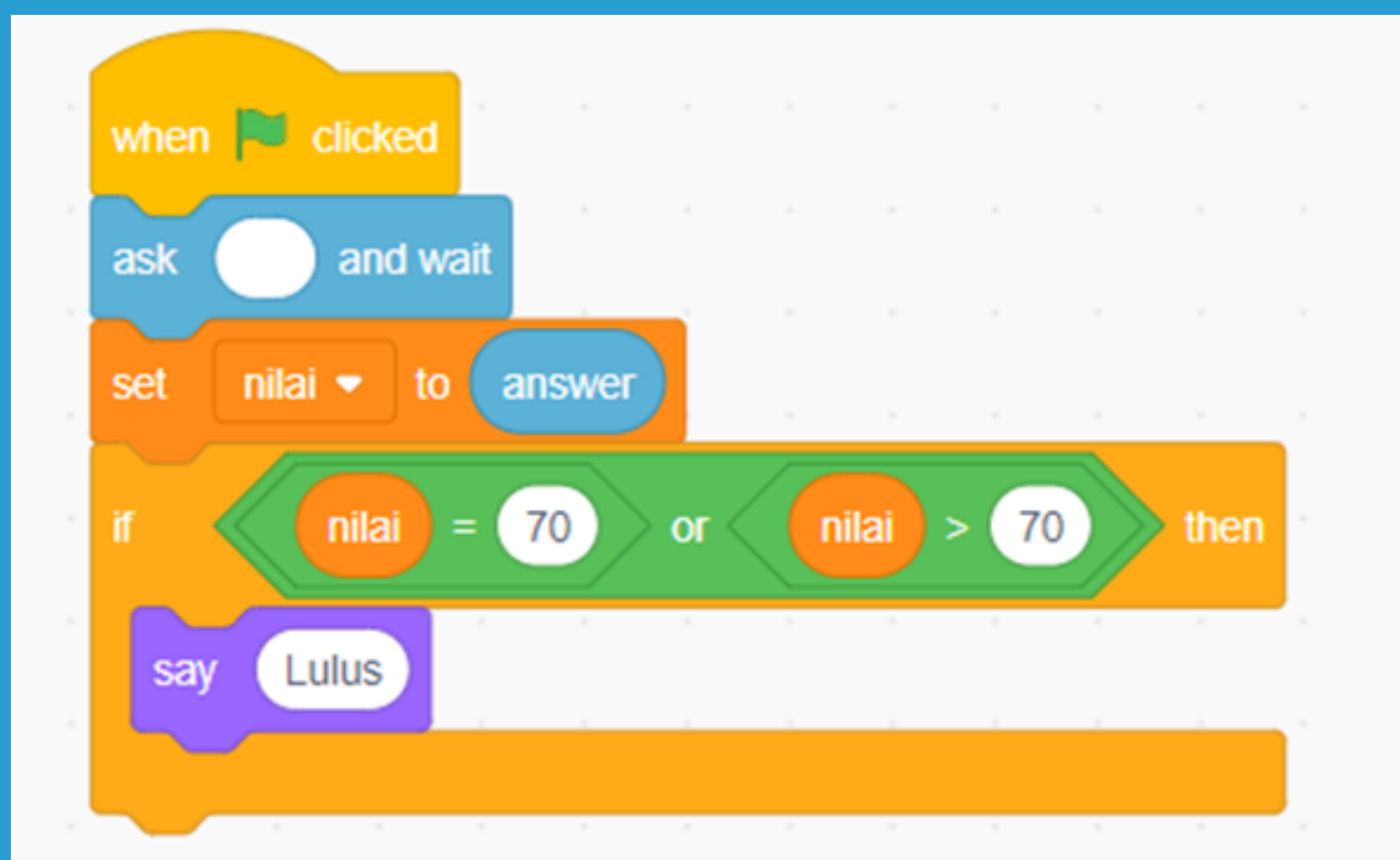
Ketika dimasukan angka 60

```
60  
Tidak lulus
```



2. Bagaimana agar jika nilai lebih kecil dari 70 maka tidak dilakukan aksi caranya adalah dengan menghapus aksi alternatif atau elsenya dan mengubah menjadi percabangan tunggal

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int nilai;
5      cin>>nilai;
6      if(nilai >= 70){
7          cout<<"Lulus";
8
9      return 0;
10 }
```



Percabangan Tiga Kasus/ Lebih

Jika semakin kompleks, struktur if else dapat dikembangkan (Musthofa, 2021). Kondisi ini biasanya dibutuhkan ketika kondisi percabangan untuk banyak syarat dan hanya ingin agar program memilih salah satu (Sukamto, 2018).

Contoh percabangan dua kasus tadi digunakan ketika hanya memiliki dua kondisi saja. Pada contoh kasus tadi, kita hanya menguji apakah bilangan tersebut merupakan bilangan ganjil atau bilangan genap Lantas bagaimana jika terdapat tiga atau lebih pilihan misal..

Jika sekarang masih jam 9 atau kurang dari jam 9 maka akan tampil pada running text “Selamat Pagi”. Ketika jam sudah di atas jam 9 sampai jam 3 sore maka tulisan yang tampil pada running text adalah “Selamat Siang” dan Jika sudah di atas jam 3 sore maka tampil “Selamat Malam”



Gambar 5. Contoh flowchart percabangan tiga kasus

```

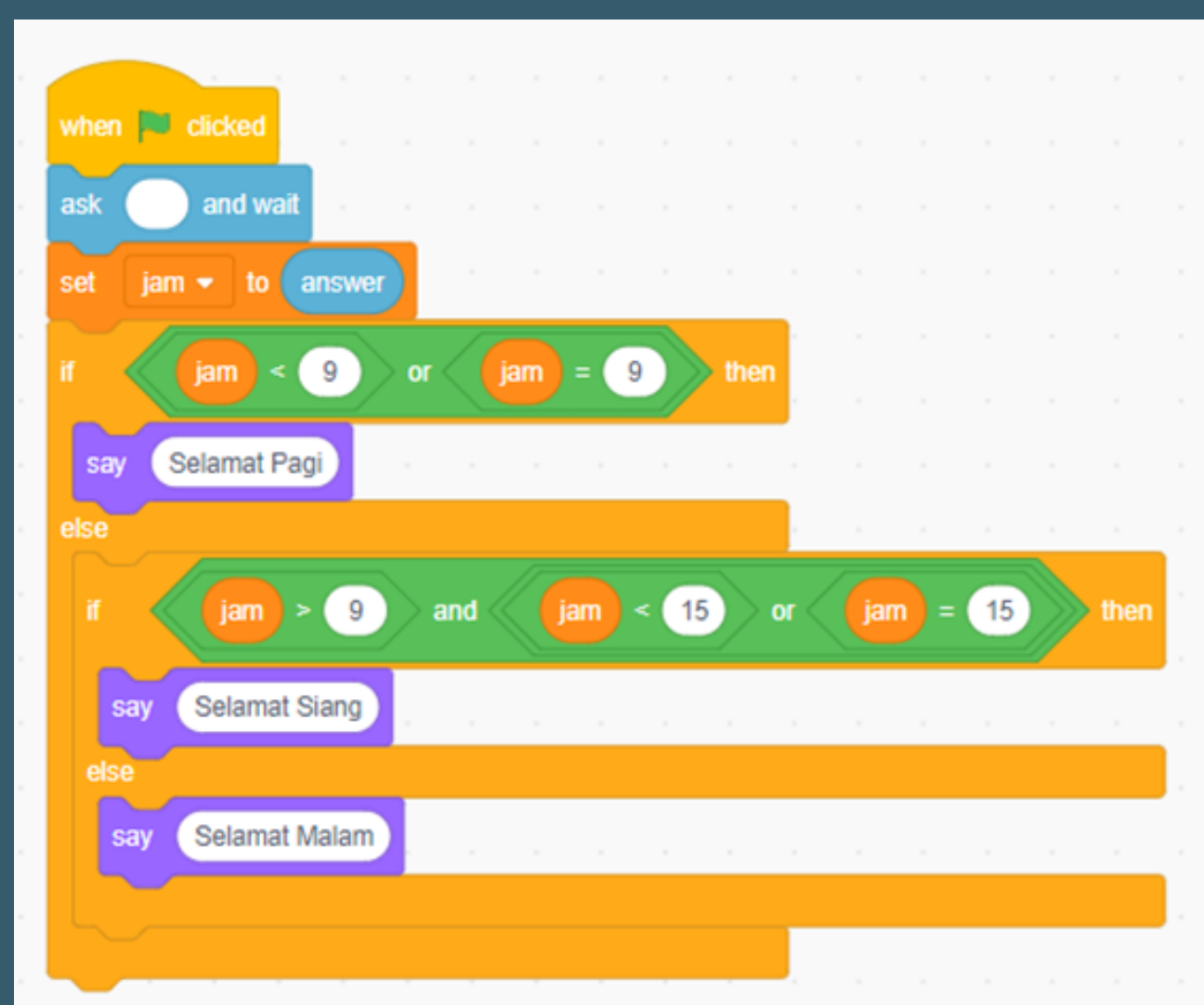
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int jam;
7      cin>>jam;
8      if(jam <= 9){
9          cout<<"Selamat Pagi";
10     }else if(jam > 9 && jam <=15){
11         cout<<"Selamat siang";
12     }else{
13         cout<<"Selamat malam";
14     }
15 }

```

Gambar 6. Code Percabangan 3 Kasus untuk Menentukan Kondisi berdasarkan Jam pada C++



jdoodle.com/ia/NKb



Gambar 7. Code Percabangan 3 Kasus untuk Menentukan Kondisi berdasarkan Jam pada Scratch

Contoh Soal dan Penyelesaian

Seorang siswa ingin membeli buku. Siswa tersebut mengetahui bahwa ada tiga toko buku di kotanya, yaitu Toko Buku A, Toko Buku B, dan Toko Buku C. Siswa tersebut juga mengetahui bahwa harga buku di ketiga toko buku tersebut berbeda-beda.

Siswa tersebut ingin membeli buku yang harganya paling murah. Siswa tersebut tidak mengetahui harga buku di masing-masing toko buku. Program akan menginputkan data harga di setiap toko. Tipe data harganya adalah integer. Penamaan variable dibebaskan. Hasil outputnya berupa string tanpa new line menampilkan "Toko A paling murah" jika toko A paling murah dan seterusnya. Jika ada harga yang sama jadikan salah satu sebagai patokan anda. Apa jenis percabangan yang digunakan dan bagaimana implementasinya dalam C++?

Tahap 1 (Memahami Masalah)

1. Apakah saja data yang diketahui yang penting?

- . Data harga setiap toko buku berbeda, ada 3 data harga yang berbeda, ingin mencari toko termurah.
- . Inputan harga oleh user
- . Penamaan variable dibebaskan
- . Output berupa Toko yang paling murah

2. Apakah data yang diberikan sudah cukup untuk menyelesaikan masalah?

Sudah cukup, Karena dari data yang diberikan sudah bisa menentukan konsep pemrograman apa yang digunakan, dapat menentukan pembuatan flowchart dan code program

3. Bila diberi tahu bahwa hanya ada 2 harga saja yang dibandingkan dan menghasilkan output misalnya "Toko A paling murah" atau "Toko B paling murah" , apakah dapat menentukan bahwa percabangan yang digunakan adalah percabangan tunggal?

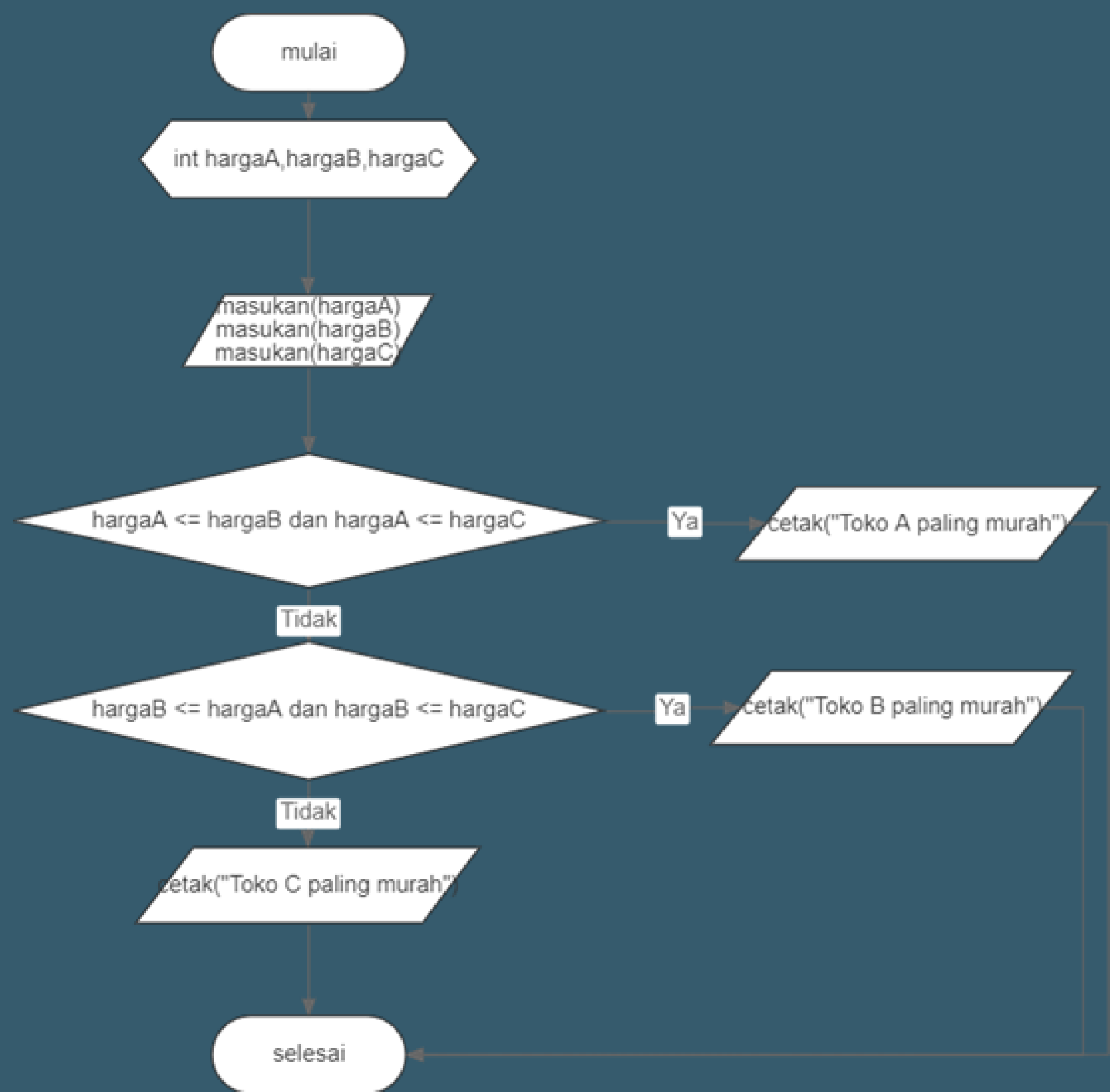
Berlebihan, karena terdapat dua kondisi pada kasus ini.

Tahap 2 (Membuat Rencana Penyelesaian Masalah):

1. Kira-kira kondisi apakah yang mirip dengan kasus tersebut?

Jawab: kondisi yang mirip adalah percabangan tiga kondisi/tiga kasus, karena hanya ada tiga aksi saja yang dilakukan

2. Bagaimana flowchartnya?



Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

1. Bagaimana code untuk kasus tersebut?

Jawab:

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int hargaA, hargaB, hargaC;
5      cin>>hargaA;
6      cin>>hargaB;
7      cin>>hargaC;
8      if(hargaA <= hargaB && hargaA <= hargaC){
9          cout<<"Toko A paling murah";
10     }else if(hargaB <= hargaA && hargaB <= hargaC){
11         cout<<"Toko B paling murah";
12     }else{
13         cout<<"Toko C paling murah";
14     }
15     return 0;
16 }
```


Tahap 4 (Memeriksa kembali)

1. Apakah kode tersebut sudah menghasilkan output yang sesuai?
- Kondisi if yang pertama menandakan bahwa harga tokoA paling murah

• Kondisi else if menandakan bahwa harga tokoB paling murah

• Kondisi else menandakan bahwa harga tokoC paling murah

Check outputnya

1. Jika Masukan harga toko A paling murah

```
10000
20000
30000
Toko A paling murah
-----
```

2. Jika masukan harga toko B paling murah

```
15000
10000
20000
Toko B paling murah
-----
```

3. Jika masukan harga toko C paling murah

```
15000
20000
10000
Toko C paling murah
-----
```

2. Bagaimana kode program untuk mengecek harga toko yang paling mahal

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4
5      int hargaA, hargaB, hargaC;
6      cin >> hargaA;
7      cin >> hargaB;
8      cin >> hargaC;
9      if(hargaA >= hargaB && hargaA >= hargaC){
10         cout << "Harga A paling mahal";
11     }else if(hargaB >= hargaA && hargaB >= hargaC){
12         cout << "Harga B paling mahal";
13     }else{
14         cout << "Harga C paling mahal";
15     }
16     return 0;
17 }
```

Dengan mengubah kondisi yang sebelumnya <= menjadi >= dan mengubah outputnya.

Switch

Percabangan depend on biasanya digunakan untuk dua kondisi atau lebih bergantung pada nilai sebuah variable. Percabangan switch ini dapat menyederhanakan penulisan IF-THEN-ELSE yang bertingkat-tingkat (Munir & Lidya, 2016). Syarat kondisi pada percabangan ini biasanya hanya sebuah nilai. Jika nilai yang diperiksa memenuhi syarat dari nilai yang didefinisikan maka akan dikerjakan prosesnya. Tipe data yang bisa dibandingkan dalam switch C++ adalah int dan char. (Sukamto, 2018)

If then memang bisa digunakan untuk memilih. Namun jika kondisi yang ingin diuji begitu banyak menyebabkan baris-baris pada script begitu melimpah. Maka dari itu dapat digunakan Switch case (Jubilee Enterprise, 2017). Namun tidak semua bahasa pemrograman mendukung switch ini (Munir & Lidya, 2016). Konstruksi Case memeriksa nilai dari evaluasi ekspresi tersebut sama dengan salah satu dari case 1, case 2, dll. Dapat disimpulkan bahwa konstruksi else dapat menyederhanakan if-else yang bertumpuk-tumpuk (Percabangan tiga kasus/lebih)

Contoh Kasus Percabangan Dua Kasus yang diubah ke dalam bentuk Switch untuk menentukan bilangan ganjil atau genap.

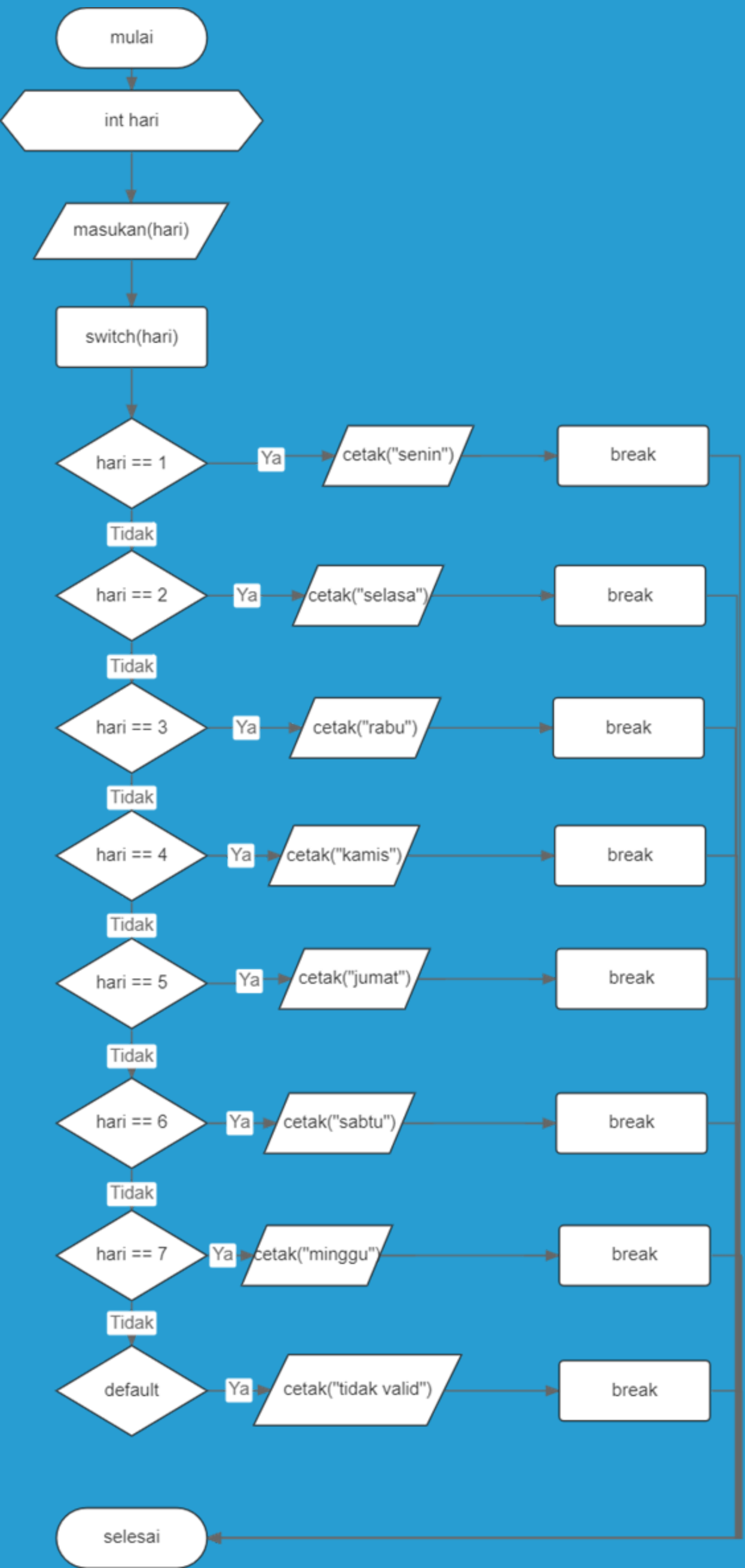
```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int bilangan;
6
7      cin >> bilangan;
8      switch (bilangan % 2) {
9          case 0:
10             cout << "Genap." << endl;
11             break;
12         default:
13             cout << "Ganjil." << endl;
14     }
15     return 0;
16 }
```

Gambar 8. Code Percabangan Switch untuk Kasus Genap/Ganjil pada C++

Contoh Kasus lain untuk Percabangan Tiga Kasus untuk menentukan suatu inputan adalah angka 1, angka 0, atau bukan 1 dan bukan 0


```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int bilangan;
6
7     cin >> bilangan;
8     switch (bilangan) {
9         case 0:
10             cout << "Nol" << endl;
11             break;
12         case 1:
13             cout<< "Satu" <<endl;
14             break;
15         default:
16             cout << "Bukan Satu Bukan Nol" << endl;
17     }
18     return 0;
19 }
```

Gambar 9. Code Percabangan Switch untuk Kasus untuk Menentukan Suatu Bilangan itu Nol, Satu, atau Bukan Keduanya



Gambar 10 Flowchart Switch untuk Menentukan Hari

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int nomorHari;
6
7      cout << "Masukkan nomor hari (1-7): ";
8      cin >> nomorHari;
9
10     switch (nomorHari) {
11         case 1:
12             cout << "Senin" ;
13             break;
14         case 2:
15             cout << "Selasa" ;
16             break;
17         case 3:
18             cout << "Rabu" ;
19             break;
20         case 4:
21             cout << "Kamis" ;
22             break;
23         case 5:
24             cout << "Jumat" ;
25             break;
26         case 6:
27             cout << "Sabtu" ;
28             break;
29         case 7:
30             cout << "Minggu" ;
31             break;
32         default:
33             cout << "Nomor hari tidak valid. Masukkan nomor antara 1 hingga 7." ;
34             break;
35     }
36
37     return 0;
38 }
```

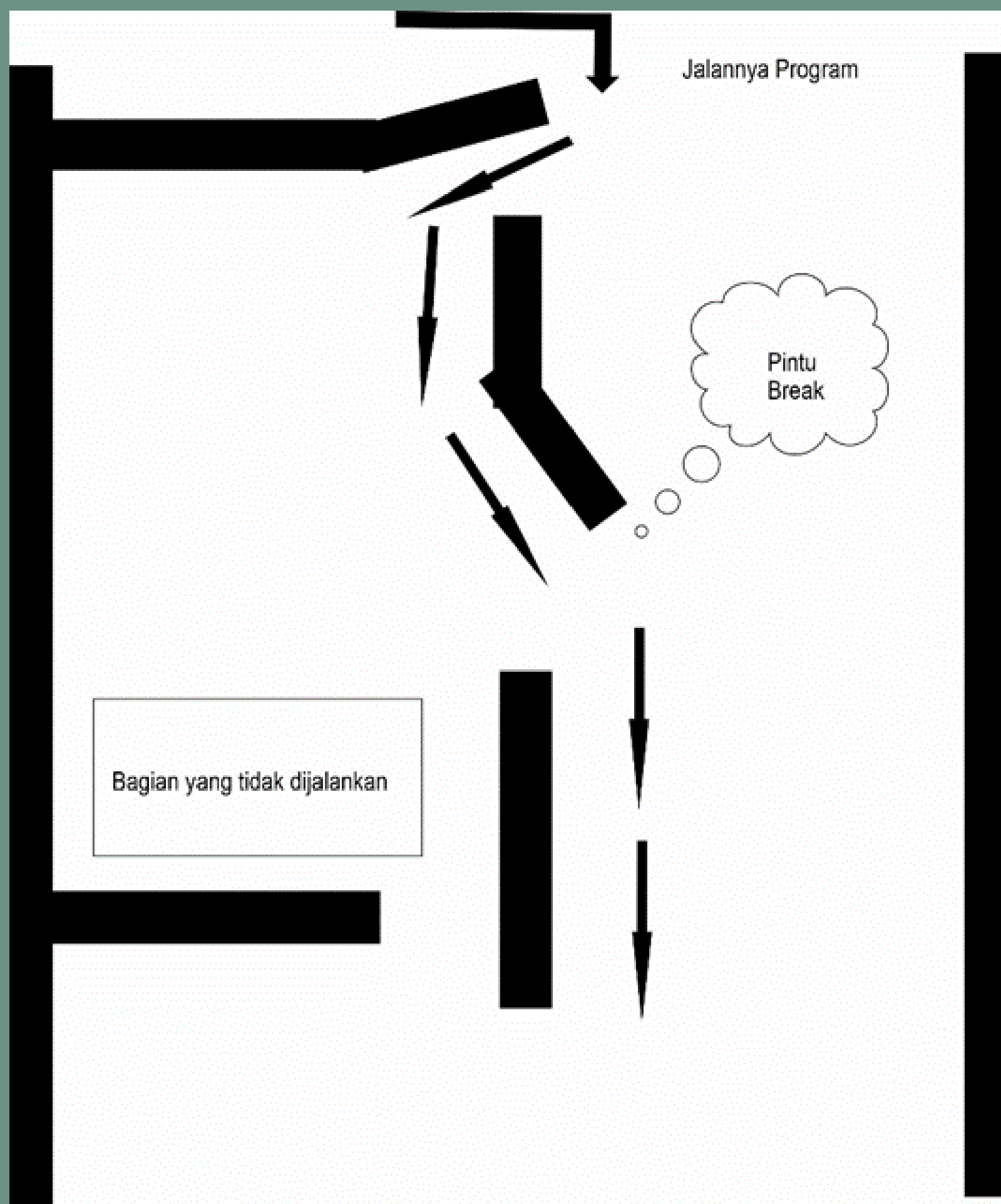
Gambar 11 Code Switch untuk Menentukan Hari pada C++

Break

```
case 5:
    cout << "Jumat" << endl;
    break;
case 6:
    cout << "Sabtu" << endl;
    break;
case 7:
    cout << "Minggu" << endl;
    break;
```

Gambar 12 Break di Switch Pada C++

Break biasanya digunakan untuk keluar dari sebuah blok program tanpa mengerjakan semua aksi yang ada setelah break (Sukamto, 2018).



Gambar 13 Gambaran Break

Sebetulnya untuk kondisi terakhir tidak memerlukan break. Jadi untuk kondisi terakhir baik dengan break ataupun tidak tidak menjadi masalah.

Contoh:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int day = 4;
6      switch (day) {
7          case 6:
8              cout << "Today is Saturday";
9              break;
10         case 7:
11             cout << "Today is Sunday";
12             break;
13         default:
14             cout << "Looking forward to the Weekend";
15     }
16     return 0;
17 }
```

Gambar 14 code tanpa break



jdoodle.com/ia/Nh3

Contoh Soal dan Penyelesaian:

Pengkodean suatu hari ditentukan berdasarkan nomor 1 sampai 7. Jika yang dimasukan adalah 1 maka hari nya adalah "senin" tanpa new line . Jika yang dimasukan adalah 2 maka harinya adalah "selasa" tanpa new line dan seterusnya. Jika angka yang dimasukan tidak sesuai akan menampilkan "Nomor tidak valid masukan nomor 1-7". Nama variable dibebaskan. Hari diinputkan oleh user. Buatlah kode percabangan tersebut pada bahasa C++

Tahap 1 (Memahami Masalah):

1. Apakah saja data yang diketahui dan penting?

- . Hari ditentukan berdasarkan kode angka dari hari 1-7
- . Hari diinputkan oleh user
- . Penamaan variable dibebaskan
- . Menghasilkan output nama hari berdasarkan kode angka.

2. Apakah data yang diberikan sudah cukup untuk menyelesaikan permasalahan?

Sudah cukup, Data-data yang diberikan sudah cukup untuk menyelesaikan permasalahan. Data tersebut sudah cukup menentukan konsep pemrograman apa yang dipergunakan dan sudah cukup untuk membuat flowchart dan code program

3. Bagaimana jika pada soal tidak diberitahukan bahasa pemrogramannya apakah kita dapat menentukan percabangan yang dipergunakan adalah percabangan switch?

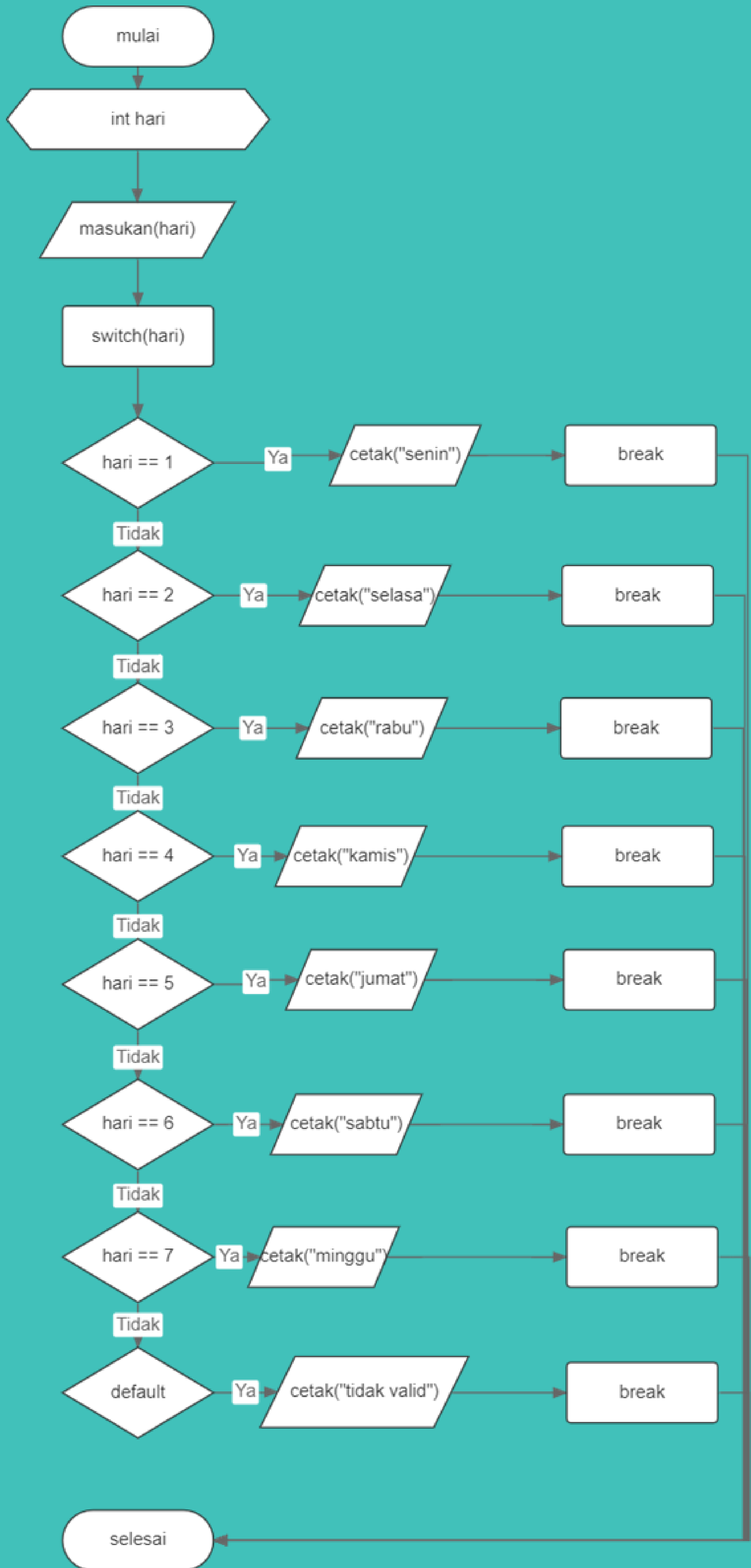
Jawab: Jika pada soal tidak diberitahukan maka data yang diberikan tidak cukup untuk menjawab bahwa percabangan yang dipergunakan adalah percabangan switch karena tidak semua bahasa pemrograman memiliki switch.

Tahap 2 (Membuat Rencana Penyelesaian Masalah):

1. Kira-kira kondisi apakah yang mirip dengan kasus tersebut?

Jawab: kondisi yang mirip adalah percabangan switch karena lebih mudah dan efektif

2. Bagaimana flowchart diagramnya?



Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

1. Bagaimana codenya dalam C++?

Jawab:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int nomorHari;
6
7      cout << "Masukkan nomor hari (1-7): ";
8      cin >> nomorHari;
9
10     switch (nomorHari) {
11         case 1:
12             cout << "Senin" ;
13             break;
14         case 2:
15             cout << "Selasa" ;
16             break;
17         case 3:
18             cout << "Rabu" ;
19             break;
20         case 4:
21             cout << "Kamis" ;
22             break;
23         case 5:
24             cout << "Jumat" ;
25             break;
26         case 6:
27             cout << "Sabtu" ;
28             break;
29         case 7:
30             cout << "Minggu" ;
31             break;
32         default:
33             cout << "Nomor hari tidak valid. Masukkan nomor antara 1 hingga 7." ;
34             break;
35     }
36
37     return 0;
38 }
```

Tahap 4 (Memeriksa Kembali):

1. Apakah kode tersebut sudah benar dan menghasilkan output dengan benar. Misal kodenya begini ?

Jawab:

Dalam Switch kita harus memeriksa apakah case memiliki break atau tidak. Kurangnya break pada suatu case dapat menyebabkan hasil yang tidak benar. Kemudian cek hasil ketika dimasukan angka-angka.

Ketika dimasukan angka yang sesuai

```
Masukkan nomor hari (1-7): 7
Minggu
```

Ketika dimasukan angka yang tidak sesuai

```
Masukkan nomor hari (1-7): 8
Nomor hari tidak valid. Masukkan nomor antara 1 hingga 7.
```

2. Bagaimana agar ketika dimasukan angka 8 tidak memunculkan apapun Dengan menghapus default

Daftar Pustaka

- Jubilee Enterprise. (2017). Belajar Vb, Visual C#, Dan Python Menggunakan Visual Studio. Jakarta: Elex Media Komputindo.
- Munir, R., & Lidya, L. (2016). Algoritma Dan Pemrograman Dalam Bahasa Pascal, C, C++ Edisi Keenam. Bandung: Informatika.
- Musthofa. (2021). Informatika untuk SMA Kelas X. Jakarta: Pusat Kurikulum dan Perbukuan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi.
- Polya, G. (2015). How to Solve It: A New Aspect of Mathematical Method. New Jersey: Princeton University Press.
- Sukanto, R. A. (2018). Logika Algoritma dan Pemograman Dasar. Bandung: Modula.
- Setiawan, R. (2017). Teknik Pemecahan Masalah dengan Algoritma dan Flowchart (Basic dan C). Jakarta:Lentera Ilmu Cendikia