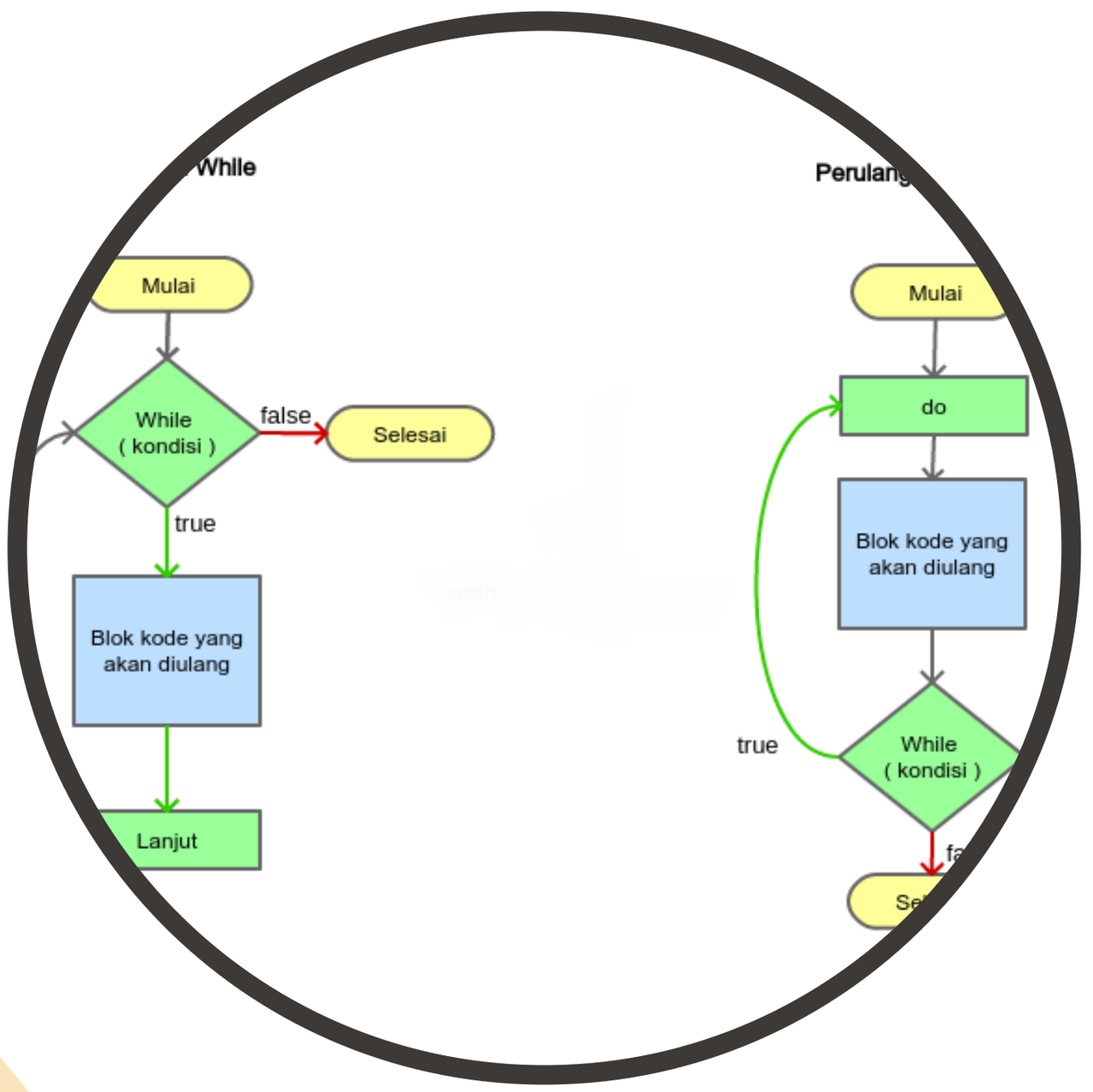
Modul Percabangan

Bagian II



Disusun Oleh :

Johannes Alexander Putra

Pengantar

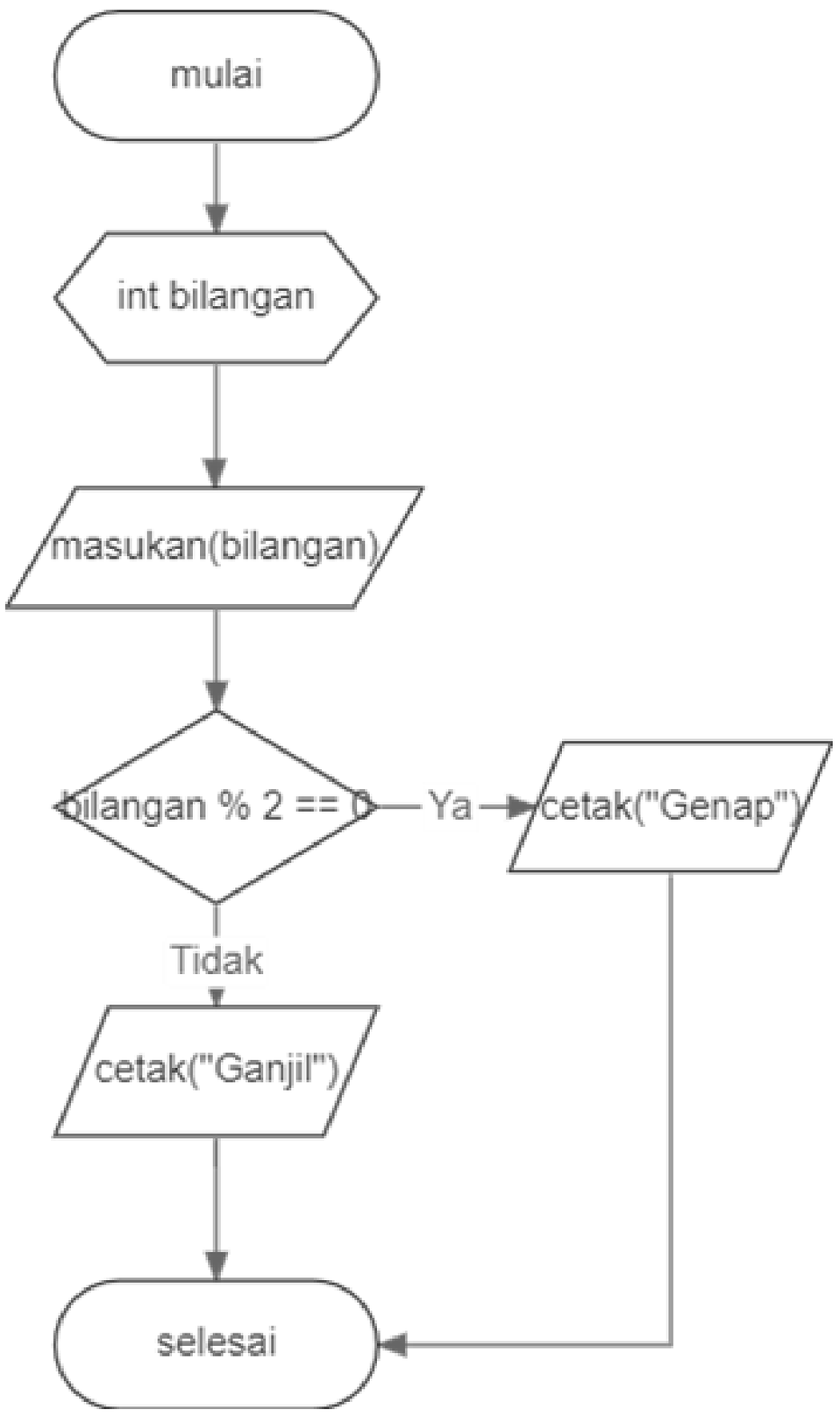
Pertemuan sebelumnya kita mempelajari mengenai percabangan satu kondisi atau percabangan tunggal. Percabangan tunggal yang kemarin hanya memfasilitasi satu kondisi saja dan tidak ada kondisi lainnya. Maka dari itu diperlukan suatu alternatif untuk percabangan tunggal agar dapat memfasilitasi lebih dari 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).

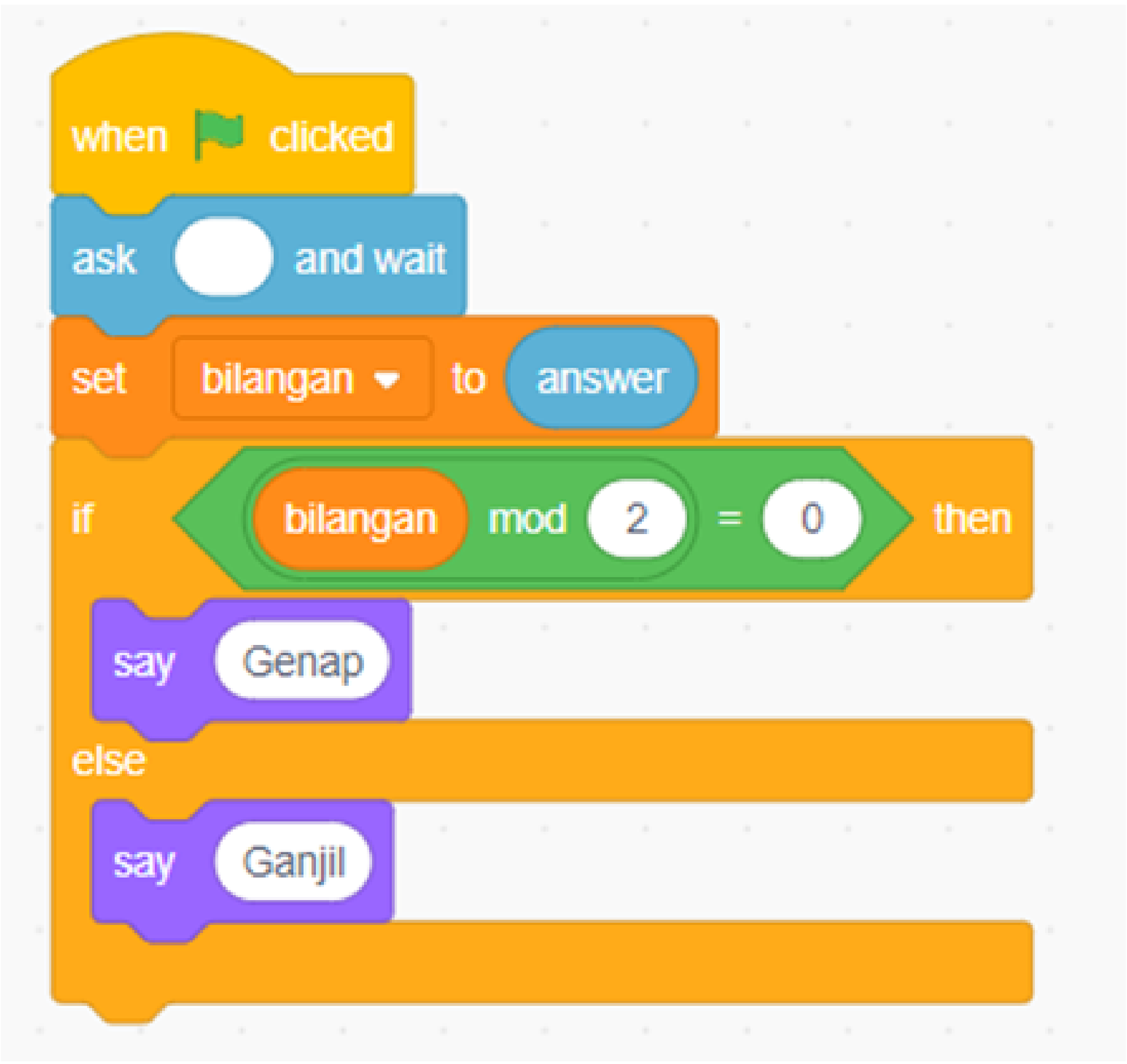
```
   if (kondisi)
<pernyataan>;
   else
<pernyataan>;
```

Gambar 2 Bentuk Umum If Else

```
#include <iostream>
using namespace std;
int main(){
    int bilangan;
    cin>> bilangan;
    if(bilangan % 2 == 0){
        cout<<"Genap";
    }else{
        cout<<"Ganjil";
    }
    return 0;
}
```

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

https://s.id/1Z34W

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

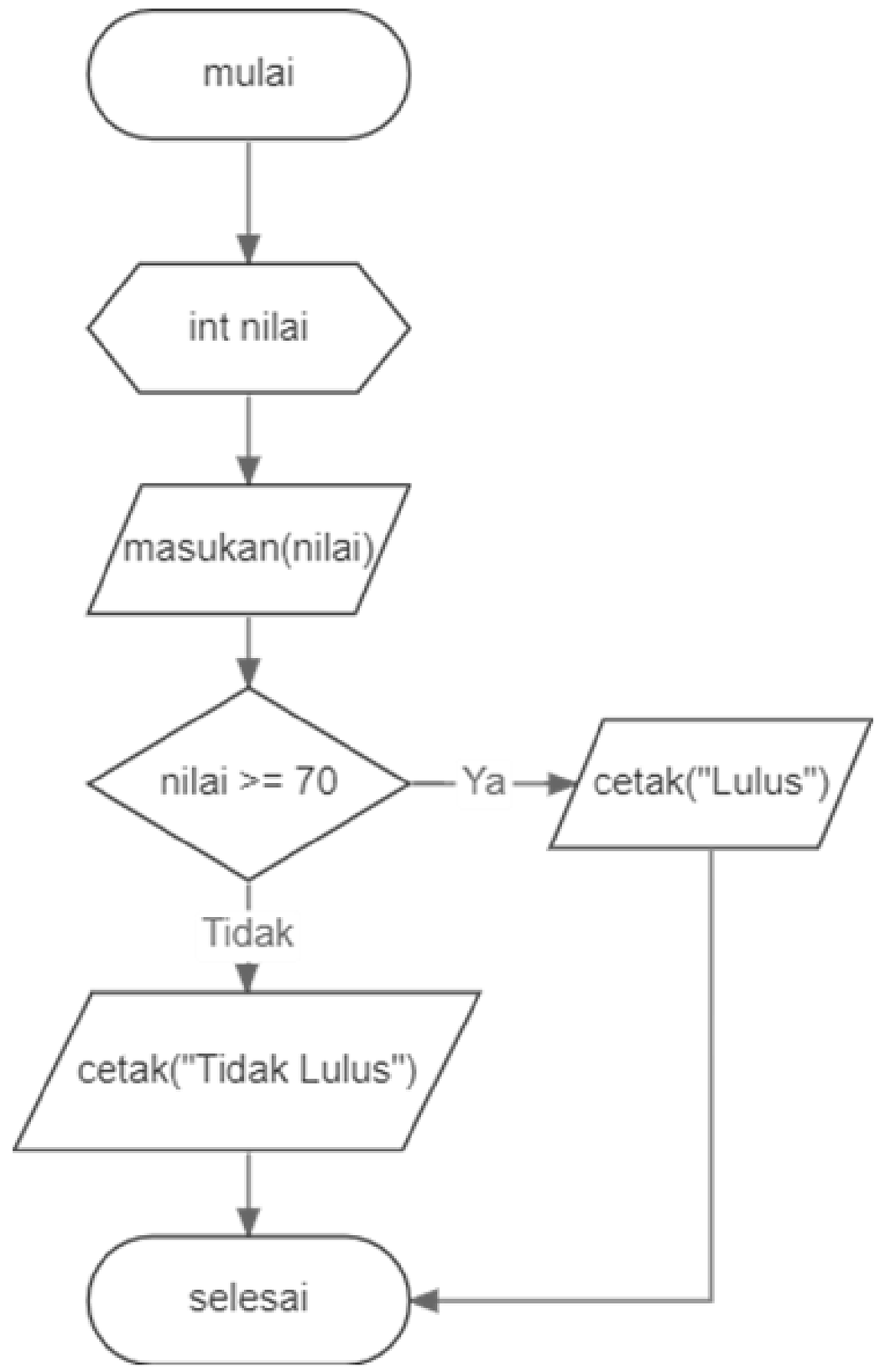
Keterangan:
Kondisi dinyatakan berlebihan bila data yang diberikan berlebihan contohnya, Jika terdapat soal: "Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan dua kasus?", Jika pada soal diberikan 3 buah kondisi dan 3 buah aksi, tentu saja hal ini berlebihan.
Kondisi dinyatakan kontradiktif. Misalnya jika terdapat soal: "Apakah kondisi yang diberikan cukup untuk menjawab bahwa percabangan tersebut adalah percabangan dua kasus? Bila pada soal diberikan informasi: Percabangan dalam kasus ini hanya ada 2 kondisi yaitu jika nilai >= 70 dan jika nilai < 70 . Jika nilai >= 70 maka cetak "Anda lulus" jika bernilai nilai < 70 maka cetak "Anda tidak lulus" Jika bernilai bukan *>= 70* dan bukan *< 70* maka cetak "Anda ditengah-tengah". Tentu saja data yang diberikan kontradiktif karena sebelumnya sudah dikatakan bahwa **hanya ada 2 kondisi** namun dikalimat berikutnya ada kondisi lain.

Kondisi dinyatakan cukup bila data yang diberikan sudah cukup untuk menjawab pertanyaan. Contohnya terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan dua kasus?”, Pada soal diberikan 2 buah kondisi dan dua aksi

Kondisi dinyatakan tidak cukup bila data yang diberikan tidak cukup untuk menjawab pertanyaan. Contoh terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan dua kasus?”, Pada soal diberikan sebuah kondisi saja. Tentu data yang diberikan kurang atau pada soal tersebut tidak diberikan aksi jika kondisi tercapai atau pada soal tidak diberi tahu bahasa pemrogramannya.

Kondisi dinyatakan tidak ada konteks yang tepat bila tidak menggambarkan sama sekali percabangan
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.

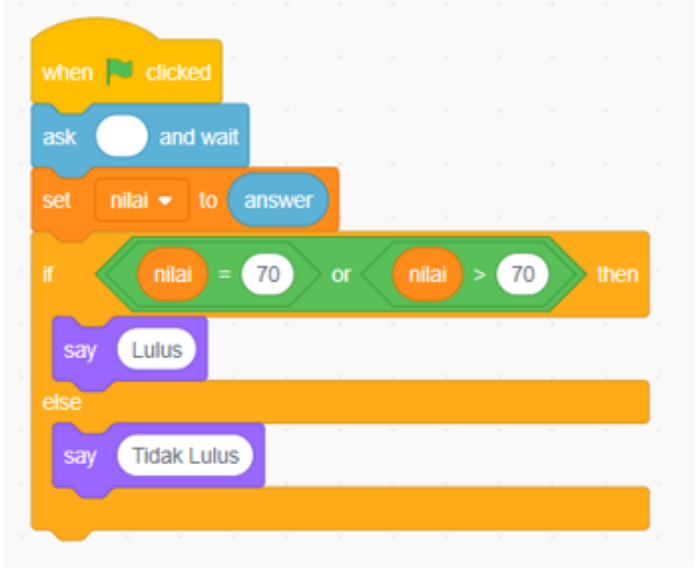
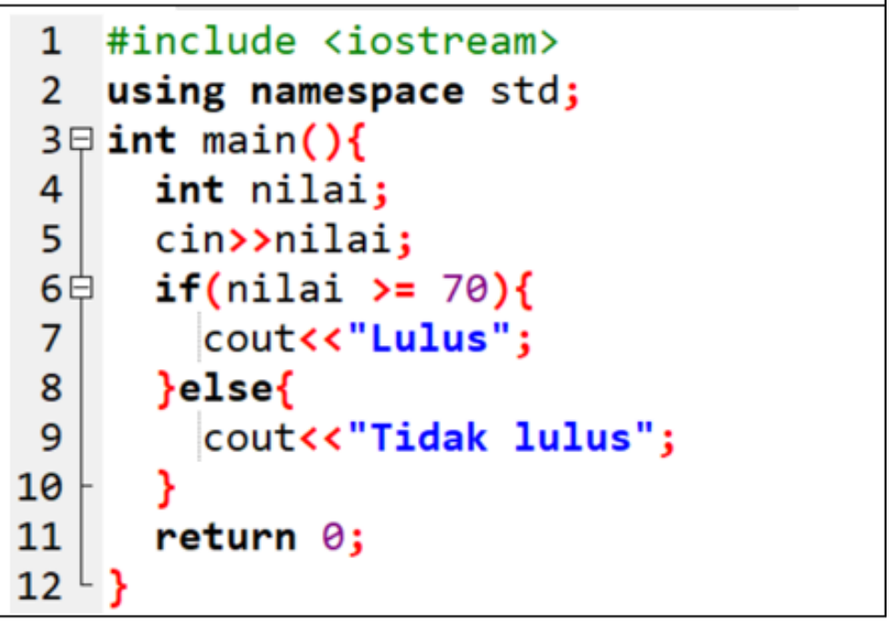
Tahap 2 (Membuat Rencana Penyelesaian Masalah):
1. Kira-kira kondisi apakah yang mirip dengan kasus tersebut?
kondisi yang mirip adalah percabangan dua kondisi/dua kasus, karena hanya ada dua kondisi yaitu ketika siswa mendapat nilai minimal 70 atau di bawah 70.
2. Bagaimana Gambaran Flowchartnya



Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

1. Bagaimana code untuk kasus tersebut?

Berikut ini adalah code untuk kasus tersebut.

Scratch	C++
	


```cpp
#include <iostream>
using namespace std;
int main(){
  int nilai;
  cin>>nilai;
  if(nilai >= 70){
    cout<<"Lulus";
  }else{
    cout<<"Tidak lulus";
  }
  return 0;
}
```

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

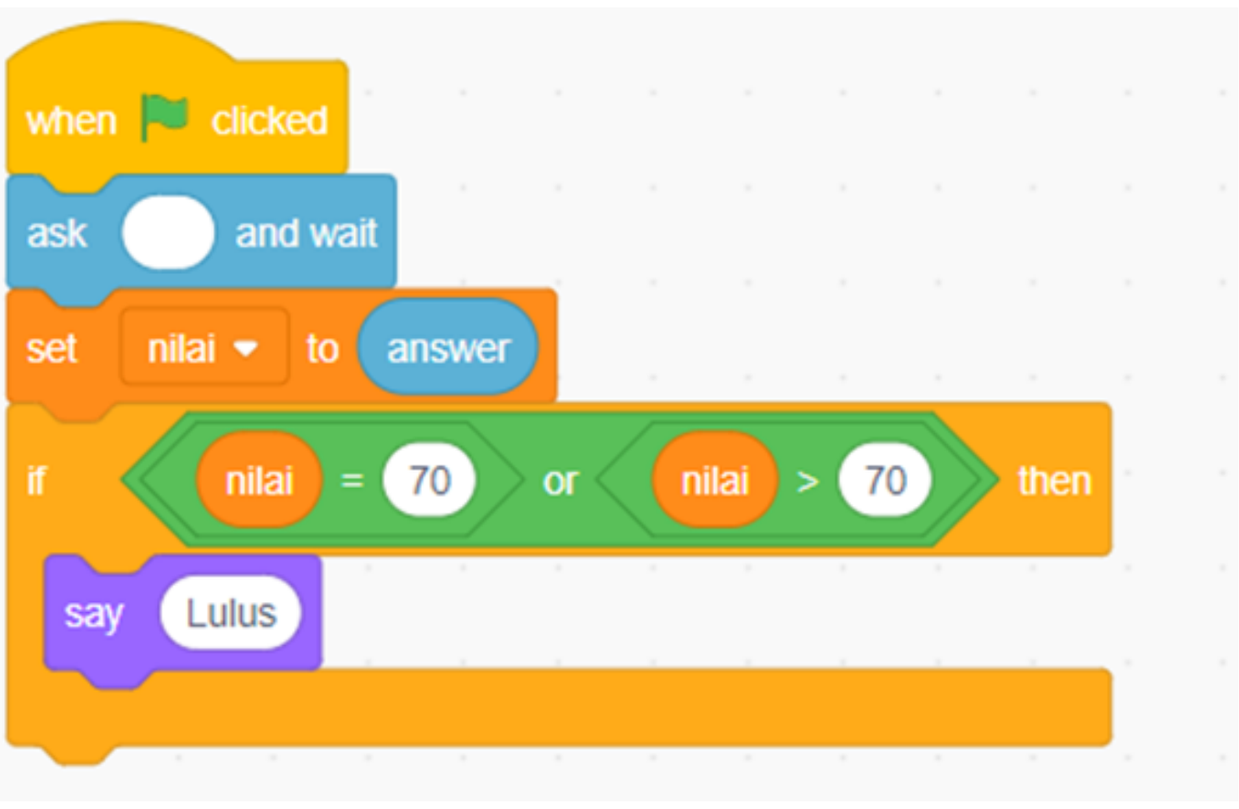
```
70
Lulus
--------------------------------
```

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

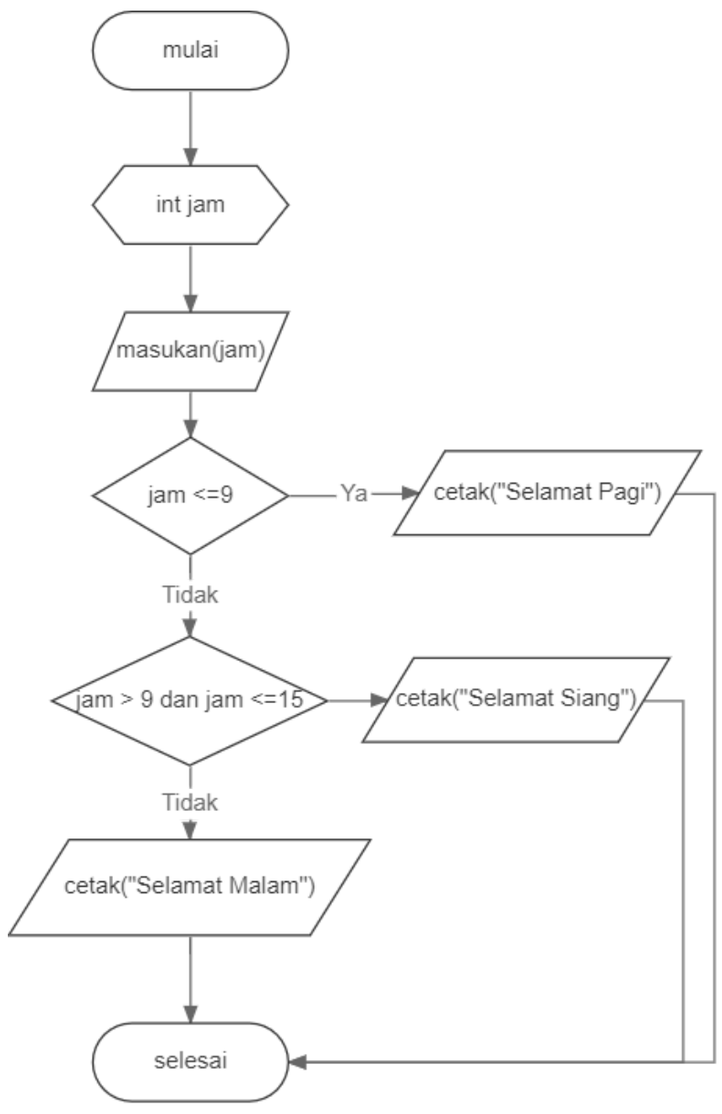
```cpp
#include <iostream>
using namespace std;
int main(){
   int nilai;
   cin>>nilai;
   if(nilai >= 70){
      cout<<"Lulus";

   return 0;
}
```



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

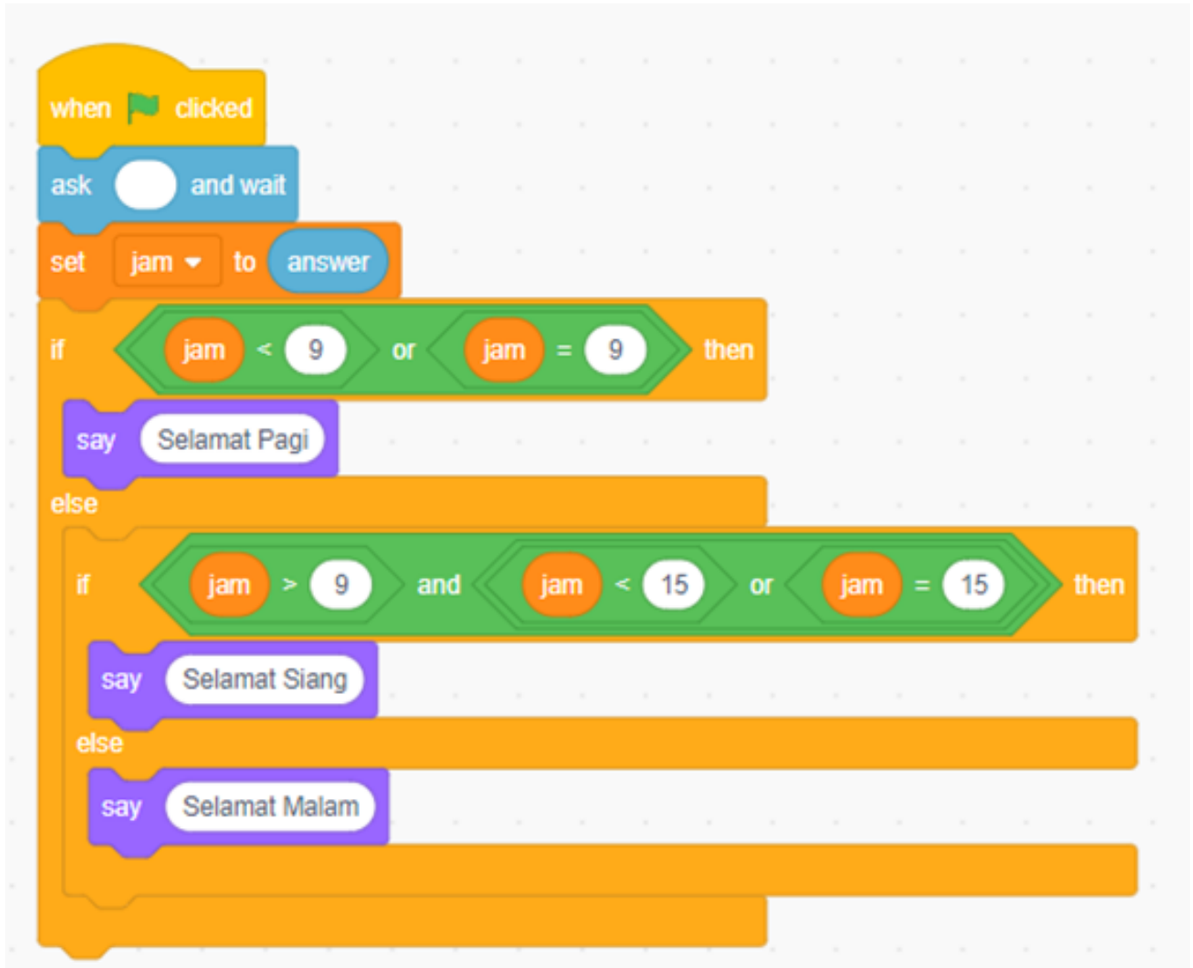
```
if (kondisi ke-1)
     <statement>
else if (kondisi ke-2)
     <statement>
else if (kondisi ke-3)
     <statement>
.....
else if (kondisi ke-n )
     <statement>
else
     <statement>
```

Gambar 6 Struktur Percabangan Tiga Kasus atau lebih

```cpp
#include <iostream>

using namespace std;

int main() {
    int jam;
    cin>>jam;
    if(jam <= 9){
        cout<<"Selamat Pagi";
    }else if(jam > 9 && jam <=15){
        cout<<"Selamat siang";
    }else{
        cout<<"Selamat malam";
    }
}
```

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

jdoodle.com/ia/NKb



Gambar 8. Code Percabangan 3 Kasus untuk Menentukan Kondisi berdasarkan Jam pada Scratch

https://s.id/1Z36d

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

Keterangan:

Kondisi dinyatakan berlebihan bila data yang diberikan berlebihan contohnya, Jika terdapat soal: "Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan tiga kasus?", Jika pada soal diberikan 4 buah kondisi dan 4 buah aksi, tentu saja hal ini berlebihan.

Kondisi dinyatakan kontradiktif. Misalnya jika terdapat soal: “Apakah kondisi yang diberikan cukup untuk menjawab bahwa percabangan tersebut adalah percabangan tiga kasus? Bila pada soal diberikan informasi: Percabangan dalam kasus ini hanya ada melibatkan 3 kondisi saja yaitu membandingkan 3 harga toko buku. Tapi di baris berikutnya ada keterangan jika harga toko buku keempat adalah ...,. dan coba bandingkan.
Jelas ini kontradiktif karena sebelumnya di awal diberikan keterangan hanya ada 3 toko buku saja yang dibandingkan tetapi di kalimat berikutnya diberikan informasi jika harga toko buku keempat adalah

Kondisi dinyatakan cukup bila data yang diberikan sudah cukup untuk menjawab pertanyaan. Contohnya terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan tiga kasus?”, Pada soal diberikan 3 buah kondisi dan 3 aksi

Kondisi dinyatakan tidak cukup bila data yang diberikan tidak cukup untuk menjawab pertanyaan. Contoh terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan tiga kasus?”, Pada soal diberikan sebuah kondisi saja. Tentu data yang diberikan kurang atau pada soal tersebut tidak diberikan aksi jika kondisi tercapai. atau pada soal tidak diberi tahu bahasa pemrogramannya

Kondisi dinyatakan tidak ada konteks yang tepat bila tidak menggambarkan sama sekali percabangan

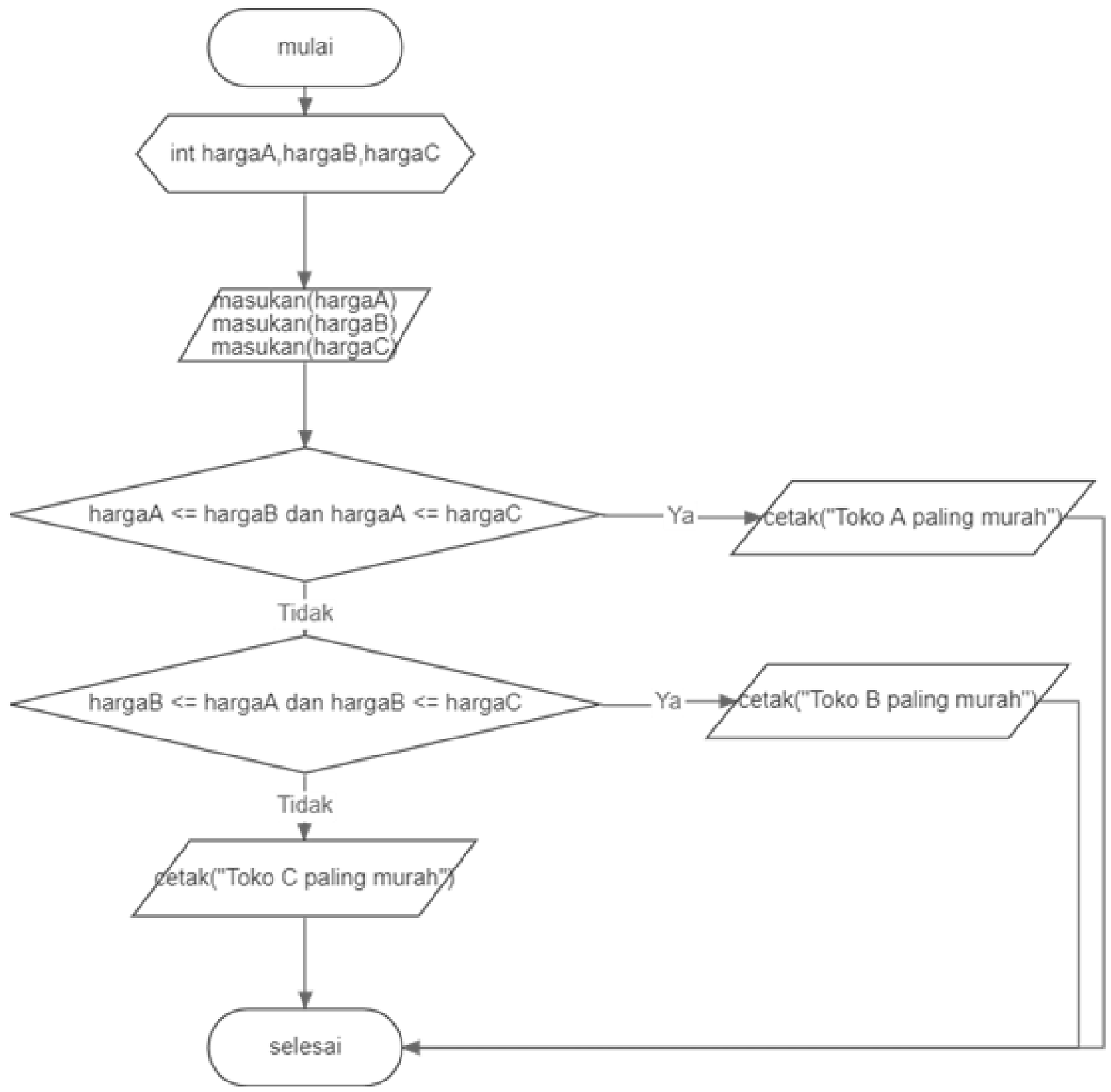
3. Bila diberi tahu bahwa hanya ada 2 harga saja yang dibandingkan dan menghasilkan output misalnya “Toko A paling murah” atau “Toko B paling murah” , apakah dapat menentukan bahwa percabangan yang digunakan adalah percabangan tunggal?
Berlebihan, karena terdapat dua kondisi pada kasus ini.

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:

```cpp
#include <iostream>
using namespace std;
int main(){
    int hargaA,hargaB,hargaC;
    cin>>hargaA;
    cin>>hargaB;
    cin>>hargaC;
    if(hargaA <= hargaB && hargaA <= hargaC){
        cout<<"Toko A paling murah";
    }else if(hargaB <= hargaA && hargaB <= hargaC){
        cout<<"Toko B paling murah";
    }else{
        cout<<"Toko C paling murah";
    }
  return 0;
}
```

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

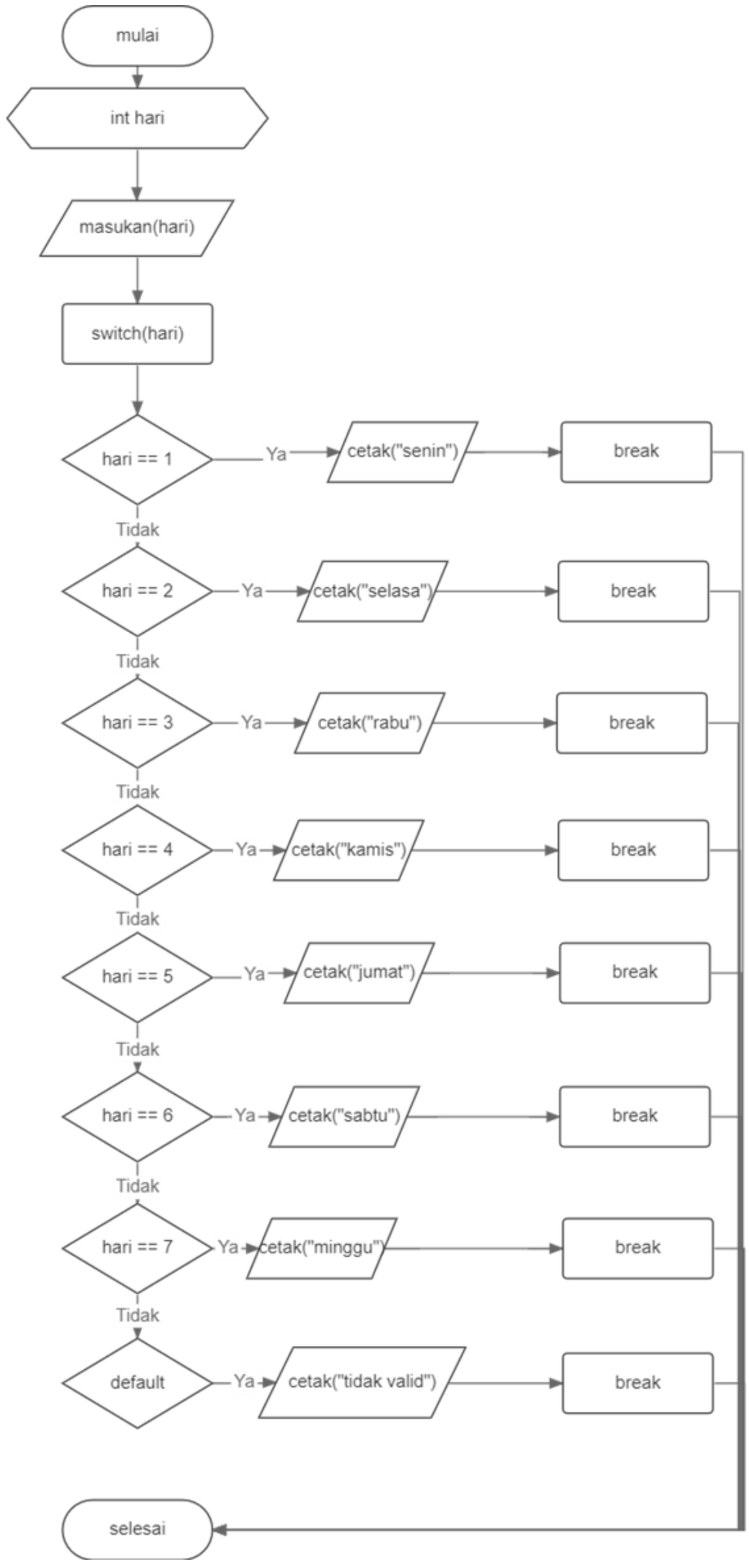
```
#include <iostream>
using namespace std;
int main() {

    int hargaA,hargaB,hargaC;
    cin>>hargaA;
    cin>>hargaB;
    cin>>hargaC;
    if(hargaA >= hargaB && hargaA >= hargaC){
        cout<<"Harga A paling mahal";
    }else if(hargaB >= hargaA && hargaB >= hargaC){
        cout<<"Harga B paling mahal";
    }else{
        cout<<"Harga C paling mahal";
    }
    return 0;
}
```

Dengan menggubah 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. Switch ini dapat menyederhanakan pembacaan if-else yang bertumpuk sehingga lebih mudah dibaca.



Gambar 9 Flowchart Switch untuk Menentukan Hari

```
switch(switch_expr)
{
   case (constant expr1) :<statement>;
      <statement>;
      break;
   case (constant expr2) :<statement>;
      <statement>;
       break;
   .....
   default                :<statement>;
      <statement>;
      break;
}
```

Gambar 10 Struktur Switch

```cpp
#include <iostream>
using namespace std;

int main() {
    int nomorHari;

    cout << "Masukkan nomor hari (1-7): ";
    cin >> nomorHari;

    switch (nomorHari) {
        case 1:
            cout << "Senin" ;
            break;
        case 2:
            cout << "Selasa" ;
            break;
        case 3:
            cout << "Rabu" ;
            break;
        case 4:
            cout << "Kamis" ;
            break;
        case 5:
            cout << "Jumat" ;
            break;
        case 6:
            cout << "Sabtu" ;
            break;
        case 7:
            cout << "Minggu" ;
            break;
        default:
            cout << "Nomor hari tidak valid. Masukkan nomor antara 1 hingga 7." ;
            break;
    }

    return 0;
}
```

Gambar 11 Code Switch untuk Menentukan Hari pada C++

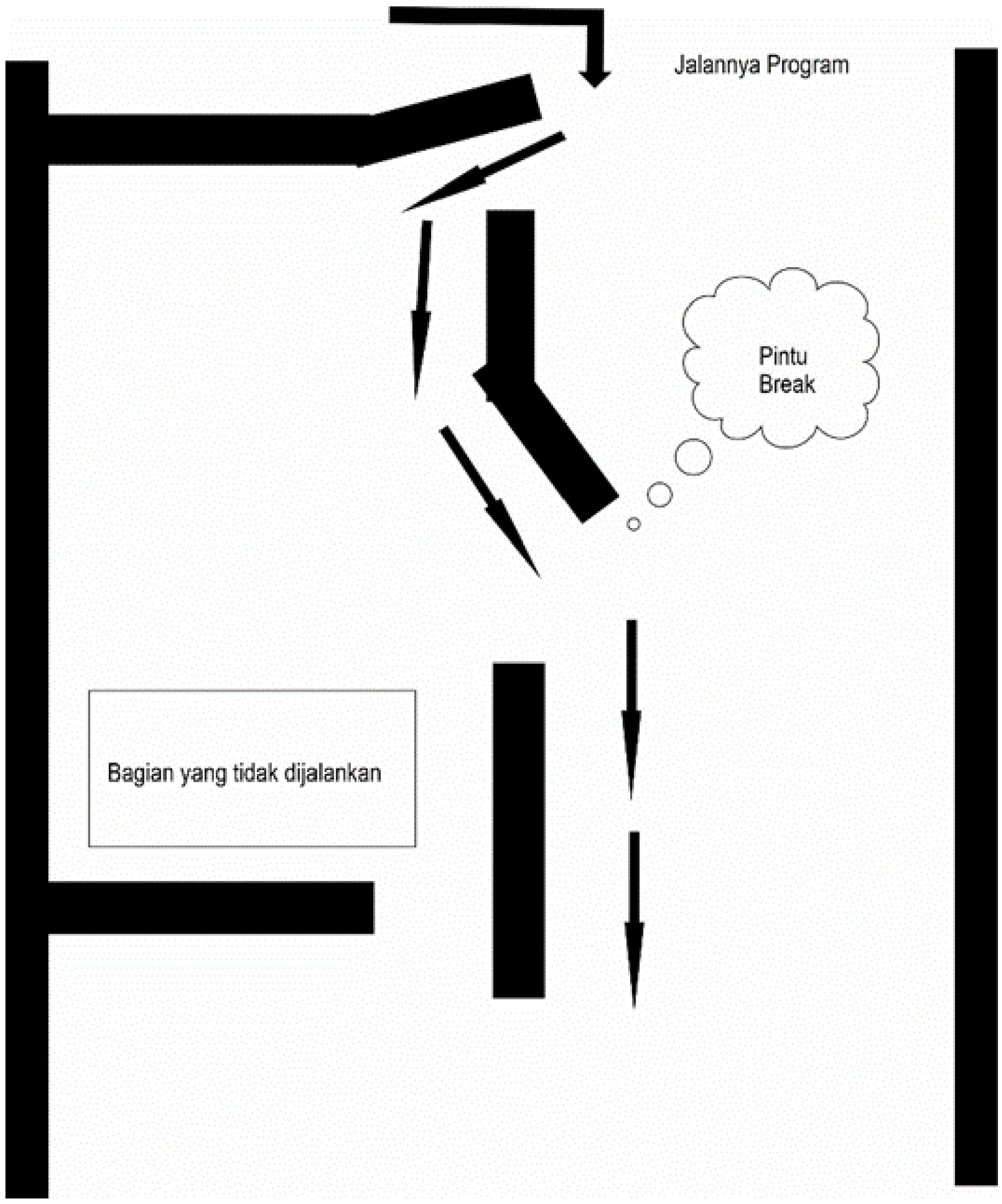
https://www.jdoodle.com/ia/SUa

Break

```cpp
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

Default

Kondisi Default pada switch itu sifatnya optional/tidak wajib. Jika tidak ada default pun tidak apa-apa

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++ dengan percabangan yang paling tepat dan meringkas sesuatu yang bertingkat-tingkat
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

Keterangan:
Kondisi dinyatakan berlebihan bila data yang diberikan berlebihan contohnya, Jika terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan switch?”, Jika pada soal tersebut ditambahkan informasi: “Jika percabangan yang akan digunakan ini nanti bukan hanya dapat memvalidasi angka saja akan tetapi bisa juga memvalidasi kata-kata. misalnya jika hari = “senin” maka mencetak “senin”. Jelas ini informasi yang berlebihan karena kalau tanpa informasi percabangan tersebut dapat memvalidasi kata-kata sudah dapat dipastikan bahwa percabangan yang cocok adalah switch

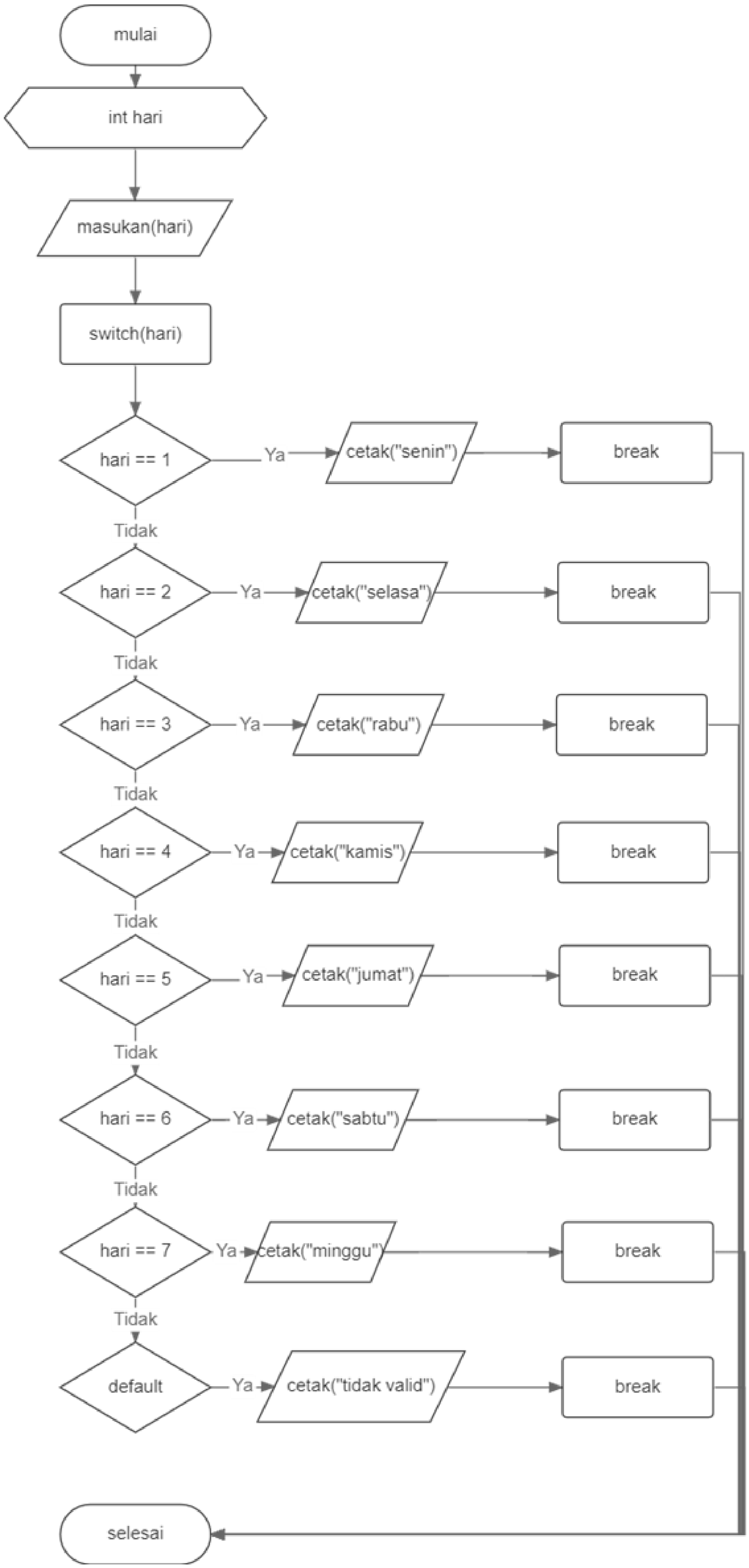
Kondisi dinyatakan kontradiktif. Misalnya jika terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan switch?”, Jika pada soal tersebut ditambahkan informasi: Percabangan ini hanya dapat memvalidasi angka dan karakter saja namun ada informasi tambahan “Jika percabangan yang akan digunakan ini nanti bukan hanya dapat memvalidasi angka saja akan tetapi bisa juga memvalidasi kata-kata. misalnya jika hari = “senin” maka mencetak “senin”. Jelas ini informasi yang kontradiktif karena sudah diberikan informasi bahwa percabangannya hanya dapat memvalidasi angka dan karakter saja.

Kondisi dinyatakan cukup bila data yang diberikan sudah cukup untuk menjawab pertanyaan. Contohnya terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan switch?”, Pada soal diberikan beberapa kondisi berupa membandingkan angka saja.

Kondisi dinyatakan tidak cukup bila data yang diberikan tidak cukup untuk menjawab pertanyaan. Contoh terdapat soal: “Apakah kondisi tersebut cukup untuk menentukan bahwa percabangannya adalah percabangan switch?”, Bila pada soal tidak diberikan kondisi yang dibandingkan atau kondisi yang dibandingkan terlalu sedikit misal hanya 1 kondisi saja atau pada soal tidak diberi tahu bahasa pemrogramannya

Kondisi dinyatakan tidak ada konteks yang tepat bila tidak menggambarkan sama sekali percabangan

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 ringkas penulisannya
2. Bagaimana flowchart diagramnya?


mulai
int hari
masukan(hari)
switch(hari)
hari == 1
Ya
cetak("senin")
break
Tidak
hari == 2
Ya
cetak("selasa")
break
Tidak
hari == 3
Ya
cetak("rabu")
break
Tidak
hari == 4
Ya
cetak("kamis")
break
Tidak
hari == 5
Ya
cetak("jumat")
break
Tidak
hari == 6
Ya
cetak("sabtu")
break
Tidak
hari == 7
Ya
cetak("minggu")
break
Tidak
default
Ya
cetak("tidak valid")
break
selesai

Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):
1. Bagaimana codenya dalam C++?
Jawab:

```cpp
#include <iostream>
using namespace std;

int main() {
    int nomorHari;

    cout << "Masukkan nomor hari (1-7): ";
    cin >> nomorHari;

    switch (nomorHari) {
        case 1:
            cout << "Senin" ;
            break;
        case 2:
            cout << "Selasa" ;
            break;
        case 3:
            cout << "Rabu" ;
            break;
        case 4:
            cout << "Kamis" ;
            break;
        case 5:
            cout << "Jumat" ;
            break;
        case 6:
            cout << "Sabtu" ;
            break;
        case 7:
            cout << "Minggu" ;
            break;
        default:
            cout << "Nomor hari tidak valid. Masukkan nomor antara 1 hingga 7." ;
            break;
    }

    return 0;
}
```

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

Munir, R., & Lidya, L. (2016). Algoritma Dan Pemrograman Dalam Bahasa Pascal, C, C++ Edisi Keenam. Bandung: Informatika.

Polya, G. (2014). How to Solve It: A New Aspect of Mathematical Method. New Jersey: Princeton University Press.

Sukamto, R. A. (2018). Logika Algoritma dan Pemograman Dasar. Bandung: Modula.