

Analisis Kasus - C++

Bahan Kuliah SD3105 Algoritma dan Pemrograman

Sevi Nurafni

Fakultas Sains dan Teknologi

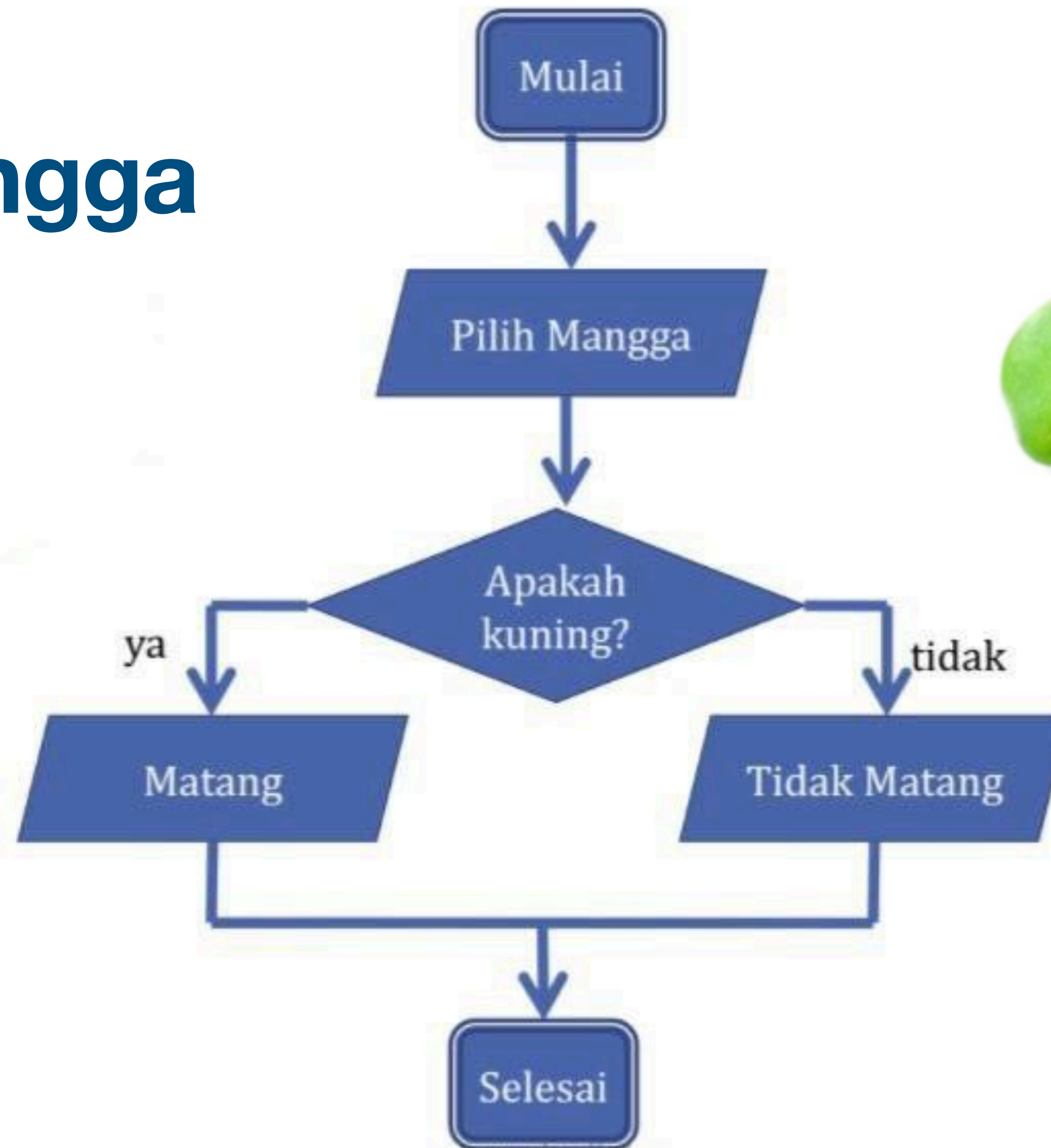
Universitas Koperasi Indonesia 2024

Contoh-1: Memilih Mangga

- Analisis kasus dapat digunakan dalam kehidupan sehari-hari,
Contoh: memilih mangga
- Mangga yang sudah matang dan siap dimakan adalah mangga yang berwarna kuning
- Jika tidak berwarna kuning maka tidak matang



Flowchart Memilih Mangga



Pseudocode - Memilih Mangga

PillihMangga

If (ApakahKuning? = true) **then**

Matang

Else {ApakahKuning? = false}

Tidak Matang

Analisis Kasus



- Memungkinkan kita membuat teks yang sama, namun menghasilkan eksekusi berbeda
- Sering disebut **percabangan / kondisional**
 - Dari satu langkah ada pilihan (bercabang) ke beberapa langkah
- Terdiri atas:
 - Kondisi: ekspresi yang menghasilkan true dan false
 - Aksi: statement yang dilaksanakan jika kondisi yang berpasangan dengan aksi dipenuhi

Analisis Kasus



- Analisis kasus harus memenuhi 2 kriteria:
 - **COMPLETE**: semua kasus terdefinisi secara lengkap
 - **DISJOINT**: tidak ada kasus yang tumpang tindih/overlapped
- Contoh: Diberikan sebuah bilangan bulat, misalnya A, nyatakan apakah bilangan tersebut adalah bilangan positif, negatif, atau nol
- Ada 3 kasus yang *complete* dan *disjoint*:
 - $A > 0$
 - $A < 0$
 - $A = 0$
 - Tidak ada kasus lain yang bisa didefinisikan dan ketiga kasus tersebut tidak tumpang tindih

Analisis Kasus: Satu kasus

```
if ( kondisi ) {  
    aksi;  
}
```

- Kondisi benar \rightarrow aksi dilakukan
- Kondisi tidak benar \rightarrow tidak terjadi apa-apa

Contoh: Satu kasus

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

int main () {
//KAMUS
    int i;

//ALGORITMA
    cin >> i;
    if (i >= 0) {
        cout << "positif" << endl;
    }
    return 0;
}
```


Analisis Kasus: Dua kasus



```
if ( kondisi ) {  
    aksi-1;  
}  
else { //not kondisi  
    aksi-2;  
}
```

- Perhatikan kondisi eksplisitnya!

Contoh: Dua kasus

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

int main () {
    //KAMUS
    int i;

    //ALGORITMA
    cin >> i;
    if (i >= 0) {
        cout << "positif" << endl;
    }
    else { //i < 0
        cout << "negatif" << endl;
    }
    return 0;
}
```

Analisis Kasus: Banyak kasus



```
if ( kondisi-2 ) {  
    aksi-1;  
}  
else if ( kondisi-2 ) {  
    aksi-2;  
}  
else {  
    ...  
}
```

Contoh: Dua kasus

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

int main () {
    //KAMUS
    int i;

    //ALGORITMA
    cin >> i;
    if (i > 0) {
        cout << "positif" << endl;
    }
    else if (i == 0) {
        cout << "nol" << endl;
    }
    else {
        cout << "negatif" << endl;
    }
    return 0;
}
```

Switch

```
Switch (var) {  
    case val-1: aksi-1; break;  
    case val-2: aksi-2; break;  
    ...  
    default: aksi-n; break;  
}
```

- **var** bertipe ordinal, **val-1**, **val-2**, dst adalah constant
- Break untuk keluar dari switch
- Default jika tidak ada yang cocok

Contoh: Switch

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

int main () {
    //KAMUS
    int i;

    //ALGORITMA
    cin >> i;
    switch (i) {
        case 1:
            cout << "Januari" << endl;
            break;
        case 2:
            cout << "Februari" << endl;
            break;
        default:
            cout << "Bulan lain" << endl;
            break;
    }
    return 0;
}
```


Analisis Kasus sebagai sarana validasi



```
if ( not kondisi-benar ) {  
    pesan-kesalahan;  
}  
else { //kondisi-benar  
    aksi;  
}
```

- Digunakan untuk memvalidasi masukan program sehingga masukan yang salah tidak diproses
- `aksi` hanya dilakukan jika `kondisi-benar` terpenuhi, jika tidak maka diberikan `pesan-kesalahan`.

Contoh: Analisis Kasus untuk validasi



- Buatlah program yang digunakan untuk menuliskan ke layar nama-nama bulan dari nomor bulan, yaitu 1 s.d. 12 yang dibaca dari keyboard.
- Program memvalidasi terlebih dahulu apakah masukan dari keyboard sudah benar atau belum masukan dari keyboard sudah benar atau belum, yaitu apakah nomor bulan sudah tepat 1 s.d. 12. Jika tidak tepat, maka dikeluarkan pesan kesalahan yaitu “Masukan nomor bulan tidak tepat”.

Contoh: Analisis Kasus untuk validasi

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

int main() {
    // KAMUS
    int nobulan;

    // ALGORITMA
    cin >> nobulan;
    if ((nobulan < 1) || (nobulan > 12)) {
        cout << "Masukan nomor bulan tidak tepat" << endl;
        // pesan kesalahan
    } else {
        switch (nobulan) {
            case 1:
                cout << "Januari" << endl;
                break;
            case 2:
                cout << "Februari" << endl;
                break;
            ... //lanjutkan sendiri
            default:
                cout << "Bulan tidak valid" << endl;
                break;
        }
    }

    return 0;
}
```

Soal-1

- Buatlah program yang menerima masukan 2 buah integer dan menghasilkan manakah dari kedua integer tersebut yang paling besar.
- Misal integernya adalah A dan B:
 - $A \geq B$ maka A
 - $A < B$ maka B
 - $A = B$ maka tidak ada yang lebih besar

Soal-2



- Buatlah program yang menerima 3 buah hambatan (R1, R2, R3) dan menghasilkan hambatan total (RT) jika dirangkai seri.

$$\mathbf{RT = R1 + R2 + R3}$$

- R1, R2, dan R3 tidak boleh bernilai negatif. Jika satu saja hambatan bernilai negatif, maka total hambatan tidak bisa dihitung dan tuliskan ke layar pesan kesalahan “Hambatan total tidak bisa dihitung”.

**SELAMAT
BELAJAR**