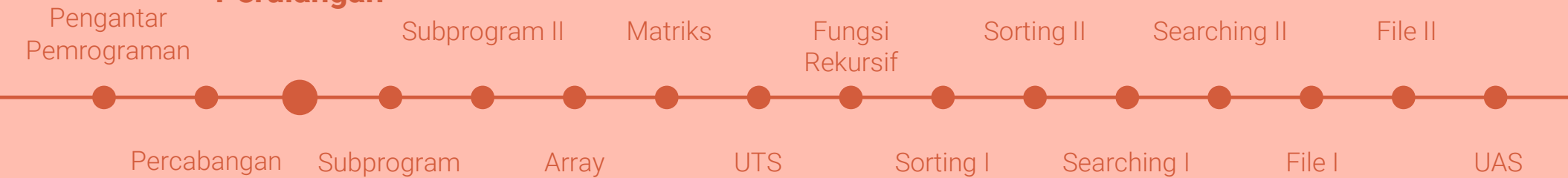


DASAR PEMROGRAMAN

Pertemuan II

Perulangan





Tujuan

- Mahasiswa memahami jenis-jenis perulangan dan penggunaannya serta memahami elemen-elemen dalam perulangan.
- Mahasiswa dapat menggunakan notasi perulangan yang sesuai dengan benar.
- Mahasiswa dapat memanfaatkan jenis-jenis perulangan dengan tepat dalam menyelesaikan persoalan sederhana yang diberikan.





Materi

STRUKTUR PERULANGAN

JENIS PERULANGAN

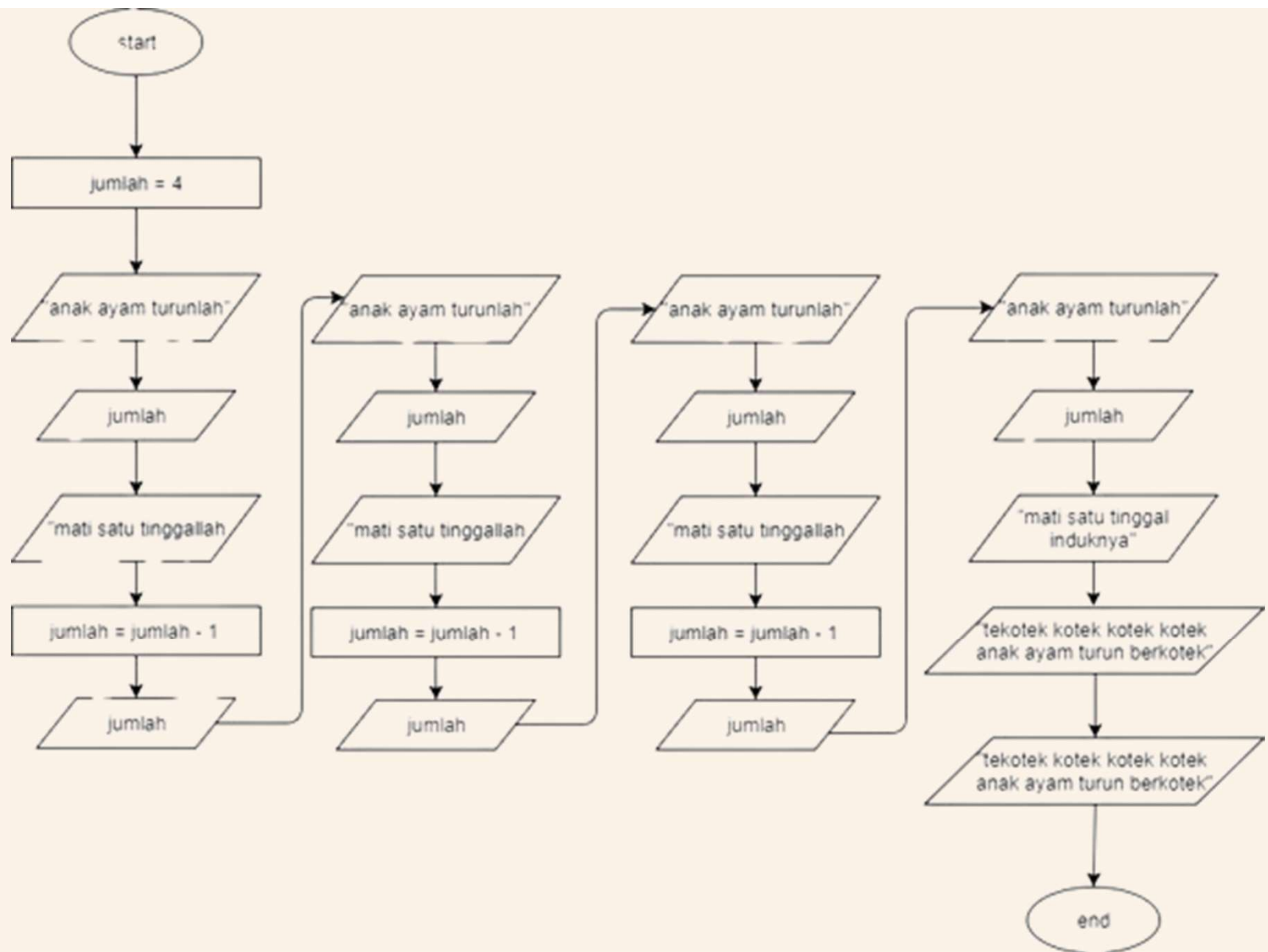
PERMASALAHAN UMUM

PERULANGAN

INGAT LAGU INI?



Bagaimana algoritma untuk
menampilkan lirik lagu tersebut?



Bagaimana jika ada

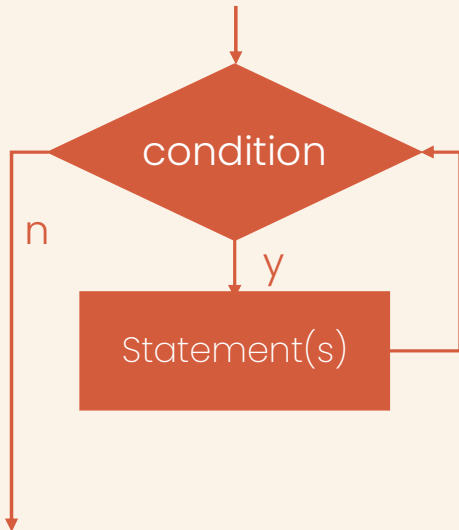
1.000.000 anak ayam?



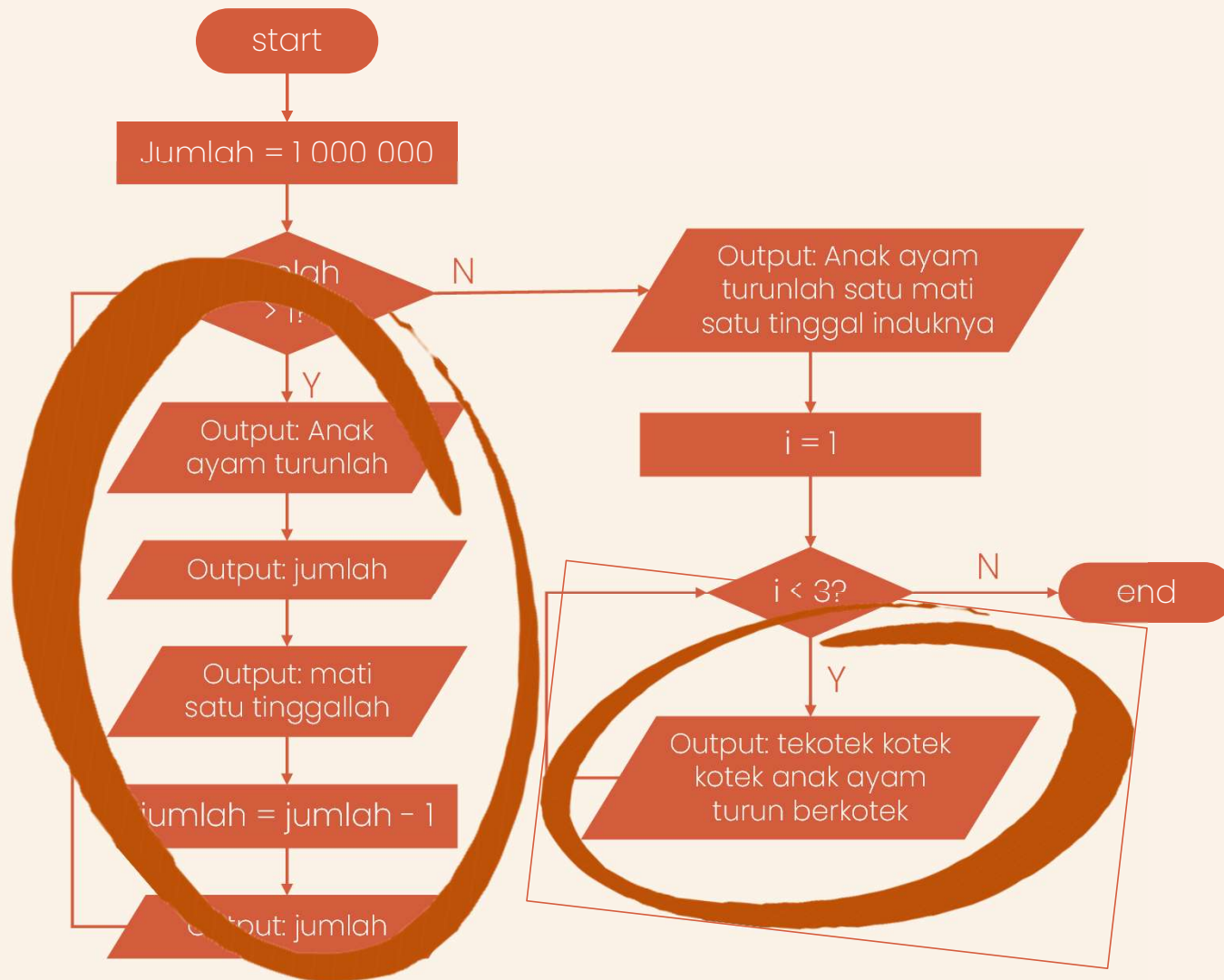
**Pengulangan menjalankan sebuah
atau serangkaian perintah secara
berulang-ulang.**

STRUKTUR PERULANGAN

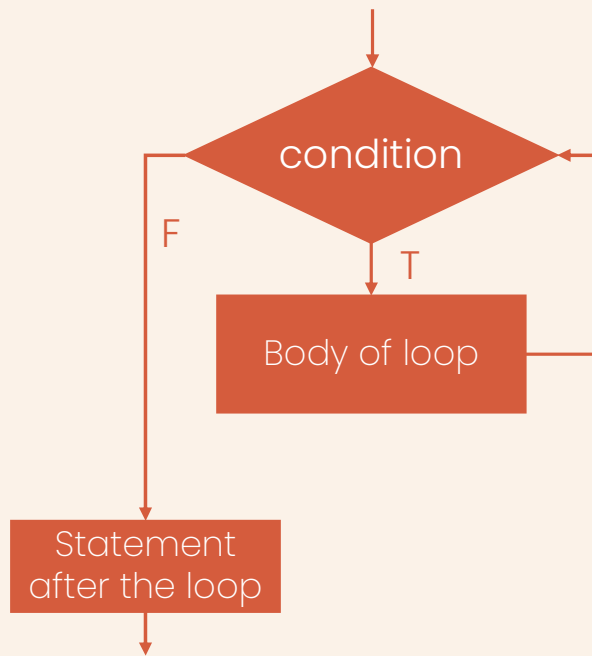
PENGULANGAN



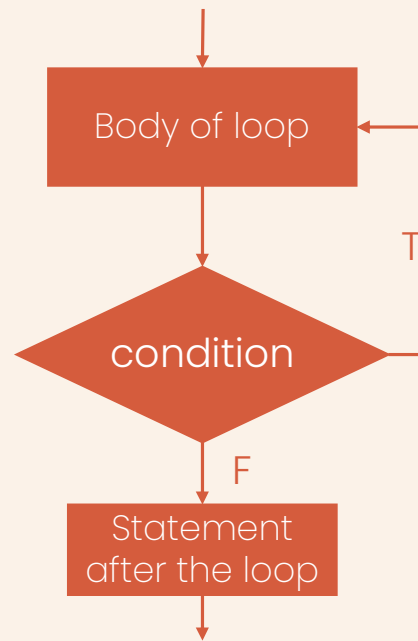
- Struktur algoritma dan pemrograman dimana serangkaian aksi dilakukan berulang untuk item yang berbeda hingga mencapai batas tertentu.
- Terdiri dari **kondisi/syarat** dan *body of loop*.



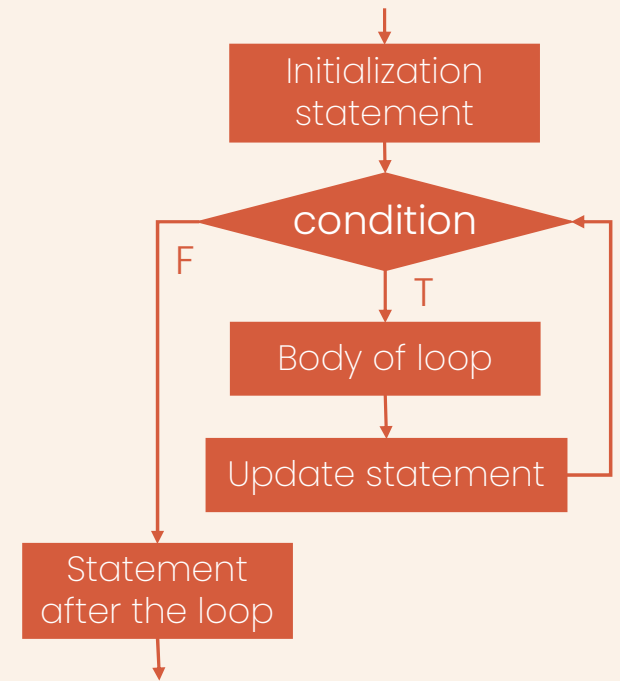
JENIS PERULANGAN



WHILE



DO-WHILE



FOR

```

#include <iostream>

using namespace std;

int main()
{
    int number, i = 1, factorial = 1;

    cout << "Enter a positive integer";
    cin >> number;

    while (i <= number)
    {
        factorial = factorial * i;
        i = i + 1;
    }

    cout << "Factorial of " << number
    << " = " << factorial;

    return 0;
}

```

WHILE

```

#include <iostream>

using namespace std;

int main()
{
    int number, i = 1, factorial = 1;

    cout << "Enter a positive integer";
    cin >> number;

    do
    {
        factorial = factorial * i;
        i = i + 1;
    }while (i <= number);

    cout << "Factorial of " << number
    << " = " << factorial;

    return 0;
}

```

DO-WHILE

```

#include <iostream>

using namespace std;

int main()
{
    int number, i, factorial = 1;

    cout << "Enter a positive integer";
    cin >> number;

    for (i = 1 ; i <= number ; i = i + 1)
    {
        factorial = factorial * i;
    }

    cout << "Factorial of " << number
    << " = " << factorial;

    return 0;
}

```

FOR

WHILE

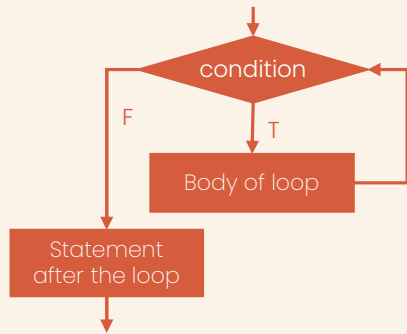
```
while (kondisi)
{
    // aksi yang diulang;
}
```

DO-WHILE

```
do
{
    // aksi yang diulang;
} while (kondisi);
```

FOR

```
for(inisialisasi;kondisi;update statement)
{
    // aksi yang diulang;
}
```



```
#include <iostream>

using namespace std;

int main()
{
    int number, i = 1, factorial = 1;

    cout << "Enter a positive integer";
    cin >> number;

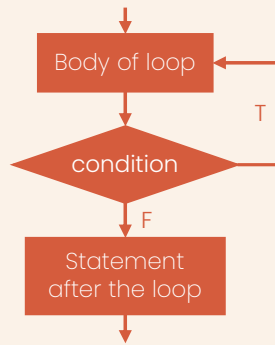
    while (i <= number)
    {
        factorial = factorial * i;
        i = i + 1;

        cout << "Factorial of " << number
        << " = " << factorial;

        return 0;
    }
}
```

WHILE

- 1 Pengulangan memeriksa kondisi pada syarat
- 2 Jika hasil pemeriksaan bernilai BENAR perintah pada body of loop dijalankan
- 3 Lakukan langkah **1** **2** , hingga hasilnya bernilai SALAH.
- 4 Pengulangan berakhir saat hasil pemeriksaan bernilai SALAH



DO-WHILE

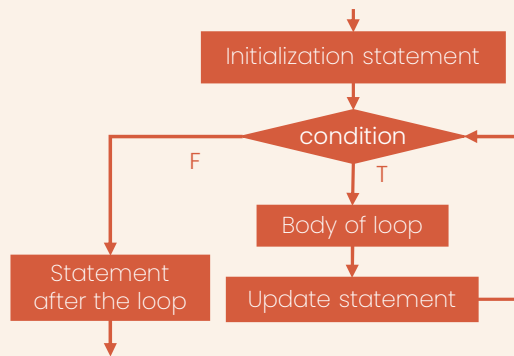
- 1 Perintah pada body of loop dijalankan setidaknya satu kali.
- 2 Kondisi pada syarat diperiksa.
- 3 Jika hasil pemeriksaan bernilai BENAR, lakukan langkah 1 2, hingga langkah 2 hasilnya SALAH.
- 4 Pengulangan berakhir saat hasil pemeriksaan bernilai SALAH

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

int main()
{
    int number, i = 1, factorial = 1;
    cout << "Enter a positive integer";
    cin >> number;

    do
    {
        1 factorial = factorial * i;
        i = i + 1;
    }while (i <= number); 2
    4 cout << "Factorial of " << number
    << " = " << factorial; 3

    return 0;
}
```



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

int main()
{
    int number, i, factorial = 1;

    cout << "Enter a positive integer";
    cin >> number;

    for (i = 1 ; i <= number ; i = i + 1)
    {
        factorial = factorial * i;
    }

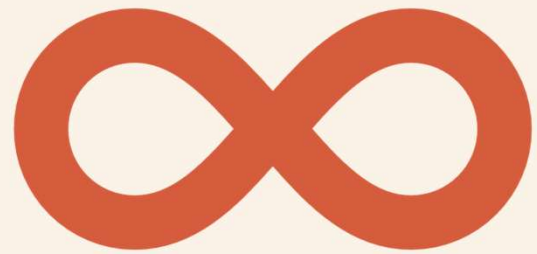
    cout << "Factorial of " << number
    << " = " << factorial;

    return 0;
}
```

FOR

- 1 Initialization statemen dijalankan 1 kali saja di awal.
- 2 Kondisi pada syarat diperiksa.
- 3 Jika hasil pemeriksaan bernilai BENAR perintah pada body of loop dijalankan.
- 4 Update statement dijalankan.
- 5 Lakukan langkah 2 3 4 , hingga hasilnya bernilai SALAH.
- 6 Pengulangan berakhir saat hasil pemeriksaan bernilai SALAH

PERMASALAHAN UMUM



The infinite loop
(Looping Forever)



MENGHENTIKAN PENGULANGAN

- Pastikan variable yang berubah di dalam body of loop mencapai nilai tertentu yang tidak memenuhi syarat pada conditional statement.
- Ada `break` statement pada kondisi tertentu.
- Ada suatu cara dimana pengguna bisa menginputkan sesuatu yang nilainya tidak memenuhi syarat pada conditional statement.

LATIHAN SOAL

LATIHAN 1

Buatlah program yang menerima masukan 10 buah bilangan integer (dari keyboard) dan menuliskan ke layar jumlah total ke-10 integer tersebut.

Contoh:

Input:

2
1
0
-9
7
13
2
2
1
-1

Output: 18

LATIHAN 2

Buatlah program yang membaca sejumlah bilangan integer dari keyboard sampai pengguna memasukkan angka -999 (angka -999 tidak termasuk bilangan yang diolah).

Tuliskan berapa banyak bilangan yang dimasukkan, nilai total, dan rata-rata semua bilangan

Jika dari masukan pertama sudah menuliskan -999, maka tuliskan pesan "Tidak ada data yang diolah"