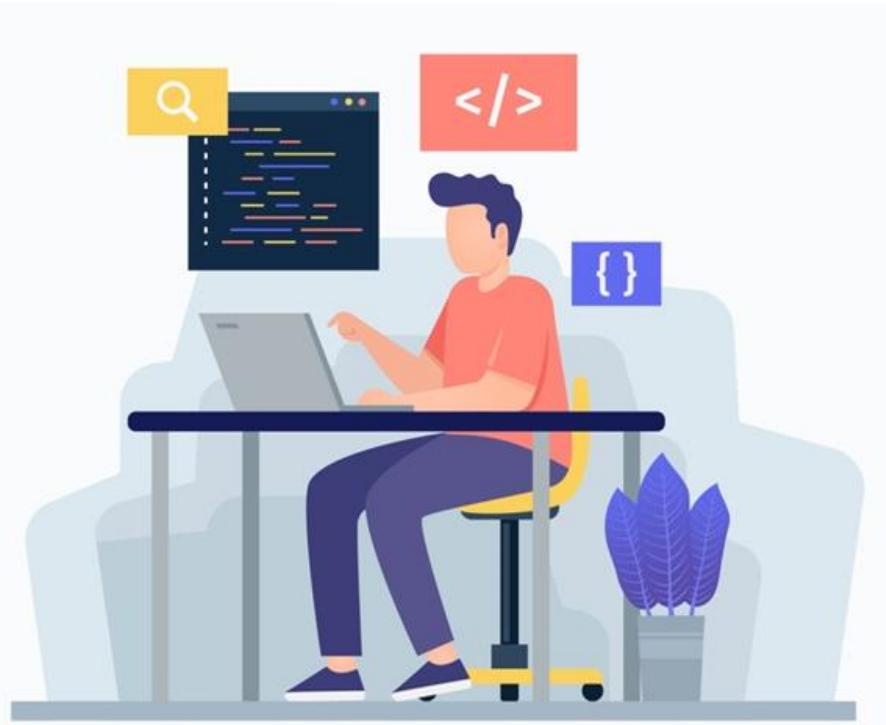




**PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

Mata Kuliah
Dasar Pemrograman



Perulangan (Looping)

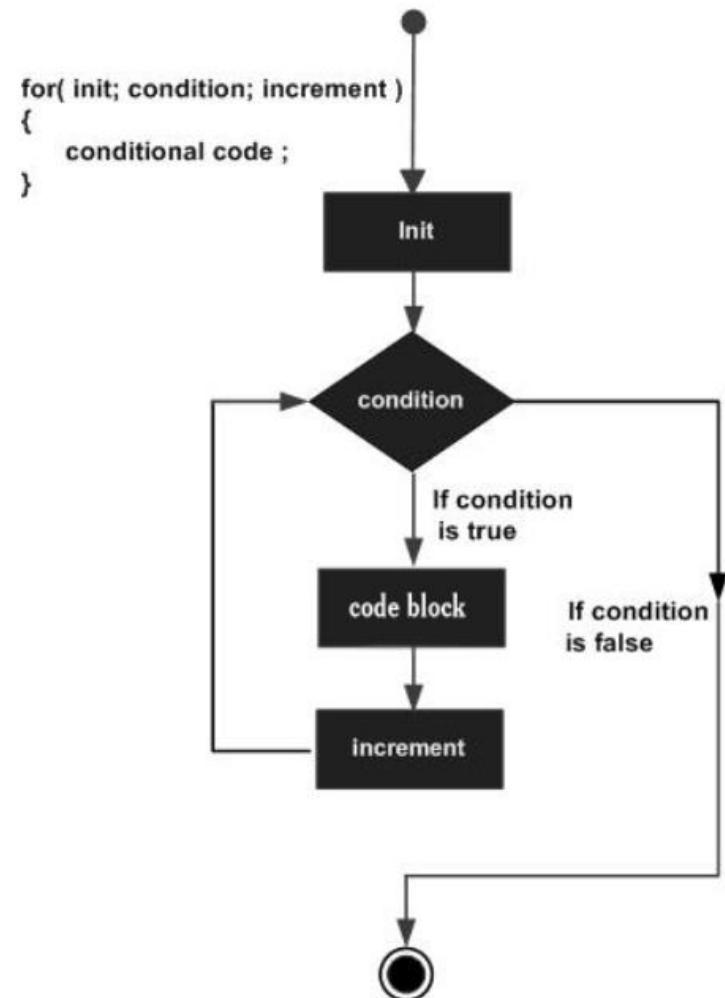
TIM DASAR PEMROGRAMAN
TEKNIK INFORMATIKA S1
UNIVERSITAS DIAN NUSWANTORO

Capaian Pembelajaran

- Setelah mengikuti kuliah ini mahasiswa dapat menjelaskan dan mempraktekkan jenis pengulangan berdasarkan jumlah pengulangan dan berdasarkan kondisi kondisi, dua aksi, dan pencacah dalam pemrograman
- Setelah mengikuti kuliah ini mahasiswa dapat menjelaskan dan mempraktekkan jenis pengulangan bersarang

Pengulangan C++ (*looping in C++*)

- Bisa dengan blok kode for
- Bisa dengan blok kode while
- Bisa dengan blok kode do.. While

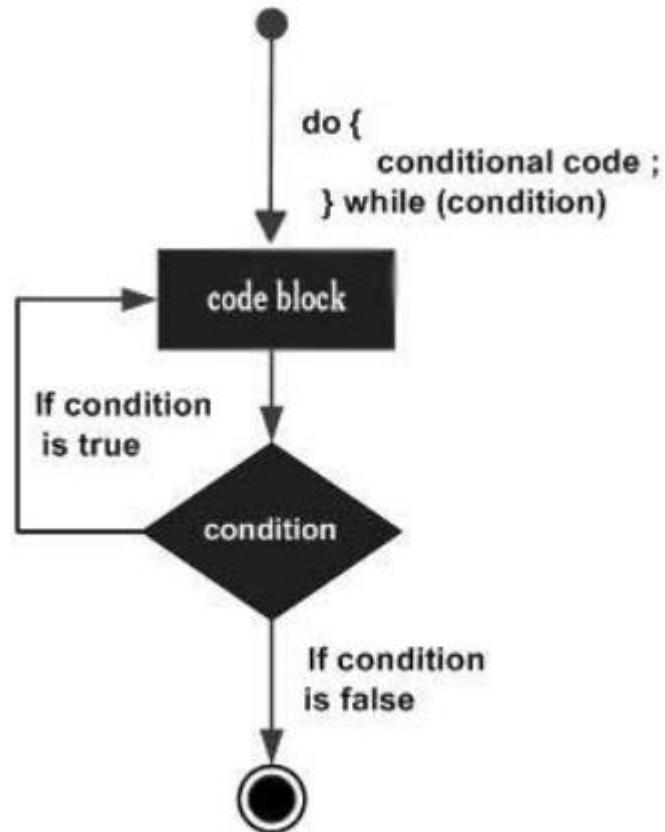
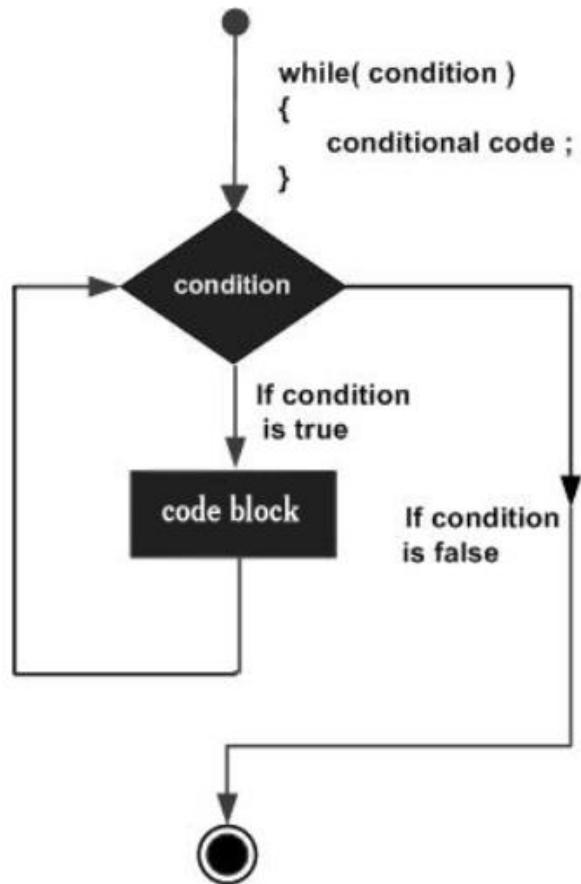


Sintak:

```
for ( <init>; <kondisi>; <increment> ) {  
    aksi;  
}
```

*Init merupakan variabel dengan suatu inisialisasi nilai.
Kondisi kondisi variabel tersebut sebagai syarat pengulangan
Increment/decrement memungkinkan pengulangan akan naik atau turun atau dengan step tertentu*

While vs Do .. While

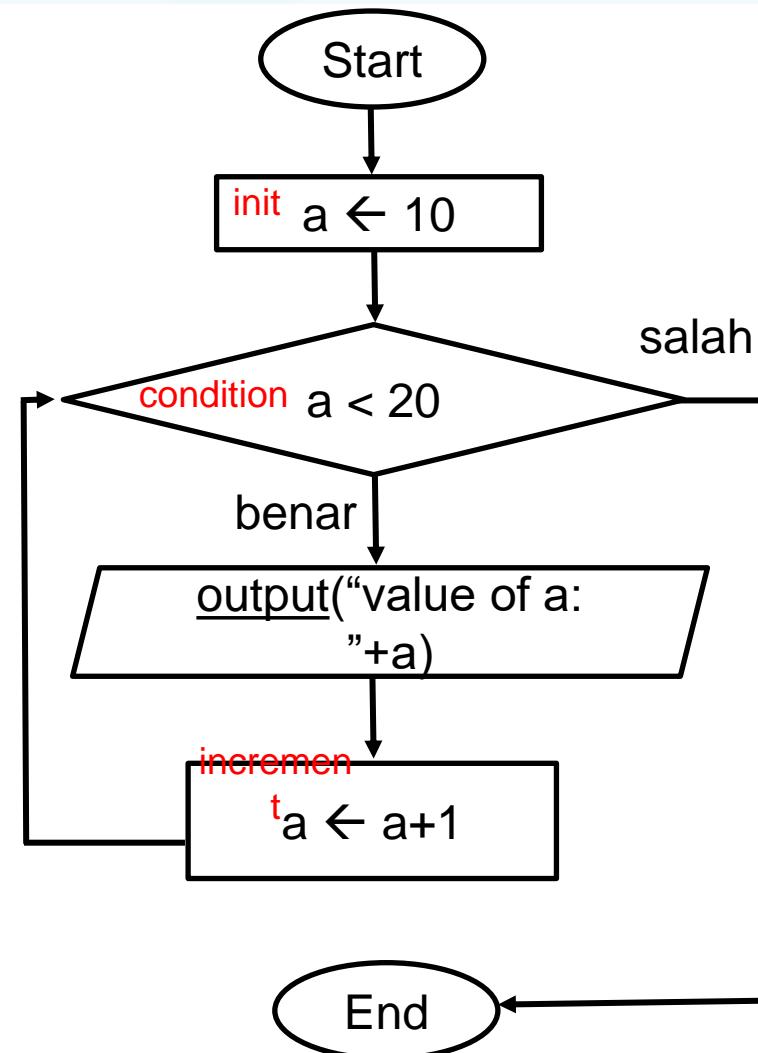


Contoh Kode For

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

int main () {
    // for loop execution
    for( int a = 10; a < 20; a = a + 1 ) {
        cout << "value of a: " << a << endl;
    }

    return 0;
}
```



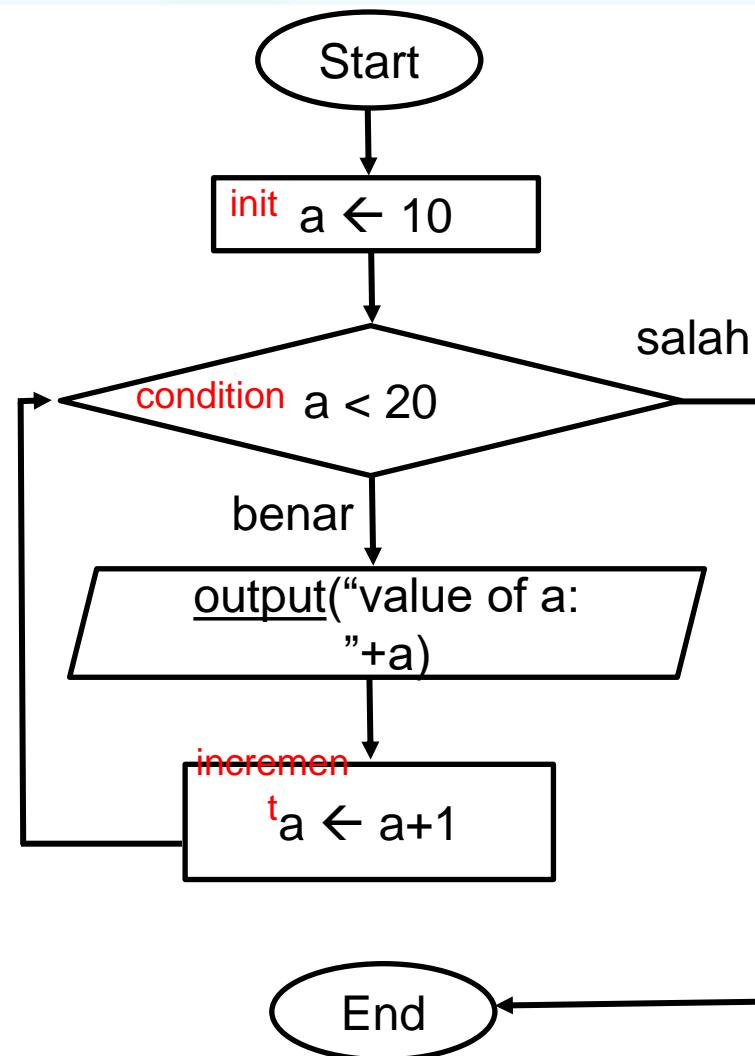
Contoh Kode While

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

int main () {
    // Local variable declaration:
    int a = 10;

    // while loop execution
    while( a < 20 ) {
        cout << "value of a: " << a << endl;
        a++;
    }

    return 0;
}
```



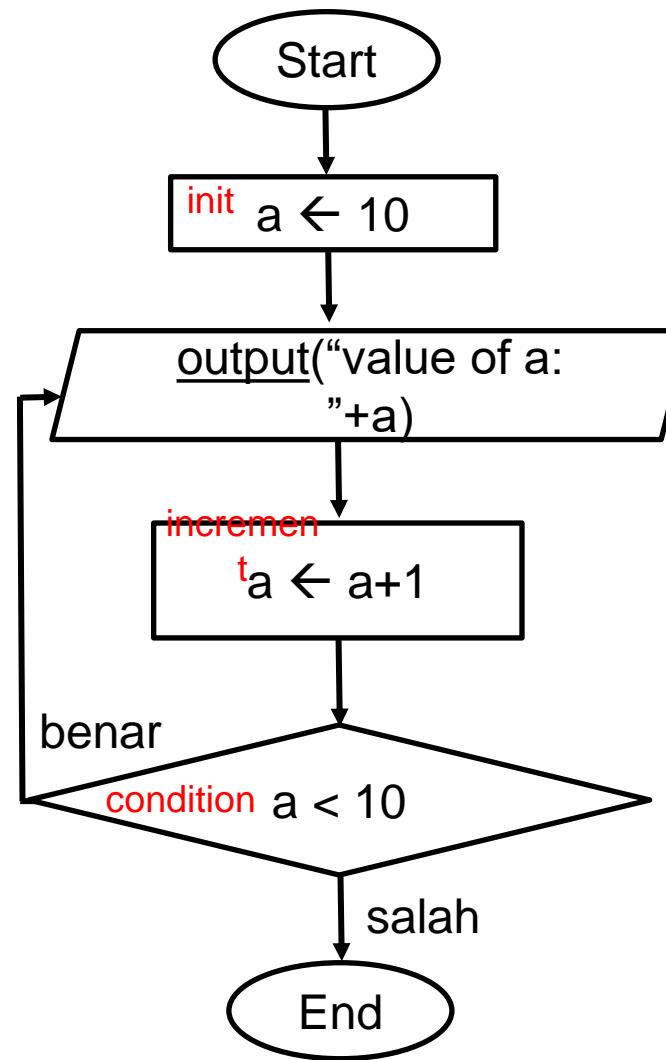
Contoh Kode Do.. While

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

int main () {
    // Local variable declaration:
    int a = 10;

    // do loop execution
    do {
        cout << "value of a: " << a << endl;
        a = a + 1;
    } while( a < 20 );

    return 0;
}
```



Beberapa percobaan!

- Berikut adalah cara untuk membuat pengulangan yang tidak pernah berhenti:

```
while(true)
{
    //aksi
}

do
{
    //aksi
} while(true);
```

```
for( ; ; )
{
    //aksi
}
```

Notasi pengulangan

- Jenis notasi pengulangan:
 1. Berdasarkan jumlah pengulangan
 2. Berdasarkan kondisi berhenti
 3. Berdasarkan kondisi pengulangan
 4. Berdasarkan dua aksi
 5. Berdasarkan pencacah

Program CetakMahasiswa

Program CetakMahasiswa

{Program untuk mencetak mahasiswa berkali-kali dengan batasan n input oleh user}

KAMUS

n : integer

ALGORITMA

input(n)
repeat n times
 output("mahasiswa")

Program CetakMahasiswa C++

```
main.cpp x
1  /*
2  JUDUL: Program CetakMahasiswa
3  {Program untuk mencetak mahasiswa berkali-kali dengan batasan n input oleh user}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int n;
13     //Algoritma
14     cin >> n; // pastikan bahwa n lebih dari 0
15     for(;n>0;n--)
16     {
17         cout << "mahasiswa" << endl;
18     }
19     return 0;
20 }
```

Contoh Persoalan Komputasional

- Tulis semua bilangan positif lebih dari 0 dari 1, 2, 3, ..., sampai N. Perhatikan bahwa N merupakan bilangan bulat positif yang di inputkan oleh user.

Program CetakBilanganPositif

{Tulis semua bilangan positif lebih dari 0 dari 1, 2, 3, ..., sampai N. Perhatikan bahwa N merupakan bilangan bulat positif yang di inputkan oleh user.}

KAMUS

n , i : integer {i merupakan bilangan yang akan di outputkan nantinya didalam pengulangan}

ALGORITMA

```
i ← 1  
input(n)  
repeat n times  
    output(i)  
    i ← i+1
```

Jawaban di kode bahasa C++

```
*main.cpp x
1  /*
2  JUDUL: CetakBilanganPositif
3  {Tulis semua bilangan positif lebih dari 0 dari 1, 2, 3, ..., sampai N.
4  Perhatikan bahwa N merupakan bilangan bulat positif yang di inputkan oleh user.}
5  */
6  #include <iostream>
7
8  using namespace std;
9
10 int main()
11 {
12     //Kamus
13     int n,i;
14
15     //Algoritma
16     i = 1;
17     cin >> n; // pastikan bahwa n lebih dari 0
18     for(;n>0;n--)
19     {
20         cout << i << endl;
21         i=i+1;
22     }
23     return 0;
24 }
```

Jawaban di kode bahasa C++ versi singkat

```
main.cpp x
1  /*
2  JUDUL: CetakBilanganPositif
3  {Tulis semua bilangan positif lebih dari 0 dari 1, 2, 3, ..., sampai N.
4  Perhatikan bahwa N merupakan bilangan bulat positif yang di inputkan oleh user.}
5  */
6 #include <iostream>
7
8 using namespace std;
9
10 int main()
11 {
12     //Kamus
13     int n;
14
15     //Algoritma
16     cin >> n; // pastikan bahwa n lebih dari 0
17     for(int i=1;n>0;n--,i++)
18     {
19         cout << i << endl;
20     }
21     return 0;
22 }
```

Program ChatbotSederhana

Program ChatbotSederhana

{program chatbot sederhana, tanya “apa mau belok kiri? (y/t)” dengan diberikan variabel belokkiri yang merupakan input user yang bertipe karakter}

KAMUS

belokkiri : char

ALGORITMA

```
repeat
    output("apa mau belok kiri? (y/t)")
    input(belokkiri)
until belokkiri = 'y'
output("finish")
```

Program ChatbotSederhana di C++

```
main.cpp x
1  /*
2  JUDUL: Program ChatbotSederhana
3  {program chatbot sederhana, tanya "apa mau belok kiri? (y/t)"
4  dengan diberikan variabel belokkiri yang merupakan input user yang bertipe karakter
5  */
6  #include <iostream>
7
8  using namespace std;
9
10 int main()
11 {
12     //Kamus
13     char belokkiri;
14
15     //Algoritma
16     do
17     {
18         cout << "apa mau belok kiri? (y/t)";
19         cin >> belokkiri;
20     }while(belokkiri != 'y'); //jika repeat ... until syaratnya harus di negasikan
21     cout <<"finish";
22
23     return 0;
24 }
```

Contoh Persoalan Komputasional yang lain

- Tulis semua bilangan positif lebih dari 0 dari 1, 2, 3, ..., sampai N. Perhatikan bahwa N merupakan bilangan bulat positif yang di inputkan oleh user.

KAMUS

n , i : integer {i merupakan bilangan yang akan di outputkan nantinya didalam pengulangan}

ALGORITMA

```
input(n)  
i ← 1  
repeat  
    output(i)  
    i ← i+1  
until i = n {perhatikan i = n ini juga boleh diganti i >= n}
```

Jawaban

```
main.cpp x
1  /*
2  JUDUL: Program CetakBilanganPositifVer2
3  {Tulis semua bilangan positif lebih dari 0 dari 1, 2, 3, ..., sampai N.
4  Perhatikan bahwa N merupakan bilangan bulat positif yang di inputkan oleh user.}
5  */
6  #include <iostream>
7
8  using namespace std;
9
10 int main()
11 {
12     //Kamus
13     int n,i;
14
15     //Algoritma
16     i = 1;
17     cin >> n; // pastikan bahwa n lebih dari 0
18     do
19     {
20         cout << i << endl;
21         i=i+1;
22     }while (i!=n); // jika n ikut serta ganti while (i<=n);
23     return 0;
24 }
```

Notasi pengulangan

- Jenis notasi pengulangan:
 1. Berdasarkan jumlah pengulangan
 2. Berdasarkan kondisi berhenti
 3. **Berdasarkan kondisi pengulangan**
 4. **Berdasarkan dua aksi**
 5. **Berdasarkan pencacah**

- Bagaimana cara mengkoding output di bawah ini? Sebaiknya menggunakan looping yang mana?

```
Selamat datang di chatbot SimSimi
...
Apakah mau tanya lagi (y/t)? y

Selamat datang di chatbot SimSimi
...
Apakah mau tanya lagi (y/t)? y

Selamat datang di chatbot SimSimi
...
Apakah mau tanya lagi (y/t)? t

Terima kasih sudah bertanya di chatbot SimSimi
```

Program ChatbotSederhana#2 C++

```
main.cpp x
1  /*
2  JUDUL: Program ChatbotSede
3  {program chatbot sederhana, tanya "apa mau belok kiri? (y/t)" dengan diberikan variabel belokkiri
4  yang merupakan input user yang bertipe karakter}
5  */
6  #include <iostream>
7
8  using namespace std;
9
10 int main()
11 //kamus
12 char tanyaLagi;
13
14 //algoritma
15 do{
16     cout << "Selamat datang di chatbot SimSimi" << endl;
17     cout << "..." << endl;
18     cout << "Apakah mau tanya lagi (y/t)? ";
19     cin >> tanyaLagi;
20     cout << endl;
21 }while(tanyaLagi=='y');
22 cout << "Terima kasih sudah bertanya di chatbot SimSimi" << endl;
23
24 return 0;
25 }
```

Program HalloNTimes

Program HalloNTimes

{program mencetak kata “Hallo” sebanyak n kali}

KAMUS

n,i : int

ALGORITMA

```
input(n)
i ← 0
iterate
    i ← i+1
    output("Hallo")
    stop i ≥ n:
        print("stop")
        break
```

Program HalloNTimes C++

```
main.cpp x
1  /*
2  JUDUL: HalloNTimes
3  {program mencetak kata "Hallo" sebanyak n kali}
4  */
5 #include <iostream>
6
7 using namespace std;
8
9 int main()
10{
11    //Kamus
12    int n,i;
13    //Algoritma
14    cin >> n;
15    i = 0;
16    while(true) //bisa di ganti for(;;) atau while(1)
17    {
18        i += 1; // atau n = n+1
19        cout << "hallo" << endl;
20        if(i>=n)
21        {
22            cout <<"stop";
23            break;
24        }
25    }
26    return 0;
27}
```

Program CetakMahasiswa

Program CetakMahasiswa

{Program untuk mencetak mahasiswa berkali-kali dengan batasan n input oleh user}

KAMUS

n, i : integer

ALGORITMA

input(n)

i traversal [1 ... n]

output("mahasiswa")

output("finish")

Program CetakMahasiswa

```
main.cpp x
1  /*
2  JUDUL: CetakMahasiswa
3  {Program untuk mencetak mahasiswa berkali-kali dengan batasan n input oleh user}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int n, i;
13     //Algoritma
14     cin >> n;
15     for(i=1; i<=n; i++)
16     {
17         cout << "mahasiswa" << endl;
18     }
19     cout << "finish" << endl;
20
21     return 0;
22 }
```

Contoh Persoalan Komputasional

Program CetakBilanganPositifRange

{Tulis semua bilangan positif lebih dari 0 mulai dari x sampai y. Perhatikan bahwa x dan y merupakan bilangan bulat positif yang diinputkan oleh user.}

KAMUS

x, y , i : integer {i merupakan bilangan yang akan di outputkan nantinya didalam pengulangan}

ALGORITMA

```
input(x,y)
i traversal [x ... y]
    output(i)
output("finish")
```

Contoh Persoalan Komputasional

```
main.cpp x
1  /*
2  JUDUL: CetakBilanganPositifRange
3  {Tulis semua bilangan positif lebih dari 0 mulai dari x sampai y.
4  Perhatikan bahwa x dan y merupakan bilangan bulat positif yang di inputkan oleh user.}
5  */
6  #include <iostream>
7
8  using namespace std;
9
10 int main()
11 {
12     //Kamus
13     int x,y,i;
14     //Algoritma
15     cin >> x >>y;
16     for(i=x;i<=y;i++)
17     {
18         cout << "mahasiswa" <<endl;
19     }
20     cout << "finish" <<endl;
21
22     return 0;
23 }
```

Contoh Persoalan Komputasional

Program CetakBilanganGenapPositifRange

{Tulis semua bilangan genap positif lebih dari 0 mulai dari 0 sampai n. Perhatikan bahwa n merupakan bilangan bulat positif yang diinputkan oleh user.}

KAMUS

n, i : integer {i merupakan bilangan yang akan di outputkan nantinya didalam pengulangan}

ALGORITMA

```
input(n)
i traversal [1 ... n]
    i ← i+1           {variabel i akan digunakan sebagai jeda}
    output(i)
    output("finish")
```

Jawaban

```
main.cpp x
1  /*
2  JUDUL: CetakBilanganGenapPositifRange
3  {Tulis semua bilangan genap positif lebih dari 0 mulai dari 0 sampai n.
4  Perhatikan bahwa n merupakan bilangan bulat positif yang di inputkan oleh user.}
5
6  */
7  #include <iostream>
8
9  using namespace std;
10
11 int main()
12 {
13     //Kamus
14     int n,i;
15     //Algoritma
16     cin >> n;
17     for(i=1;i<=n;i++)
18     {
19         i = i+1; //atau i++;
20         cout << i << endl;
21     }
22     cout << "finish" << endl;
23
24     return 0;
25 }
```

Alternatif Jawaban lain

```
main.cpp x
1  /*
2  JUDUL: CetakBilanganGenapPositifRange
3  {Tulis semua bilangan genap positif lebih dari 0 mulai dari 0 sampai n.
4  Perhatikan bahwa n merupakan bilangan bulat positif yang di inputkan oleh user.}
5
6  */
7  #include <iostream>
8
9  using namespace std;
10
11 int main()
12 {
13     //Kamus
14     int n,i;
15     //Algoritma
16     cin >> n;
17     for(i=0;i<=n;i+=2)
18     {
19         if(i == 0)
20             continue;
21         cout << i << endl;
22     }
23     cout << "finish" << endl;
24
25     return 0;
26 }
```

Alternatif Jawaban lain

```
main.cpp x
1  /*
2  JUDUL: CetakBilanganGenapPositifRange
3  {Tulis semua bilangan genap positif lebih dari 0 mulai dari 0 sampai n.
4  Perhatikan bahwa n merupakan bilangan bulat positif yang di inputkan oleh user.}
5
6  */
7  #include <iostream>
8
9  using namespace std;
10
11 int main()
12 {
13     //Kamus
14     int n,i;
15     //Algoritma
16     cin >> n;
17     for(i=1;i<=n;i++)
18     {
19         if(i%2==0)
20             cout << i << endl;
21     }
22     cout << "finish" << endl;
23
24     return 0;
25 }
```

Break vs Continue

- Perintah break digunakan didalam pengulangan untuk segera mengehentikan pengulangan.
- Aksi setelah perintah break tidak akan di eksekusi dan pengulangan berhenti langsung

Perintah continu digunakan untuk meneruskan pengulangan artinya, ketika continu ini di panggil didalam pengulangan maka aksi setelahnya tidak akan di eksekusi di iterasi sebelumnya. Istilahnya skip. Dalam hal ini pengulangan terus berjalan

Algoritma Penjumlahan

- Penjumlahan yang berkaitan dengan deret angka.
- Biasanya deret angka tersebut kita implementasikan di List/Array. Tapi mari kita coba dengan sederet angka yang di masukkan user.
- Contoh:
 - $\sum_{i=0}^n x_i = x_0 + x_1 + x_2 + \dots + x_n$
 - $\sum_{i=0}^n 2 + x_i = (2 + x_0) + (2 + x_1) + (2 + x_2) + \dots + (2 + x_n)$

Contoh input: 3 (ket: banyaknya input user)

1 (ket: inputan ke 1)

1 (ket: inputan ke 2)

1 (ket: inputan ke 3)

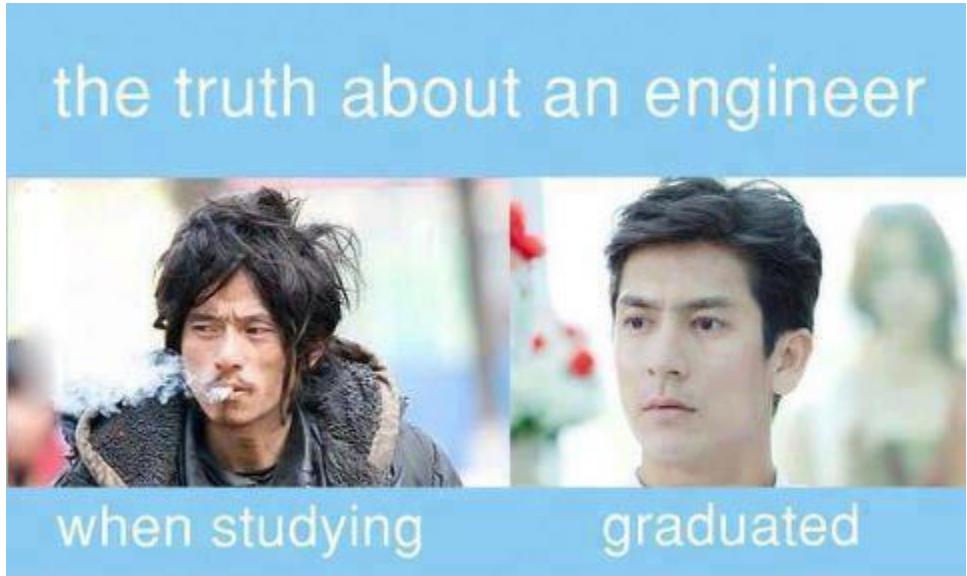
Contoh output: 3 (hasil penjumlahan ketiga inputan)

Jawaban

```
main.cpp x
1  /*
2  JUDUL: CetakBilanganGenapPositifRange
3  {Tulis semua bilangan genap positif lebih dari 0 mulai dari 0 sampai n.
4  Perhatikan bahwa n merupakan bilangan bulat positif yang di inputkan oleh user.}
5
6  */
7  #include <iostream>
8
9  using namespace std;
10
11 int main()
12 {
13     //Kamus
14     int n, i, a, sum;
15     //Algoritma
16     cin >> n;
17     sum = 0;
18     for(i=0; i<n; i++)
19     {
20         cin >> a;
21         sum = sum+a;
22     }
23     cout << sum << endl;
24
25     return 0;
26 }
```

Cara cepat jadi programmer?

- BERLATIH BERLATIH BERLATIH !
 - RESEP: NGODING 3X SEHARI SETELAH MAKAN



Referensi

Utama:

1. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional

Pendukung:

1. Introduction to Computer Science and Programming in Python, MIT
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016>
2. Introduction to Computer Science and Programming, MIT
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm>



TERIMA KASIH

ANY QUESTIONS?