



Perulangan Lanjut

Tim Olimpiade Komputer Indonesia

Pendahuluan

Melalui dokumen ini, kalian akan:

- Memahami penggunaan perulangan yang bersarang.
- Memecahkan beberapa persoalan dengan perulangan.



Bagian 1

Perulangan Bersarang



Motivasi: Pola 0

- Pak Dengklek akan memberikan sebuah bilangan, misalnya **N**.
- Anda diminta untuk mencetak karakter bintang (*) yang tersusun N baris.
- Contoh untuk $N = 3$:

*
*
*



Motivasi (lanj.)

- Tentu saja solusinya sederhana, cukup gunakan salah satu struktur perulangan yang kalian kuasai.
- Misalnya menggunakan for:

```
for (int i = 0; i < N; i++) {  
    printf("\n");  
}
```



Motivasi: Pola 1

- Kemudian Pak Dengklek memberikan persoalan yang sedikit lebih sulit.
- Diberikan dua bilangan, misalnya N dan M .
- Cetak karakter bintang (*) yang tersusun N baris dan M kolom!
- Contoh untuk $N = 3$ dan $M = 5$:

```
*****  
*****  
*****
```

- Kali ini, untuk setiap barisnya kita perlu melakukan perulangan untuk mencetak M karakter bintang!



Contoh Program: pola1_1.cpp

- Kita bisa membuat "for di dalam for", sehingga membentuk struktur yang bersarang.
-

```
#include <stdio>

int main() {
    int N, M;
    scanf("%d %d", &N, &M);

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            printf("*");
        }
        printf("\n");
    }
}
```



Contoh Program: pola1_2.cpp

- Tentu saja kita bisa melakukannya dengan struktur perulangan yang lain:
-

```
#include <stdio>
```

```
int main() {  
    int N, M;  
    scanf("%d %d", &N, &M);  
  
    int i = 0;  
    while (i < N) {  
        int j = 0;  
        while (j < M) {  
            printf("*");  
            j++;  
        }  
        i++;  
        printf("\n");  
    }  
}
```



Contoh Lain: Pola 2

- Soal "Pola 1" dapat diselesaikan dengan mudah. Dengan demikian Pak Dengklek memberikan soal yang lebih menantang.
- Diberikan sebuah bilangan, misalnya N .
- Cetak "struktur segitiga rata kiri" yang terdiri dari N baris.
- Misalnya untuk $N = 5$, hasilnya adalah:

```
*  
**  
***  
****  
*****
```



Contoh Solusi: pola2.cpp

- Berikut ini adalah contoh solusinya, dimodifikasi dari pola1_1.cpp:

```
#include <stdio>
```

```
int main() {  
    int N;  
    scanf("%d", &N);  
  
    for (int i = 0; i < N; i++) {  
        for (int j = 0; j <= i; j++) {  
            printf("*");  
        }  
        printf("\n");  
    }  
}
```



Latihan: Pola 3

- Pak Dengklek kemudian memberikan tugas yang lebih sulit lagi, yang kali ini perlu Anda kerjakan sendiri.
- Diberikan sebuah bilangan, misalnya N.
- Cetak "struktur segitiga rata kanan" yang terdiri dari N baris.
- Misalnya untuk $N = 5$, hasilnya adalah:

```
  *
 **
 ***
 ****
 *****
```

- Petunjuk: cetak bagian kiri terlebih dahulu!



Bagian 2

Break & Continue



Break & Continue

- Kadang kala, kita membutuhkan suatu perulangan untuk diberhentikan secara paksa atau lompat ke iterasi berikutnya.
- C++ menyediakan kedua fitur tersebut, yaitu dengan kata kunci **break** dan **continue**.



Break & Continue (lanj.)

Break

Penggunaan **break** akan membuat program keluar dari perulangan yang mengandung kata kunci tersebut.

Continue

Penggunaan **continue** akan membuat program kembali ke baris awal perulangan, yaitu baris "for", atau "while".



Contoh Soal: Berhitung 1

- Setelah mahir dalam menggambar pola, kini Pak Dengklek ingin mengajar tentang berhitung.
- Pak Dengklek akan memberikan dua bilangan, yaitu N dan M .
- Anda diminta untuk menuliskan bilangan dari 1 sampai dengan N . Namun, ketika bilangan yang hendak ditulis adalah M , jangan cetak bilangan itu dan jangan cetak bilangan apapun lagi.
- Setelah selesai mencetak bilangan, cetak "selesai".
- Contoh untuk $N = 10$ dan $M = 5$:

1
2
3
4
selesai



Contoh Program: break.cpp

- Berikut ini adalah contoh solusi dari soal "Berhitung 1".

```
#include <stdio>

int main() {
    int N, M;
    scanf("%d %d", &N, &M);

    for (int i = 1; i <= N; i++) {
        if (i == M) {
            break;
        }

        printf("%d\n", i);
    }
    printf("selesai\n");
}
```



Penjelasan Program: break.cpp

- Ketika *break* ditemui, perulangan "for" akan diberhentikan secara paksa dan lanjut mengeksekusi perintah selanjutnya, yaitu mencetak tulisan "selesai".



Contoh Soal: Berhitung 2

- Kali ini Pak Dengklek mengubah soalnya: diberikan dua bilangan, yaitu N dan M .
- Anda diminta untuk menuliskan bilangan dari 1 sampai dengan N . Namun, ketika bilangan yang hendak ditulis adalah **kelipatan** dari M , jangan cetak bilangan itu.
- Setelah selesai mencetak bilangan, cetak "selesai".
- Contoh untuk $N = 10$ dan $M = 2$:

1
3
5
7
9
selesai



Contoh Program: continue.cpp

- Berikut ini adalah contoh solusi dari soal "Berhitung 2".

```
#include <stdio>
```

```
int main() {  
    int N, M;  
    scanf("%d %d", &N, &M);  
  
    for (int i = 1; i <= N; i++) {  
        if (i % M == 0) {  
            continue;  
        }  
  
        printf("%d\n", i);  
    }  
    printf("selesai\n");  
}
```



Penjelasan Program: `continue.cpp`

- Ketika *continue* ditemui, eksekusi perintah di dalam "for" untuk *i* tersebut langsung dilewati dan lanjut ke bagian perubahan.
- Artinya, untuk $N = 10$ dan $M = 2$, ketika nilai $i = 2$ dan "continue" ditemui, eksekusi akan dilewati langsung ke bagian perubahan $i++$.
- Selanjutnya, perulangan dilanjutkan pada $i = 3$.



Contoh Soal: Tes Keprimaan

- Diberikan sebuah bilangan positif yang lebih dari 1, misalnya N .
- Suatu bilangan N dikatakan prima apabila N positif dan hanya habis dibagi oleh 1 dan dirinya sendiri.
- Jika N prima, cetak " $\langle N \rangle$ adalah bilangan prima" dan jika tidak, cetak " $\langle N \rangle$ bukan bilangan prima".

Bagaimanakah kalian akan menyelesaikan persoalan ini?



Solusi 1

- Salah satu solusi yang sederhana adalah: periksa semua bilangan di antara 2 sampai dengan $N-1$.
- Jika ada setidaknya satu bilangan yang habis membagi N , artinya N bukan prima.



Solusi 1: prima1_1.cpp

```
#include <stdio>

int main() {
    int N;
    scanf("%d", &N);

    bool prima = true;
    for (int i = 2; i <= N-1; i++) {
        if (N % i == 0) {
            prima = false;
        }
    }

    if (prima) {
        printf("%d adalah bilangan prima\n", N);
    } else {
        printf("%d bukan bilangan prima\n", N);
    }
}
```



Solusi 2

- Solusi 1 melakukan pemeriksaan dari 2 sampai dengan $N-1$, artinya dibutuhkan pemeriksaan sebanyak $N-2$ kali.
- Sebetulnya pemeriksaan bisa dihentikan ketika ditemukan setidaknya satu saja bilangan yang habis membagi N .
- Dengan demikian bisa digunakan `break` untuk memberhentikan perulangan begitu ditemukan bilangan yang habis membagi N .



Solusi 2: prima1_2.cpp

```
#include <stdio>

int main() {
    int N;
    scanf("%d", &N);

    bool prima = true;
    for (int i = 2; i <= N-1; i++) {
        if (N % i == 0) {
            prima = false;
            break;
        }
    }

    if (prima) {
        printf("%d adalah bilangan prima\n", N);
    } else {
        printf("%d bukan bilangan prima\n", N);
    }
}
```



Contoh Soal: Pembangkit Prima

- Diberikan sebuah bilangan bulat N . Pak Dengklek meminta Anda untuk menuliskan N bilangan prima pertama.
- Contoh untuk $N = 5$:

2
3
5
7
11



Solusi: Pembangkit Prima

- Salah satu strategi yang dapat kalian gunakan adalah "selama belum ditemukan N bilangan prima, cari bilangan prima!".
- Bagaimana mencari bilangan prima? Coba saja dari 2, 3, 4, dan seterusnya sampai ditemukan N bilangan prima.



Contoh Solusi: prima2.cpp

```
#include <stdio>

int main() {
    int N;
    scanf("%d", &N);

    int count = 0; // Banyaknya prima yang sudah ditemukan
    int cur = 2; // nilai yang akan diperiksa keprimaannya

    while (count < N) {
        bool prima = true;
        for (int i = 2; i <= cur-1; i++) {
            if (cur % i == 0) {
                prima = false;
                break;
            }
        }
    }
```



Contoh Solusi: prima2.cpp (lanj.)

```
if (prima) {  
    // Ditemukan prima!  
    // Cetak dan tambahkan prima yg sudah ditemukan  
    printf("%d\n", cur);  
    count++;  
}  
  
// Entah ini prima atau bukan, lanjut untuk  
// memeriksa bilangan berikutnya  
cur++;  
}  
// Keluar dari while, dipastikan count = N  
}
```



Penutup

- Percabangan dan perulangan merupakan dua struktur kontrol yang sangat penting pada pemrograman.
- Kalian diharapkan berlatih sampai lancar di kedua hal tersebut, baru lanjut untuk mempelajari materi selanjutnya.

