

# **LAPORAN TENTANG PERULANGAN**



Oleh:

Nama : L HAFIDL ALKHAIR  
NIM : 2023903430060  
Kelas : TRKJ 1.C  
Jurusan : Teknologi Informasi dan Komputer  
Program Studi : Teknologi Rekayasa Komputer Jaringan  
Dosen Pembimbing : Indrawati, SST. MT



**JURUSAN TEKNOLOGI, KOMPUTER, DAN INFORMASI  
PRODI TEKNOLOGI REKAYASA KOMPUTER DAN JARINGAN  
POLITEKNIK NEGERI LHOKSEUMAWE  
TAHUN AJARAN 2022/2023**

---

### **LEMBAR PENGESAHAN**

No. Praktikum : 03 /TIK/TRKJ-1C/ Data Structure And Algorithms Practice

Judul : Laporan Tentang Perulangan

Nama : L HAFIDL ALKHAIR

NIM : 2023903430060

Kelas : TRKJ-1C

Jurusan : Teknologi Informasi Dan Komputer

Prodi : Teknologi Rekayasa Komputer Jaringan

Tanggal Praktikum : 25 September 2023

Tanggal Penyerahan : 2 Oktober 2023

Buketrata, 2 Oktober 2023

Dosen Pembimbing,

Indrawati, SST.MT

Nip. 19740815 200112 2 001

---

## DAFTAR ISI

|                              |    |
|------------------------------|----|
| A. Tujuan .....              | 4  |
| B. Dasar teori.....          | 4  |
| C. Alat dan bahan .....      | 5  |
| D. Perulangan ‘For.....      | 5  |
| E. Perulangan ‘While .....   | 9  |
| F. Perulangan do-while ..... | 12 |
| BAB II.....                  | 17 |
| PENUTUP .....                | 17 |
| Kesimpulan .....             | 17 |
| DAFTAR PUSTAKA .....         | 18 |

---

## A. Tujuan

1. Mahasiswa dapat menghemat waktu dan upaya karena kita tidak perlu menulis perintah berulang kali.
2. Memungkinkan tugas-tugas yang sama diulang dengan sedikit perubahan pada setiap iterasi.
3. Membuat kode lebih mudah dipahami dengan ekspresi yang jelas tentang tindakan yang diulang.
4. Berguna untuk bekerja dengan koleksi data seperti array atau daftar.
5. Dapat mengatur alur eksekusi program berdasarkan kondisi tertentu.

## B. Dasar teori

Perulangan dalam bahasa C adalah konsep yang memungkinkan kita untuk menjalankan sekelompok perintah atau instruksi berulang kali selama kondisi tertentu terpenuhi. Ada tiga jenis perulangan yang umum digunakan di C:

- Perulangan for: Perulangan for digunakan ketika kita tahu berapa kali iterasi akan dilakukan. Ini terdiri dari tiga bagian: inisialisasi variabel penghitung, kondisi untuk menjalankan perulangan, dan perubahan variabel penghitung setiap iterasi.
- Perulangan while: Perulangan while digunakan ketika kita ingin menjalankan perintah selama kondisi tertentu terpenuhi. Perintah akan dieksekusi selama kondisi bernilai benar (non-nol).
- Perulangan do-while: Perulangan do-while mirip dengan while, tetapi perintah di dalam blok akan dieksekusi setidaknya satu kali

---

sebelum kondisi diperiksa. Kemudian, perulangan akan berlanjut jika kondisi masih terpenuhi.

### C. Alat dan bahan

Di dalam pembuatan sebuah Bahasa pemrograman tentu saja memerlukan alat dan bahan-bahan supaya program tersebut dapat di jalankan dengan baik dan tidak mengalami yang namanya error saat program atau kode tersebut di jalankan. Adapun alat dan bahan sebagai berikut:

- a. Laptop atau device



- b. Tools atau aplikasi pemrograman (disini kita memakai aplikasi Dev-C++)



- c. Reverensi ataupun ide dalam membuat sebuah program

### D. Perulangan 'For

---

Contoh perulangan 'For : Mencetak Angka 1 Sampai 10

```
1  #include <stdio.h>
2
3  int main() {
4      for (int i = 1; i <= 10; i++) {
5          printf("%d ", i);
6      }
7
8      return 0;
9  }
```

Hasil Program Sebagai Berikut :

```
1 2 3 4 5 6 7 8 9 10
-----
```

Penjelasan dari Program di atas :

- Perulangan for ini digunakan untuk mencetak angka dari 1 hingga 10.
- Variabel i digunakan sebagai variabel penghitung.
- Perulangan akan berjalan selama i kurang dari atau sama dengan 10.
- Setiap iterasi, nilai i akan bertambah satu, dan angka tersebut akan dicetak ke layar.

Contoh kedua 'For : Menghitung Angka Ganjil 1 – 100

```
1  #include <stdio.h>
2
3  int main() {
4      int sum = 0;
5
6      for (int i = 1; i <= 100; i++) {
7          if (i % 2 != 0) {
8              sum += i;
9          }
10     }
11
12     printf("Jumlah bilangan ganjil dari 1 hingga 100: %d\n", sum);
13
14     return 0;
15 }
```

Hasil Program :

```
Jumlah bilangan ganjil dari 1 hingga 100: 2500
-----
```

Analisa Program :

- Perulangan for ini digunakan untuk menjalankan iterasi dari 1 hingga 100.
- Pada setiap iterasi, kita memeriksa apakah nilai *i* adalah bilangan ganjil dengan menggunakan operasi modulo ( $i \% 2 \neq 0$ ).
- Jika nilai *i* adalah bilangan ganjil, kita menambahkannya ke variabel *sum*.
- Setelah perulangan selesai, kita mencetak hasil jumlah bilangan ganjil dari 1 hingga 100.

Contoh ketiga : Membuat Segitiga Angka

```
1  #include <stdio.h>
2
3  int main() {
4      int tinggi;
5
6      printf("Masukkan tinggi segitiga: ");
7      scanf("%d", &tinggi);
8
9      for (int i = 1; i <= tinggi; i++) {
10         for (int j = 1; j <= i; j++) {
11             printf("%d ", j);
12         }
13         printf("\n");
14     }
15
16     return 0;
17 }
```

Hasil Program :

```
Masukan tinggi segitiga :20
1
12
123
1234
12345
123456
1234567
12345678
123456789
12345678910
1234567891011
123456789101112
12345678910111213
1234567891011121314
123456789101112131415
12345678910111213141516
1234567891011121314151617
123456789101112131415161718
12345678910111213141516171819
1234567891011121314151617181920
-----
```



Analisa :

- Program ini meminta pengguna memasukkan tinggi segitiga.
- Kemudian, terdapat dua perulangan for bersarang. Perulangan pertama (i) digunakan untuk mengontrol baris-baris dari segitiga.
- Perulangan kedua (j) digunakan untuk mencetak angka-angka dalam setiap baris.
- Pada setiap baris, angka-angka dari 1 hingga nomor baris tersebut dicetak dengan spasi di antaranya. Setelah satu baris selesai, kita mencetak baris baru (\n) untuk membuat bentuk segitiga.

## E. Perulangan ‘While

Contoh program ; mencetak angka 1 sampai 5

```
1  #include <stdio.h>
2
3  int main() {
4      int i = 1;
5
6      while (i <= 5) {
7          printf("%d ", i);
8          i++;
9      }
10
11     return 0;
12 }
```

Hasil program

```
1 2 3 4 5
-----
```

---

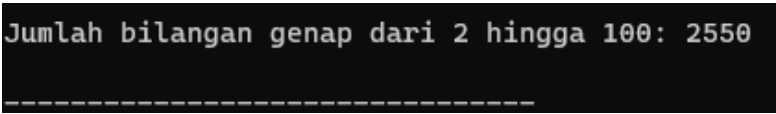
### Analisa program

- Pada contoh ini, kita menggunakan perulangan while untuk mencetak angka dari 1 hingga 5.
- Variabel i digunakan sebagai penghitung awal dengan nilai 1.
- Perulangan akan terus berjalan selama kondisi  $i \leq 5$  benar (true).
- Setiap iterasi mencetak nilai i ke layar dan kemudian variabel i ditingkatkan sebesar 1.

### Contoh 2: Menghitung Jumlah Bilangan Genap dari 2 hingga 100

```
1  #include <stdio.h>
2
3  int main() {
4      int sum = 0;
5      int i = 2;
6
7      while (i <= 100) {
8          sum += i;
9          i += 2;
10     }
11
12     printf("Jumlah bilangan genap dari 2 hingga 100: %d\n", sum);
13
14     return 0;
15 }
```

### Hasil program



```
Jumlah bilangan genap dari 2 hingga 100: 2550
```

---

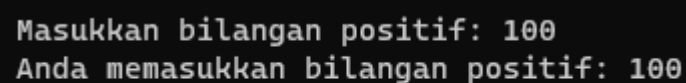
### Analisa program

- Pada contoh ini, kita menggunakan perulangan while untuk menghitung jumlah bilangan genap dari 2 hingga 100.
- Variabel sum digunakan untuk menyimpan hasil penjumlahan, dan variabel i digunakan sebagai penghitung awal dengan nilai 2.
- Perulangan akan berlanjut selama i kurang dari atau sama dengan 100.
- Pada setiap iterasi, nilai i (bilangan genap) ditambahkan ke sum, dan kemudian i ditingkatkan sebesar 2.

### Contoh 3: Input Validasi Bilangan Positif

```
1  #include <stdio.h>
2
3  int main() {
4      int bilangan;
5
6      printf("Masukkan bilangan positif: ");
7      scanf("%d", &bilangan);
8
9      while (bilangan <= 0) {
10         printf("Bilangan harus positif. Masukkan lagi: ");
11         scanf("%d", &bilangan);
12     }
13
14     printf("Anda memasukkan bilangan positif: %d\n", bilangan);
15
16     return 0;
17 }
```

### Hasil program



Masukkan bilangan positif: 100  
Anda memasukkan bilangan positif: 100  
-----

---

## Analisa Program

- Pada contoh ini, kita menggunakan perulangan while untuk memvalidasi input pengguna sehingga hanya bilangan positif yang diterima.
- Pengguna diminta untuk memasukkan sebuah bilangan, dan input tersebut disimpan dalam variabel bilangan.
- Perulangan while akan berlanjut selama bilangan kurang dari atau sama dengan 0.
- Jika pengguna memasukkan bilangan negatif atau nol, program akan menampilkan pesan kesalahan dan meminta input lagi hingga pengguna memasukkan bilangan positif.

Dengan menggunakan perulangan while, kita dapat mengendalikan alur program berdasarkan kondisi yang diberikan, sehingga memungkinkan untuk melakukan berbagai tugas seperti pencetakan, perhitungan, dan validasi input.

## F. Perulangan do-while

### Contoh 1: Input Validasi Bilangan Positif

```
1  #include <stdio.h>
2
3  int main() {
4      int bilangan;
5
6      do {
7          printf("Masukkan bilangan positif: ");
8          scanf("%d", &bilangan);
9      } while (bilangan <= 0);
10
11     printf("Anda memasukkan bilangan positif: %d\n", bilangan);
12
13     return 0;
14 }
```

---

Hasil program

```
Masukkan bilangan positif: -10
Masukkan bilangan positif: -10
Masukkan bilangan positif: 10
Anda memasukkan bilangan positif: 10
-----
```

### Analisa Program

- Pada contoh ini, kita menggunakan perulangan do-while untuk memvalidasi input pengguna sehingga hanya bilangan positif yang diterima.
- Pengguna diminta untuk memasukkan sebuah bilangan, dan input tersebut disimpan dalam variabel bilangan.
- Perulangan do-while akan selalu menjalankan setidaknya satu iterasi (pertama kali) tanpa memeriksa kondisi. Kemudian, setelah itu, kondisi bilangan  $\leq 0$  diperiksa.
- Jika pengguna memasukkan bilangan negatif atau nol, program akan menampilkan pesan kesalahan dan meminta input lagi hingga pengguna memasukkan bilangan positif.

### Contoh 2: Menjumlahkan Angka hingga Pengguna Memilih untuk Berhenti

```
1  #include <stdio.h>
2
3  int main() {
4      int sum = 0;
5      int pilihan;
6
7      do {
8          printf("Masukkan angka: ");
9          scanf("%d", &pilihan);
10         sum += pilihan;
11     } while (pilihan != 0);
12
13     printf("Total jumlah angka yang dimasukkan: %d\n", sum);
14
15     return 0;
16 }
```

---

## Hasil Program

```
Masukkan angka: 20
Masukkan angka: 20
Masukkan angka: 10
Masukkan angka: 50
Masukkan angka: 100
Masukkan angka: 0
Total jumlah angka yang dimasukkan: 200
-----
```

## Analisa Program

- Pada contoh ini, kita menggunakan perulangan do-while untuk meminta pengguna memasukkan angka-angka dan menjumlahkannya hingga pengguna memilih untuk berhenti (dengan memasukkan 0).
- Variabel sum digunakan untuk menyimpan hasil penjumlahan.
- Perulangan do-while akan selalu menjalankan setidaknya satu iterasi (pertama kali) tanpa memeriksa kondisi. Kemudian, setelah itu, kondisi pilihan `!= 0` diperiksa.
- Jika pengguna memasukkan angka selain 0, angka tersebut ditambahkan ke sum. Perulangan akan terus berlanjut sampai pengguna memasukkan 0.

### Contoh 3: Menampilkan Menu Pilihan

```
1  #include <stdio.h>
2
3  int main() {
4      int pilihan;
5
6      do {
7          printf("Menu:\n");
8          printf("1. Cetak 'Hello'\n");
9          printf("2. Keluar\n");
10         printf("Pilih menu: ");
11         scanf("%d", &pilihan);
12
13         switch (pilihan) {
14             case 1:
15                 printf("Hello\n");
16                 break;
17             case 2:
18                 printf("Keluar dari program.\n");
19                 break;
20             default:
21                 printf("Pilihan tidak valid.\n");
22                 break;
23         }
24     } while (pilihan != 2);
25
26     return 0;
27 }
```

---

## Hasil Program

```
Menu:
1. Cetak 'Hello'
2. Keluar
Pilih menu: 1
Hello
Menu:
1. Cetak 'Hello'
2. Keluar
Pilih menu: 1
Hello
Menu:
1. Cetak 'Hello'
2. Keluar
Pilih menu: 2
Keluar dari program.
```

## Analisa Program

- Pada contoh ini, kita menggunakan perulangan do-while untuk menampilkan menu pilihan kepada pengguna dan mengambil input mereka untuk menjalankan aksi yang sesuai.
- Program akan terus menampilkan menu dan meminta input hingga pengguna memilih untuk keluar (memasukkan 2).
- Perulangan do-while selalu menjalankan setidaknya satu iterasi pertama kali tanpa memeriksa kondisi. Kemudian, kondisi pilihan `!= 2` diperiksa.
- Pengguna dapat memilih untuk mencetak "Hello" atau keluar dari program dengan memasukkan nomor menu yang sesuai. Jika pengguna memasukkan pilihan yang tidak valid, pesan kesalahan akan ditampilkan.

Dengan menggunakan perulangan do-while, kita dapat menjalankan kode setidaknya sekali sebelum memeriksa kondisi, yang berguna untuk tugas-tugas seperti validasi input atau tampilan menu interaktif kepada pengguna.



---

## BAB II

### PENUTUP

#### Kesimpulan

Ketiga contoh di atas menunjukkan fleksibilitas perulangan dalam bahasa C untuk berbagai tugas pemrograman. Perulangan `for` cocok untuk mengulangi tugas dengan jumlah iterasi yang telah diketahui sebelumnya, perulangan `while` berguna untuk mengulangi tugas hingga kondisi tertentu terpenuhi, dan perulangan `do-while` berguna untuk memastikan setidaknya satu iterasi dilakukan sebelum memeriksa kondisi.

Pemahaman yang baik tentang penggunaan perulangan adalah kunci dalam pengembangan perangkat lunak yang efisien dan dapat diandalkan. Dalam kasus-kasus ini, perulangan digunakan untuk mencetak, menghitung, dan memvalidasi input, yang merupakan tugas umum dalam pemrograman. Perulangan membantu mengorganisasi alur program dengan baik dan mengurangi penulisan kode yang berlebihan.

Kesimpulannya, perulangan adalah alat penting dalam pemrograman yang memungkinkan kita mengotomatisasi tugas berulang, meningkatkan efisiensi, dan mengelola input pengguna dengan baik. Dalam bahasa C, pemahaman yang kuat tentang perulangan `for`, `while`, dan `do-while` akan membantu Anda mengembangkan aplikasi yang lebih kuat dan efisien.

---

## DAFTAR PUSTAKA

<https://www.petanikode.com/c-perulangan/>

<https://www.duniailkom.com/tutorial-belajar-c-perulangan-for-bahasa-c/>

<https://socs.binus.ac.id/2017/03/20/perulangan-dalam-bahasa-c/>

<https://www.sobatambisius.com/2021/09/belajar-bahasa-c-5-perulanganlooping.html>

<https://www.sobatambisius.com/2021/09/belajar-bahasa-c-5-perulanganlooping.html>

<https://www.studocu.com/id/document/universitas-muhammadiyah-malang/pemrograman-dasar/perulangan-bahasa-c/34600065>

[https://id.wikibooks.org/wiki/Pemrograman\\_C/Perulangan](https://id.wikibooks.org/wiki/Pemrograman_C/Perulangan)