

Laporan Resmi
Praktikum Algoritma dan Struktur Data
Rekursif



Dr. Tita Karlita S.Kom, M.Kom

Nama : Marits Ikmal Yasin
Kelas : 1D4 IT B
NRP : 3121600047

1. rekursi menampilkan bil n sd 0

Kode :

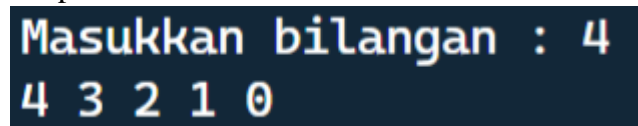
```
#include <stdio.h>
```

```
void mundur(int bil)
{
    if (bil >= 0)
    {
        printf("%d ", bil);
        mundur(bil - 1);
    }
}
```

```
int main()
{
    int bilangan;

    printf("Masukkan bilangan : ");
    scanf("%d", &bilangan);
    mundur(bilangan);
    return 0;
}
```

Output :



```
Masukkan bilangan : 4
4 3 2 1 0
```

2. rekursi menampilkan bil 0 sd n

Kode :

```
#include <stdio.h>
```

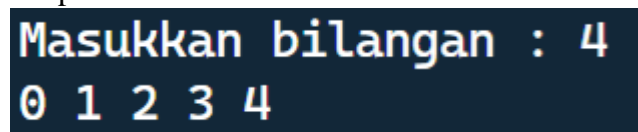
```
void maju(int bil)
{
    int static number = 0;

    if (bil >= 0)
    {
        printf("%d ", number++);
        maju(bil - 1);
    }
}
```

```
int main()
{
    int bilangan;

    printf("Masukkan bilangan : ");
    scanf("%d", &bilangan);
    maju(bilangan);
    return 0;
}
```

Output :



```
Masukkan bilangan : 4
0 1 2 3 4
```

3. Membalik Kata

Kode :

```
#include <stdio.h>
```

```

#include <stdlib.h>
#include <string.h>
#define MAKS 25

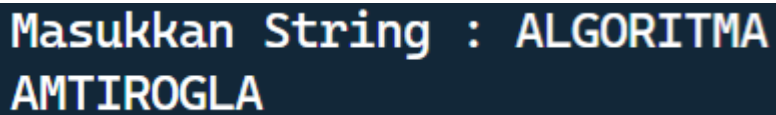
void balik(char kalimat[], int panjang)
{
    if (panjang > 0)
    {
        printf("%c", kalimat[panjang - 1]);
        balik(kalimat, --panjang);
    }
}

int main()
{
    char string[MAKS];

    printf("Masukkan String : ");
    gets(string);
    balik(string, strlen(string));
    return 0;
}

```

Output :



Masukkan String : ALGORITMA
AMTIROGLA

4. Faktorial (Perbandingan Waktu)

Kode :

```

#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>

int iterasi(int bil)
{
    int temp = 1;

    for (int i = 1; i <= bil; i++)
    {
        usleep(1000);
        temp *= i;
    }
    return temp;
}

int rekursi(int bil)
{
    usleep(1000);
    if (bil < 0)
        return 0;
    else if (bil == 1 || bil == 0)
        return 1;
    else
        return bil * rekursi(bil - 1);
}

int rekursiTail(int bil, int a)

```

```

{
    usleep(1000);
    if (bil < 0)
        return 0;
    else if (bil == 1 || bil == 0)
        return a;
    else
        return rekursiTail(--bil, a * bil);
}

void menu()
{
    puts("1. Iterasi");
    puts("2. Rekursi");
    puts("3. Rekursi Tail");
}

int main()
{
    int pilihan, bilangan;
    struct timeval stop, start;

    menu();
    printf("Masukkan pilihan : ");
    scanf("%d", &pilihan);
    printf("Masukkan bilangan : ");
    scanf("%d", &bilangan);
    // clock_t start = clock();
    gettimeofday(&start, NULL);
    switch (pilihan)
    {
    case 1:
        printf("%d != %d\n", bilangan, iterasi(bilangan));
        break;
    case 2:
        printf("%d != %d\n", bilangan, rekursi(bilangan));
        break;
    case 3:
        printf("%d != %d\n", bilangan, rekursiTail(bilangan, 1));
        break;
    default:
        puts("input yang benar");
    }
    gettimeofday(&stop, NULL);
    printf("took %lu us\n", ((stop.tv_sec - start.tv_sec) * 1000000 + stop.tv_usec -
start.tv_usec));
    // clock_t finish = clock();
    // double selisih = (double)(finish - start) / CLOCKS_PER_SEC;
    // printf("Ini selisih %lf\n", selisih);
    return 0;
}

```

Output :

Pilihan 1 :

```
1. Iterasi
2. Rekursi
3. Rekursi Tail
Masukkan pilihan : 1
Masukkan bilangan : 5
5 ! = 120
took 71707 us
```

Pilihan 2 :

```
1. Iterasi
2. Rekursi
3. Rekursi Tail
Masukkan pilihan : 2
Masukkan bilangan : 5
5 ! = 120
took 73779 us
```

Pilihan 3 :

```
1. Iterasi
2. Rekursi
3. Rekursi Tail
Masukkan pilihan : 3
Masukkan bilangan : 5
5 ! = 120
took 66304 us
```

5. Permutasi dan Kombinasi

Kode :

```
#include <stdio.h>
```

```
int fakt(int);
int permutasi(int, int);
int kombinasi(int, int);
```

```
int main()
{
```

```
    int n, r;
```

```
    printf("n = ");
    scanf("%d", &n);
    printf("r = ");
    scanf("%d", &r);
    printf("P(%d, %d) = %d\n", n, r, permutasi(n, r));
    printf("C(%d, %d) = %d\n", n, r, kombinasi(n, r));
```

```

    return 0;
}

int fakt(int x)
{
    if (x < 0)
        return 0;
    else if (x == 0 || x == 1)
        return 1;
    else
        return x * fakt(x - 1);
}

int permutasi(int n, int r)
{
    return fakt(n) / fakt(n - r);
}

int kombinasi(int n, int r)
{
    return fakt(n) / (fakt(r) * fakt(n - r));
}

```

Output :

```

n = 3
r = 2
P(3, 2) = 6
C(3, 2) = 3

```

Analisa :

Rekursif merupakan perulangan menggunakan fungsi di mana fungsi itu memanggil fungsinya sendiri. Agar perulangan itu Berhenti tentunya kita membutuhkan kondisi agar rekursif itu berhenti. Oleh karena itu perulangan rekursif menggunakan if. Sedangkan perulangan menggunakan for, while, ataupun do while sudah memiliki batasan untuk berhenti seperti apa. Oleh karena itu perulangan menggunakan looping function lebih mudah untuk dipahami dan tentunya lebih mudah untuk debugging. Contoh penggunaan rekurdif bisa dilakukan untuk program menampilkan bilangan dari 0 ke n maupun sebaliknya. Selain itu juga bisa digunakan untuk program membalik kata dan juga factorial. Sebenarnya banyak program yang bisa dibuat menggunakan rekursif, tetapi lebih sulit untuk menggunakannya karena perlu ketelitian yang lebih. Mengenai performa maka perulangan menggunakan function builtin pasti akan lebih cepat ketimbang menggunakan rekursif. Selain itu lebih menghemat memori daripada rekursif.