

1 D4 - TEKKOM B

LAPORAN RESMI MENGEMBANGKAN FUNGSI-FUNGSI SENDIRI



Nama : Septian Bagus Jumanoro
Kelas : 1 – D4 Teknik Komputer B
NRP : 3221600039
Dosen : Ir Sigit Wasista, M.Kom.
Mata Kuliah : Praktikum Pemrograman Dasar 1
Hari/Tgl. Praktikum : 08 Oktober 2021

PERCOBAAN 6

6.1 JUDUL: MENGEMBANGKAN FUNGSI-FUNGSI SENDIRI

6.2 TUJUAN: Mempelajari pembuatan fungsi-fungsi diluar fungsi-fungsi yang ada, serta mempelajari penggunaan variable global, local, eksternal, otomatis, statis dan register.

6.3 TEORI: Suatu fungsi adalah bagian program yang terpisah dari fungsi program utamanya `main()` dan berdiri sendiri untuk mengerjakan suatu proses tertentu. Suatu fungsi dapat digunakan sebagai fungsi itu sendiri, subrutin dan prosedur seperti istilah yang digunakan pada bahasa pemrograman BASIC atau PASCAL. Tujuan digunakannya fungsi adalah untuk mencegah penggunaan program secara berulang-ulang, sehingga program menjadi sangat panjang. Pada percobaan-percobaan diatas telah digunakan fungsi-fungsi dari C, seperti `printf()`, `clrscr()`, `scanf()`, `getc ()`, `puts()`, `for()` dan sebagainya, maka disini dicoba untuk membuat fungsi baru dimana tidak ada dalam pustaka bahasa C (Turbo C). Secara aturan fungsi baru dibuat setelah fungsi `main()` dibuat, sehingga penempatan fungsi adalah dibawah fungsi `main()`, tatapi ada kalanya dapat diletakkan diatas fungsi `main()`, seperti fungsi-fungsi yang ada dalam `#include`. Bila fungsi diletakkan diatas fungsi `main()` maka tidak diperlukan deklarasi fungsi lagi.

6.4 PROGRAM PERCOBAAN

6.4.1 Untuk membuat fungsi maka disiapkan nama fungsi `cetak_pesan()`, kemudian bila diletakkan dibawah fungsi `main()`, maka fungsi tersebut harus dideklarasikan diatas fungsi `main()`. Fungsi ini dipanggil 5 kali.

```
/* Nama File : FUNGSI1.C */

void cetak_pesan(void);

void main()
{
    int i;

    for(i=1; i<=5; i++) {
        printf("Pesan ke-%d : ",i);
        cetak_pesan();
    }
    getc ();
}

void cetak_pesan()
{
    printf("Ini dulu! baru itu!\n\n");
}
```

6.4.2 Peletakan suatu fungsi dapat diatas fungsi `main()`, sehingga tidak diperlukan deklarasi fungsi. Masukan suatu fungsi dilewatkan melalui suatu argument yang disebut passing parameter. Fungsi `hitung_triangular()` dibawah ini mempunyai argument `int n`, yang digunakan untuk melewatkan suatu nilai dari program pemanggil ke dalam suatu fungsi.

```
/* Nama File : FUNGSI2.C */

void hitung_triangular(int n)
{
    int i, jumla =0;

    for (i=1, ic=n; ++i)
        jumla =jumla + i;
    printf("Jumla triangular %d adala %d\n", n, jumla );
}

void main()
{
    hitung_triangular(10);
    hitung_triangular(20);
    hitung_triangular(50);
   getc ();
}
```

6.4.3 Fungsi berikut menghasilkan Faktor Persekutuan TerBesar (FPB)

```
/* Nama File : FUNGSI3.C */

void fpb(int, int);

void main()
{
    fpb(150, 35);
    fpb(1026, 405);
    fpb(83, 240);
   getc ();
}

void fpb(int u, int v)
{
    int tampung;

    printf("FPB dari %d dan %d adala ", u, v);
    while(v != 0) {
        tampung = u % v;
        u = v;
        v = tampung;
    }
    printf("%d\n", u);
}
```

6.4.4 Suatu fungsi dapat memberikan nilai balik (return value), bila fungsi tersebut mempunyai tipe dan bukannya void seperti fungsi-fungsi diatas, dimana digunakan sebagai prosedur saja (tidak mempunyai nilai balik). Pada fungsi `fpb()` terdapat tipe `int`, dimana fungsi ini akan memberikan nilai balik bertipe integer (bulat), dan untuk memberikan nilai balik ini maka didalam fungsi `fpb()` harus ditambahkan statement `return()`, yang digunakan untuk memberikan suatu nilai yang dikembalikan oleh fungsi.

```
/* Nama File : FUNGSI4.C */

#include cstdio. S
#include cconio. S

int fpb(int, int);

void main()
{
    int asil;

    clrscr();

    asil = fpb(150, 35);
    printf("FPB dari 150 dan 35 adala  %d\n",  asil);

    asil = fpb(1026, 405);
    printf("FPB dari 1026 dan 405 adala %d\n", asil);

    printf("FPB dari 83 dan 240 adala %d\n", fpb(83, 240));

    getch ();
}

int fpb(int u, int v)
{
    int tampung;

    while(v != 0) {
        tampung = u % v;
        u = v;
        v = tampung;
    }

    return(u);
}
```

6.4.5 Program dibawah ini terdapat beberapa fungsi, dimana fungsi satu memanggil fungsi yang lain. Fungsi akar() memanggil fungsi absolute() untuk menentukan nilai absolute dari nilai perhitungan akar, yang dihitung menggunakan iterasi sampai didapatkan nilai yang mendekati. Fungsi akar juga memeriksa nilai yang akan dicari akarnya, bila bernilai negative maka akan diberikan pesan kesalahan dan dikembalikan nilai -1.

```
/* Nama File : FUNGSI5.C */

#include cconio. S
#include cstdio. S

float absolut(x)
float x;
{
    if (x < 0)
        x = -x;

    return (x);
}

float akar(x)
float x;
{
    float epsilon = .00001;
    float guess = 1.0;

    if (x < 0) {
        printf("Argumen negatif!\n");
        return (-1.0);
    }

    while (absolut(guess * guess - x) >= epsilon)
        guess = (x / guess + guess) / 2.0;

    return (guess);
}

void main()
{
    clrscr();

    printf("akar (4.0)      = %f\n", akar(4.0));
    printf("akar (625.0)   = %f\n", akar(625.0));
    printf("akar (-39.5)   = %f\n", akar(-39.5));

    getch ();
}
```

6.4.6 Fungsi berikut digunakan untuk menentukan 2 bilangan yang terkecil.

```
/* Nama File : FUNGSI6.C */

float minimum(float x, float y)
{
    if (x < y)
        return(x);
    else
        return(y);
}

void main()
{
    float a, b;

    printf("Masukkan nilai a : "); scanf("%f", &a);
    printf("Masukkan nilai b : "); scanf("%f", &b);

    printf("\nBilangan terkecil antara %g dan %g adala ");
    printf("%g\n\n", a, b, minimum(a, b));

    getch ();
}
```

6.4.7 Program dibawah ini akan membedakan penggunaan variable otomatis dan variable statis, dimana bila menggunakan variable statis maka nilai akan tetap, sedangkan bila menggunakan variable otomatis maka nilai akan kembali seperti semula.

```
/* Nama File : FUNGSI7.C */

void demo(void)
{
    auto int var_auto = 0;
    static int var_static = 0;

    printf("auto = %d, static = %d\n", var_auto, var_static);
    ++var_auto;
    ++var_static;
}

main()
{
    int i=0;

    while(i < 3) {
        demo();
        i++;
    }
}
```

6.5 TUGAS-TUGAS (Dikumpulkan 1 minggu setelah praktikum dilaksanakan)

6.5.1 Buatlah fungsi kabisat() yang akan memberikan nilai balik 1 bila tahun adalah kabisat, dan 0 bila tahun bukan kabisat.

Source Code

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

kabisat(int x)
{
    if(x%100!=0 && x%4==0)
        return 1;

    else;
    return 0;
}

void main()
{
    int x;
    system("cls");
    printf("Program Penentu Tahun Kabisat\n");
    printf("Bernilai '1' jika tahun kabisat\n");
    printf("Bernilai '0' jika bukan tahun kabisat\n\n");
    printf("Masukkan tahun: ");
    scanf("%d", &x);
    printf("Nilai: %d\n", kabisat(x));
}
```

Output

```
Program Penentu Tahun Kabisat
Bernilai '1' jika tahun kabisat
Bernilai '0' jika bukan tahun kabisat

Masukkan tahun: 1900
Nilai: 0
PS C:\Users\Hp\Music\Bab 6> █
```

Analisa

Pada program tersebut berfungsi untuk menentukan apakah tahun yang diinputkan merupakan tahun kabisat atau bukan. Jika termasuk kabisat maka akan tercetak "1", jika bukan maka akan tercetak "0". Pada source code saya menggunakan logika matematika tahun kabisat. Lalu program diproses menggunakan statement if ($x \% 100 \neq 0 \ \&\& \ x \% 4 == 0$) maka $x = 1$, jika else maka $x = 0$.

6.5.2 Buatlah fungsi `factorial()` untuk menghitung nilai factorial, Fungsi `factorial()` memberikan nilai balik bertipe `long int` yang akan dicetak ke layar dalam fungsi `main()`.

Source Code

```
#include <stdio.h>
fact(long int x)
{
    int n, jumlah=1;
    for (n=1; n<=x; n++)
        jumlah=jumlah*n;
    return (jumlah);
}
void main()
{
    long int x;
    system("cls");
    printf("Program Fungsi Faktorial\n");
    printf("Masukkan angka: ");
    scanf("%d", &x);
    printf("\nNilai %d! adalah %d\n", x, fact(x));
}
```

Output

```
Program Fungsi Faktorial
Masukkan angka: 5
```

```
Nilai 5! adalah 120
PS C:\Users\Hp\Music\Bab 6> |
```

Analisa

Pada program tersebut berfungsi untuk menghitung bilangan factorial. Pada source code saya mendefinisikan `int n`, `jumlah = 1`. Lalu program diproses dengan `for(i=1; i<=n; i++)` lalu `jumlah *= i` dan akan return lagi ke `jumlah` dengan menggunakan `long int`. Lalu user memasukkan input dan disimpan pada variabel `x`. Lalu program di cetak dengan memanggil variabel `x` dan `fact(x)`.

6.5.3 Buatlah suatu fungsi `permutasi()` dan `kombinasi()` untuk menghitung permutasi dan kombinasi dari suatu pasangan bilangan, yang dinyatakan dengan formula:

$$\text{Permutasi} \quad : \quad P(n,r) = \frac{n!}{(n-r)!}$$

$$\text{Kombinasi} \quad : \quad C(n,r) = \frac{n!}{r!(n-r)!}$$

Source Code

```
#include <stdio.h>
float factorial(float bil)
{
    int a;
    float hasil=1;

    for(a=1; a<=bil; a++)
    {
        hasil*=a;
    }
    return (hasil);
}
float permutasi(float n, float r)
{
    float p;
    p = factorial(n) / factorial(n-r);
    return p;
}
float kombinasi(float n, float r)
{
    float c;
    c = factorial(n) / (factorial(r) * factorial(n-r));
    return c;
}
void main()
{
    int n,r;
    system("cls");
    printf("Nilai n: ");
    scanf("%d", &n);
    printf("Nilai r: ");
    scanf("%d", &r);
    printf("Nilai Permutasinya adalah : %g\n", permutasi(n,r));
    printf("Nilai Kombinasinya adalah : %g\n", kombinasi(n,r));
}
```

Output

```
Nilai n: 10
Nilai r: 5
Nilai Permutasinya adalah : 30240
Nilai Kombinasinya adalah : 252
PS C:\Users\Hp\Music\Bab 6> █
```

Anlisa

Pada program tersebut berfungsi untuk menghitung fungsi permutasi dan kombinasi. Pada source code sudah menggunakan float factorial (float bil). Lalu program diproses menggunakan looping for (a = 1; a <= bil; a++) maka hasil *= a. Lalu membuat fungsi baru untuk logika perhitungan permutasi menggunakan $p = \text{factorial}(n) / \text{factorial}(n-r)$. Lalu untuk logika perhitungan kombinasi menggunakan $c = \text{factorial}(n) / (\text{factorial}(r) * \text{factorial}(n-r))$. Lalu user memasukkan input untuk nilai n dan nilai r.

6.5.4 Buatlah fungsi float su u(s, d), untuk mengkonversikan suhu dari Celsius ke Fahrenheit, Celsius ke Reamur, Fahrenheit ke Celsius, Fahrenheit ke Reamur, Reamur ke Celsius, dan Reamur ke Fahrenheit. Dimana s adalah suhu sumber, d adalah suhu konversi, dan suhu() adalah hasil konversi suhu.

Contoh tampilan:

Masukkan su u sumber = 100 C

Masukkan su u tujuan = R

Hasil konversi su u = 80 R

Source Code

```
double suhu(float a, char s, char d)
{
    double i;
    switch(s)
    {
        case 'C':
        {
            if(d=='R')
            {
                i=a*4/5; break;
            }
            else
            {
                i=(a*9/5) + 32; break;
            }
        }
        case 'R':
        {
            if(d=='C')
            {
                i=a*5/4; break;
            }
            else
            {
                i=(a-32)*5/9; break;
            }
        }
    }
    return i;
}
```

```

        {
            i=(a*9/4) + 32; break;
        }
    }
    case 'F':
    {
        if(d=='C')
        {
            i=(a-32) * 5/9; break;
        }
        else{
            i=(a-32) * 4/9;
        }
    }
}
return i;
}
void main()
{
    int t;
    char s, d, yes='y';
    system("cls");
    while(yes=='y')
    {
        printf("Masukkan suhu (contoh: 100 C):");
        scanf("%d %c", &t, &s);
        printf("Masukkan tujuan suhu (C, R, F):");
        d = getche();
        printf("\n");
        if(d=='R')
        {
            printf("Hasil konversi suhu: %g Reamur", suhu(t, s, d));
        }
        else if(d=='F')
        {
            printf("Hasil konversi suhu: %g Fahrenheit", suhu(t, s, d));
        }
        else if(d=='C')
        {
            printf("Hasil konversi suhu: %g Celcius", suhu(t, s, d));
        }
        printf("\nLanjut? (y/n)");
        yes = getche();
        printf("\n");
    }
}

```

Output

```
Masukkan suhu (contoh: 100 C): 100 C
Masukkan tujuan suhu (C, R, F): F
Hasil konversi suhu: 212 Fahrenheit
Lanjut? (y/n) y
Masukkan suhu (contoh: 100 C): 212 F
Masukkan tujuan suhu (C, R, F): C
Hasil konversi suhu: 100 Celcius
Lanjut? (y/n) n
PS C:\Users\Hp\Music\Bab 6> |
```

Analisa

Pada program tersebut berfungsi untuk mengkonversi suhu dari Celcius ke Fahrenheit, Celcius ke Reamur, dan sebaliknya. Pada source code saya menggunakan double suhu (float a, char s, char d). Lalu program dijalankan dengan looping switch-case. Pada case terdapat logika perhitungan rumus konversi suhu. Lalu user memasukkan input beserta karakter seperti contoh. Lalu memasukkan tujuan konversi suhu dan program akan menampilkan hasilnya.

6.5.5 Buatlah fungsi prima(), yang memberikan nilai balik 1 bila bilangan yang dimasukkan adalah prima, dan 0 bila bukan bilangan prima.

Source Code

```
int prima(int n)
{
    int hasil, bil=0, i;
    for(i=1; i<=n; i++)
    {
        if(n%i==0)
        {
            bil+=1;
        }
    }
    if(bil==2)
    {
        hasil=1;
    }
    else
    {
        hasil=0;
    }
}
```

```

    }
    return hasil;
}
void main()
{
    int n, bil;
    system("cls");
    printf("Masukkan angka: ");
    scanf("%d", &n);
    printf("%d adalah ", n);
    bil = prima(n);
    if(bil==1)
    printf("bilangan prima");
    else
    printf("bukan bilangan prima");
    printf("\n");
}

```

Output

```

Masukkan angka: 31
31 adalah bilangan prima
PS C:\Users\Hp\Music\Bab 6>

```

Analisa

Pada program tersebut berfungsi untuk menentukan apakah angka yang dimasukkan termasuk prima atau tidak. Pada source code saya menggunakan int hasil, bil = 0. Lalu program diproses menggunakan looping for (i = 1; i <= n; i++). Lalu menggunakan logika matematika $n \% i == 0$ dimana bil += 1 dan juga bil += 2. Lalu user memasukkan input dan akan disimpan pada variabel n.

- 6.5.6** Buatlah program menggunakan dua variable, satu bertipe register dan lainnya bertipe integer, kemudian buatlah dua perulangan menggunakan statement for(), untuk membandingkan waktu yang diperlukan untuk masing-masing tipe variable. Gunakan fungsi gettimeofday() untuk menghitung waktu yang diperlukan oleh masing-masing perulangan.

Source Code

```

void main()
{
    system("cls");
    unsigned int Last=0;
    unsigned int Now=0;

```

```

Last = GetTickCount();
for(register i=1; i<=10000; i++)
{printf("x");}
Now = GetTickCount();

unsigned int Last1=0;
unsigned int Now1=0;
Last1 = GetTickCount();
for(int j=1; j<=10000; j++)
{printf("x");}
Now1 = GetTickCount();

printf("\n\nWaktu Eksekusi menggunakan register: %d
milliseconds.\n", (Now-Last));
printf("\n\nWaktu Eksekusi menggunakan integer: %d
milliseconds.\n\n", (Now1-Last1));
}

```

Output

```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Waktu Eksekusi menggunakan register: 6312 milliseconds.

Waktu Eksekusi menggunakan integer: 6625 milliseconds.

PS C:\Users\Hp\Music\Bab 6> 

```

Analisa

Pada program tersebut berfungsi untuk menentukan waktu yang diperlukan antara menggunakan dua variabel ber tipe register atau integer, manakah yang lebih cepat. Pada source code saya menggunakan `clock_t start = clock`, tidak lupa menggunakan `Library time.h`. Lalu jalannya program untuk tipe register menggunakan loop for (`register i=1; i<=20000; i++`). Lalu menggunakan format variabel `%g` agar angka yang dicetak lebih rapi. Lalu pada tipe integer menggunakan loop for tetapi (`int j; j<=20000; j++`). Lalu setelah user memasukkan nilai n maka `if(n==1)` akan mencetak waktunya.