

PRAKTIKUM DASAR PEMROGRAMAN

MODUL 9

FUNGSI



TUJUAN

1. Praktikan dapat mengerti dan memahami konsep dan struktur Fungsi.
2. Praktikan dapat mengerti perbedaan Fungsi dan Prosedur dalam penggunaannya pada program.
3. Praktikan dapat mengimplementasikan konsep dan struktur Fungsi dalam program maupun kasus nyata.

PENGERTIAN FUNGSI

Fungsi merupakan sebuah modul/sub-program di dalam pengkodean yang mengerjakan kasus atau pekerjaan tertentu. Fungsi tidaklah sama dengan prosedur, fungsi hanya memiliki satu tujuan dan menyelesaikan satu masalah yang hasilnya berupa sebuah nilai. Fungsi akan bekerja dengan cara mengambil data, kemudian memprosesnya, dan mengeluarkan output berupa nilai/hasil balikan sesuai dengan tipe datanya. Sebuah Fungsi tentunya dapat digunakan berulang ulang kali, maka dari itu salah satu tujuan Fungsi ialah membuat pengkodean lebih rapi dan efisien.

STRUKTUR FUNGSI

Fungsi memiliki struktur yang tidak jauh berbeda dari prosedur. Struktur Fungsi dapat dijelaskan sebagai berikut:

Deklarasi Fungsi :

```
tipe_data_balikan nama_fungsi(parameter1, parameter2, ...);
```

Contoh dalam Code Nyata :

```
int luasPersegiPanjang(int panjang, int lebar);
```

Pemakaian Fungsi :

```
tipe_data_balikan nama_fungsi(parameter1, parameter2, ...) {  
    body fungsi  
    return nama_variabel atau rumus;  
}
```

Contoh dalam Code Nyata:

```
// mengembalikan nilai variabel
int luasPersegiPanjang(int panjang, int lebar){

    int luas = panjang * lebar;

    return luas;
}

// mengembalikan nilai hasil perhitungan rumus
int luasPersegiPanjang(int panjang, int lebar){

    return panjang * lebar;
}
```

Keterangan :

1. **tipeData_balikan** – merupakan tipe data dari nilai balikan Fungsi, disini maksudnya nilai balikan adalah nilai / value yang akan dihasilkan dari Fungsi tersebut (int, char, float, bool).
2. **nama_Fungsi** – merupakan nama yang akan digunakan untuk mendeklarasikan Fungsi dan juga untuk pemakaiannya. Nama Fungsi adalah case sensitive dan memiliki format pada umumnya (luasSegitiga, volumeTabung, nilaiTerbesar).
3. **param** – merupakan parameter seperti pada prosedur dimana berfungsi sebagai placeholder atau variable-variable yang akan digunakan di dalam Fungsi tersebut. Parameter di dalam Fungsi merupakan parameter inputan dan tidak memiliki parameter output seperti Prosedur.
4. **Function body** – merupakan isi dari fungsi yang ingin dibuat, berisi kode dan syntax yang relevan dengan nama fungsinya dan mengerjakan suatu task atau perintah khusus tertentu.
5. **return** – merupakan syntax yang harus ada dan wajib ada dalam setiap pembuatan Fungsi, dimana digunakan untuk mengeluarkan nilai atau value dari hasil operasi Fungsi tersebut.

Contoh full program:

```
1
2 int luasPersegiPanjang(int panjang, int lebar);
3
4 int main() {
5
6     int Panjang, Lebar, Luas;
7
8     printf("Masukan panjang : ");
9     scanf("%d", &Panjang);
10    printf("Masukan lebar   : ");
11    scanf("%d", &Lebar);
12
13    Luas = luasPersegiPanjang(Panjang, Lebar);
14    printf("Luas persegi panjang : %d", Luas);
15
16    return 0;
17 }
18
19 int luasPersegiPanjang(int panjang, int lebar){
20
21     return panjang * lebar;
22 }
23
```

Prosedur vs Fungsi

Fungsi	Prosedur
Didefinisikan dengan tipe data yang nilainya akan dikembalikan. Contoh: <code>int luasSegitiga(int alas, int tinggi);</code>	Didefinisikan dengan void. Contoh: <code>void luasSegitigaPro(int *luas, int alas, int tinggi);</code>
Pemanggilan Fungsi harus menempel pada statement dan dapat dijadikan sebagai value. Contoh: //contoh 1 <code>luas = luasSegitiga(alas,tinggi);</code> //contoh 2 <code>printf("\n\tLuas Segitiga : %d",luasSegitiga(alas,tinggi));</code>	Pemanggilan harus berdiri sendiri dan diperlakukan sebagai statement. Contoh: <code>luasSegitigaPro(&luas, alas, tinggi);</code>
Pada Body fungsi, alur logika harus diakhiri dengan RETURN. RETURN disini digunakan untuk mengembalikan sebuah nilai. Contoh: <code>int luasSegitiga(int alas, int tinggi){ return 0.5 * alas * tinggi; }</code>	Dalam body prosedur, tidak membutuhkan sebuah nilai, cukup menggunakan parameter output. Contoh: <code>void luasSegitigaPro(int *luas, int alas, int tinggi){ (*luas) = 0.5 * alas * tinggi; }</code>
Hanya memiliki parameter input.	Dapat memiliki parameter input dan output.
Menghasilkan sebuah nilai.	Menghasilkan efek netto.

Guided Fungsi

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <string.h>
5
6  typedef char str[100];
7
8  bool validasiNoRekening(int noRekening);
9  /* Bekerja dengan mengambil 4 digit pertama dan menghitung
10     modulusnya dengan digit kedua dari akhir. Jika hasilnya sama
11     dengan digit terakhir, maka nomor tersebut dianggap valid.
12     - Nomor rekening harus 6 digit panjang */
13
14 float hitungPajakProgresif(float saldo);
15 /* fungsi ini digunakan untuk menghitung jumlah pajak yang harus dibayar
16     Pajak untuk saldo <=10000 --> 0%
17     Pajak untuk 10000 < saldo <= 50000 --> 10%
18     Pajak untuk saldo > 50000 --> 20% */
19
20 float cekSaldo (float saldo, str jenis);
21 /* fungsi ini digunakan untuk menghitung biaya admin cek saldo
22     rekening jenis gold tidak dikenakan biaya cek saldo
23     rekening jenis silver dikenakan biaya cek saldo Rp.2000
24     rekening jenis normal dikenakan biaya cek saldo 0.1% dari jumlah tabungan*/
25
26 bool login(int noRekening, str password);
27 /* gunakan fungsi validasiNoRekening untuk memastikan rekening valid
28     gunakan "atma" untuk password */
29
30 int main(int argc, char *argv[]) {
31     int rekening, menu;
32     str password, jenis;
33     float saldo = 25000, biayaAdmin;
34
35     while(1){
36         system("cls");
37         printf(" Masukkan nomor rekening: ");
38         scanf("%d", &rekening);
39
40         printf(" Masukkan password: ");
41         fflush(stdin); gets(password);
42
43         if (login(rekening, password)) {
44             break;
45         }
46         printf("\n\tLogin gagal! Nomor rekening atau password salah.\n");
47         getch();
48     }
49 }
```

```

51  do {
52      system("cls");
53      printf("\n\t Atma Bank\n");
54      printf("1. Set Saldo\n");
55      printf("2. Cek Saldo\n");
56      printf("3. Hitung Pajak\n");
57      printf("0. Keluar\n");
58      printf(">>> ");
59      scanf("%d", &menu);
60
61      switch (menu) {
62      case 1:
63          printf("Masukkan nilai saldo baru: ");
64          scanf("%f", &saldo);
65          printf("berhasil\n");
66          break;
67      case 2:
68          printf("Masukkan jenis rekening (gold/silver/normal): ");
69          scanf("%s", jenis);
70          biayaAdmin = cekSaldo(saldo, jenis);
71          saldo -= biayaAdmin;
72          printf("biaya admin   : %.2f\n", biayaAdmin);
73          printf("Saldo          : %.2f\n", saldo);
74          break;
75      case 3:
76          printf("Total pajak yang harus dibayar: %.2f\n", hitungPajakProgresif(saldo));
77          break;
78      case 0:
79          printf("Keluar dari program.\n");
80          break;
81      default:
82          printf("Pilihan tidak valid!\n");
83      }
84      getch();
85  } while (menu != 0);
86
87  return 0;
88  }
89
90  float hitungPajakProgresif(float saldo) {
91      float pajak = 0.0;
92
93      if (saldo > 50000) {
94          pajak += (saldo - 50000) * 0.20; // Pajak untuk saldo > 50000 --> 20%
95          saldo = 50000;
96      }
97      if (saldo > 10000) {
98          pajak += (saldo - 10000) * 0.10; // Pajak untuk 10000 < saldo <= 50000 --> 10%
99      }
100
101      return pajak;
102  }

```

```

104 bool validasiNoRekening(int noRekening){
105     if(noRekening > 999999 || noRekening < 100000){
106         return false;
107     }
108     //jika lebih dari atau kurang dari 6 digit maka return tidak valid
109
110     int validasi = noRekening;
111     validasi /= 100;        // mengambil empat digit pertama
112
113     int modulus = noRekening;
114     modulus %= 100;        // mengambil dua digit akhir
115     modulus /= 10;        // mengambil digit kedua dari kanan
116
117     if(modulus == 0){
118         return false;      // sama seperti pembagian, tidak boleh melakukan modulus 0
119     }
120
121     int nilaiEkspektasi = noRekening;
122     nilaiEkspektasi %= 10;    // mengambil digit terakhir
123
124     if(validasi % modulus == nilaiEkspektasi){
125         return true;
126     }else{
127         return false;
128     }
129 }
130
131 float cekSaldo(float saldo, str jenis) {
132     if (strcmp(jenis, "gold") == 0) {
133         return 0; // Tidak ada biaya untuk rekening gold
134     } else if (strcmp(jenis, "silver") == 0) {
135         return 2000; // Biaya admin Rp. 2000 untuk rekening silver
136     } else {
137         return (saldo * 0.001); // Biaya 0.1% untuk rekening normal
138     }
139 }
140
141 bool login(int noRekening, str password){
142     if(validasiNoRekening(noRekening) == true && strcmp(password, "atma") == 0){
143         return true;
144     }else{
145         return false;
146     }
147 }

```

Informasi tambahan:

Contoh nomor rekening valid = 123420

Aturan pengumpulan:

Format Pengumpulan : **GD9_X_YYYYY.zip**

X = Kelas; Y = 5 digit terakhir NPM

Dikumpul file lengkap!