

LAPORAN PRAKTIKUM PEMROGRAMAN
TERSTRUKTUR PRAKTIKUM 1 – PENDAHULUAN
KELAS B



Disusun Oleh :

Nama : Putra Dirgatama

NIM : 175090801111009

Hari/Tanggal Praktikum : Selasa, 9 April 2019

LABORATORIUM KOMPUTASI

JURUSAN FISIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS BRAWIJAYA

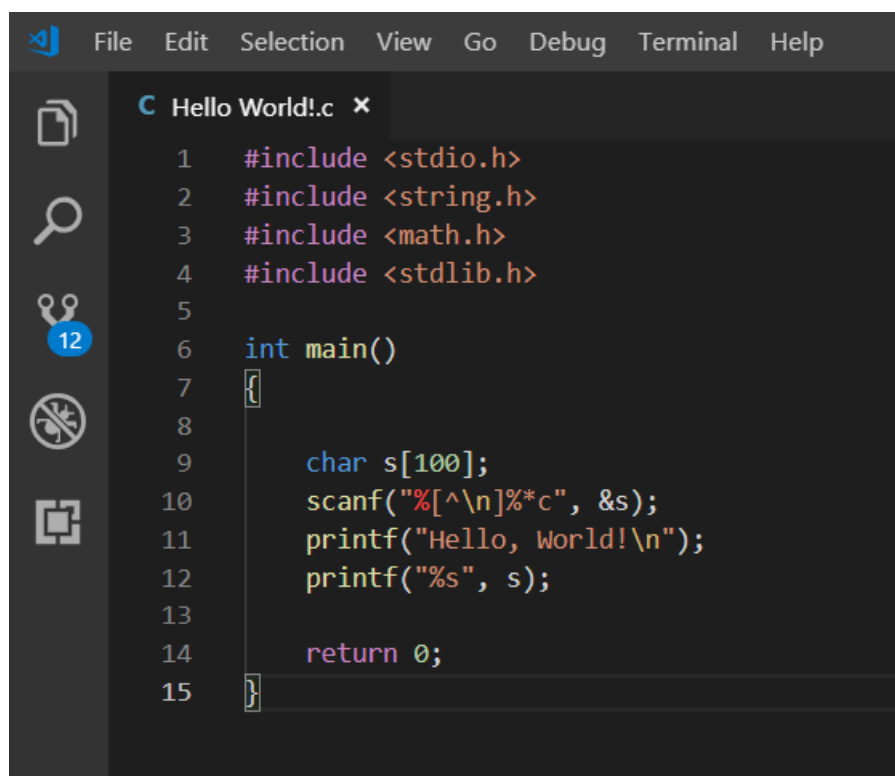
2019

A. Pendahuluan

Bahasa Pemrograman (programming language) adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Pada praktikum pemrograman terstruktur ini bahasa program yang kita pelajari adalah “Bahasa C”, ada banyak sekali software programing yang menggunakan Bahasa C salah satunya yang saya gunakan adalah “Visual Studio Code”.

B. Percobaan praktikum

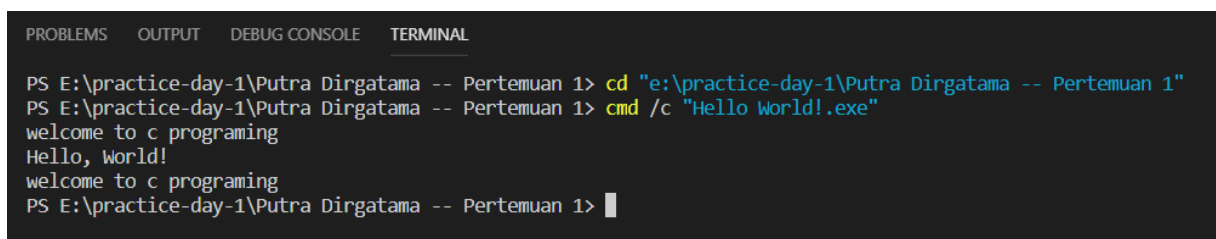
1. Kode Program 1 (Hello, World!)



```
File Edit Selection View Go Debug Terminal Help

C Hello World!.c x
1  #include <stdio.h>
2  #include <string.h>
3  #include <math.h>
4  #include <stdlib.h>
5
6  int main()
7  {
8
9      char s[100];
10     scanf("%[^\n]%", &s);
11     printf("Hello, World!\n");
12     printf("%s", s);
13
14     return 0;
15 }
```

Gambar 1. Kode Program 1 (Menampilkan output “Hello, World!”).

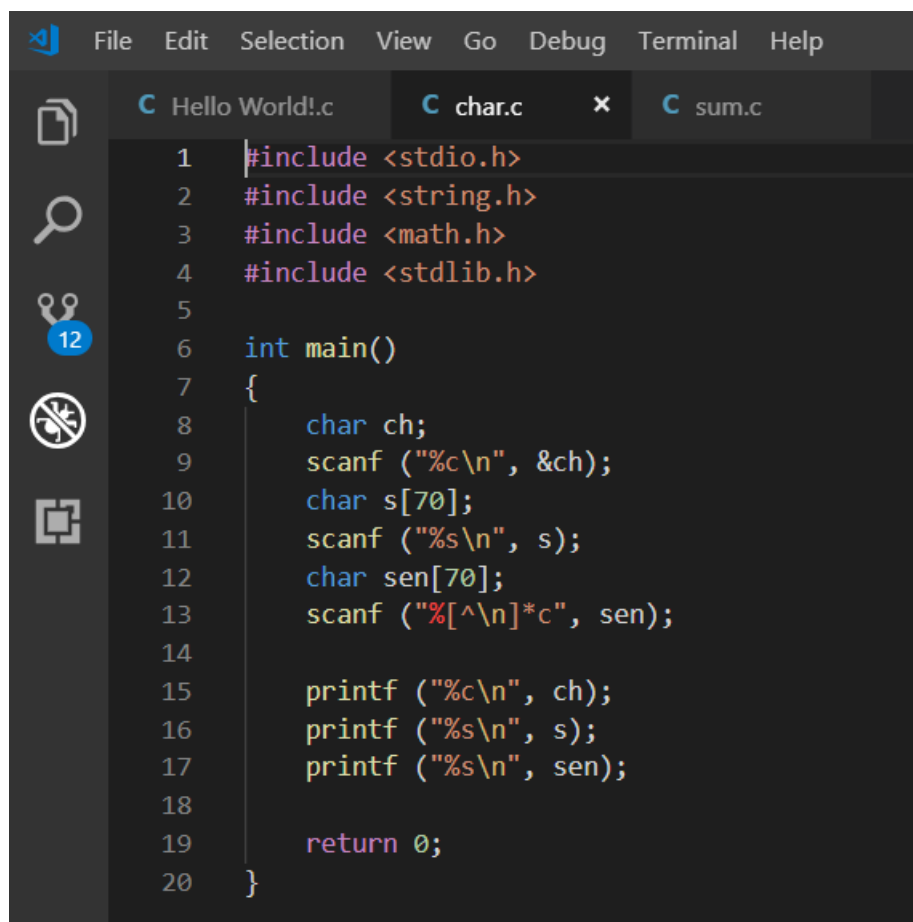


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> cd "e:\practice-day-1\Putra Dirgatama -- Pertemuan 1"
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> cmd /c "Hello World!.exe"
welcome to c programing
Hello, World!
welcome to c programing
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> |
```

Gambar 2. Hasil output dari Kode Program 1.

#include yaitu suatu syntax untuk memberitahu kepada kompilator bahwa program yang kita buat akan menggunakan file-file yang didaftarkan. *<stdio.h>* adalah sebuah file header yaitu file bantuan yang digunakan untuk menyimpan daftar-daftar fungsi yang akan digunakan didalam program bahasa C. *Int main()* adalah fungsi utama dari sebuah kode bahasa C. *char* disini berfungsi untuk membuat input dalam bentuk karakter yang jumlah/banyaknya input ditentukan dengan [...] ingin seberapa banyak kita masukkan jumlahnya bebas. *printf()* berfungsi untuk menampilkan keluaran data dan fungsi *scanf()* berguna untuk membaca masukkan data. Kedua fungsi ini terdapat pada header *<stdio.h>*. *return* ini akan mengakhiri fungsi main, *return 0* berarti nilai balik dari suatu fungsi adalah sebuah bilangan 0. Pada kode program 1 (Gambar 1) dilakukan perintah output dengan menggunakan display (“Hello, World!”), ketika dijalankan dengan MinGW (*compiler*) maka akan ditampilkan output “Hello, World!” (Gambar 2).

2. Kode Program 2 (Characters)



```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 int main()
7 {
8     char ch;
9     scanf ("%c\n", &ch);
10    char s[70];
11    scanf ("%s\n", s);
12    char sen[70];
13    scanf ("%[^\\n]*c", sen);
14
15    printf ("%c\n", ch);
16    printf ("%s\n", s);
17    printf ("%s\n", sen);
18
19    return 0;
20 }
```

Gambar 3. Kode Program 2 (Menampilkan output berupa character sesuai inputan-nya).

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> cd "e:\practice-day-1\Putra Dirgatama -- Pertemuan 1"
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> cmd /c "char.exe"
c
language
welcome to c programing !!
c
language
welcome to c programing !!
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> █
```

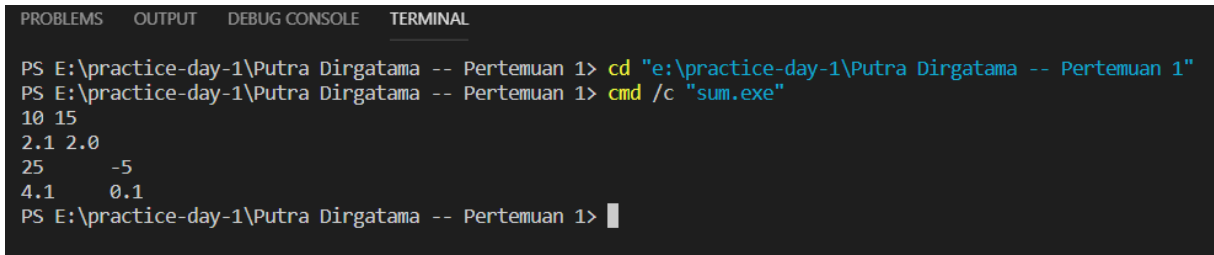
Gambar 4. Hasil output dari Kode Program 2.

Scanf ("%c\n", &ch); untuk membaca sebuah karakter lalu dalam output dia ditampilkan berada di new line, ch tersebut hanyalah sebuah variabel dari character. *Scanf* ("%s\n", s); ini berfungsi untuk membaca sebuah data string lalu di new line kan. %[^\n] digunakan menggantikan %s agar semua karakter termasuk spasi akan dibaca sampai ditemui penekan tombol enter, karena %s tidak akan membaca spasi atau tab. Untuk input string tidak perlu menggunakan operator pointer '&', karena pengenalan ini sudah berbentuk suatu pointer. *Printf* ("%c") untuk menampilkan output dalam sebuah karakter sedangkan *printf* ("%s") untuk menampilkan output string tersebut.

3. Kode Program 3 (Sum & Difference)

```
File Edit Selection View Go Debug Terminal Help
C Hello World!.c C char.c C sum.c x
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 int main()
7 {
8     int z;
9     int x;
10    int c,v;
11    float q;
12    float w;
13    float e,r;
14
15    scanf ("%d %d",&z ,&x);
16    scanf ("%f %f",&q ,&w);
17    c = z+x;
18    printf ("%d\t", c);
19    v = z-x;
20    printf ("%d\n", v);
21
22    e = q+w;
23    printf ("%1f\t", e);
24    r = q-w;
25    printf ("%1f\n", r);
26
27    return 0;
28 }
```

Gambar 5. Kode Program 3 (sum & difference).



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> cd "e:\practice-day-1\Putra Dirgatama -- Pertemuan 1"
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> cmd /c "sum.exe"
10 15
2.1 2.0
25      -5
4.1      0.1
PS E:\practice-day-1\Putra Dirgatama -- Pertemuan 1> |
```

Gambar 6. Hasil output dari Kode Program 3.

Int tipe data untuk menampung bilangan bulat, *float* tipe data untuk menampung bilangan pecahan. *Int* *z* dan *int* *x* untuk memasukan angka yang diinginkan dalam bilangan bulat, *int* *c,v* suatu variabel fungsi matematika untuk menghitung penjumlahan dan pengurangan dalam bilangan bulat. *Float* *q* dan *float* *w* untuk memasukan angka yang diinginkan dalam bilangan pecahan, *float* *e,r* suatu variabel fungsi matematika untuk menghitung penjumlahan dan pengurangan dalam bilangan pecahan. *Scanf* ("%d %d", &*z*, &*x*); ini berfungsi untuk membaca data yang dimasukkan benar-benar sebuah bilangan bulat dari *int* *z* dan *int* *x* tadi. *Scanf* ("%f %f", &*q*, &*w*); ini berfungsi untuk membaca data yang dimasukkan benar-benar sebuah bilangan pecahan dari *float* *q* dan *float* *w* tadi. Lalu data yang sudah dimasukkan di print dengan output seperti pada (Gambar 6), *\t* (*horizontal tab*) disini berfungsi agar hasil pengurangan ditampilkan sedikit berjarak dengan penjumlahannya. Lalu dalam print bilangan float terdapat *%lf* agar hasil dari outputnya terlihat 1 angka dibelakang koma.