1. **Tujuan Praktikum**

Mahasiswa mampu memahami dan menerapkan input ouput berformat dan tidak berformat dalam penyelesaian kasus bahasa pemrograman C.

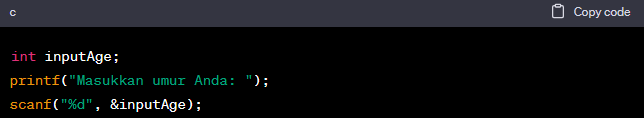
1. **Teori Singkat**

**Tipe Data**

| **Tipe Data** | | **Ukuran (bytes)** | **Format Specifier** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| int | | Setidaknya 2, biasanya 4 | %d, %i | Menyimpan bilangan bulat. |
| char | | 1 | %c | Menyimpan karakter tunggal. |
| float | | 4 | %f | Menyimpan bilangan desimal dengan presisi standar. |
| double | | 8 | %lf | Menyimpan bilangan desimal dengan presisi lebih tinggi daripada float. |
| short int | | Biasanya 2 | %hd | Menyimpan bilangan bulat dengan rentang lebih kecil daripada int. |
| unsigned int | | Setidaknya 2, biasanya 4 | %u | Menyimpan bilangan bulat non-negatif. |
| long int | | Setidaknya 4, biasanya 8 | %ld, %li | Menyimpan bilangan bulat dengan rentang lebih besar daripada int. |
| long long int | | Setidaknya 8 | %lld, %lli | Menyimpan bilangan bulat dengan rentang lebih besar daripada long int. |
| unsigned long int | | Setidaknya 4 | %lu | Menyimpan bilangan bulat non-negatif dengan rentang lebih besar. |
| unsigned long long int | | Setidaknya 8 | %llu | Menyimpan bilangan bulat non-negatif dengan rentang lebih besar. |
| signed char | | 1 | %c | Menyimpan karakter dengan tanda. |
| unsigned char | | 1 | %c | Menyimpan karakter tanpa tanda. |
| long double | | Setidaknya 10, biasanya 12 atau 16 | %Lf | Menyimpan bilangan desimal dengan presisi yang lebih tinggi daripada double. |
| **Konsep/Tipe Data** | **Deskripsi** | | | |
| Tipe Data Turunan | Tipe data yang berasal dari tipe data dasar. Misalnya: array, pointer, tipe data fungsi, struktur, dll. Tutorial selanjutnya akan membahasnya. | | | |
| Bool | Tipe data boolean untuk menyimpan nilai kebenaran (true atau false). | | | |
| Enumerasi (Enum) | Tipe data yang memungkinkan Anda mendefinisikan konstanta dengan label untuk meningkatkan keterbacaan kode. | | | |
| Tipe Data Kompleks | Tipe data seperti complex (meskipun tidak standar dalam C) untuk bekerja dengan bilangan kompleks dalam matematika. | | | |

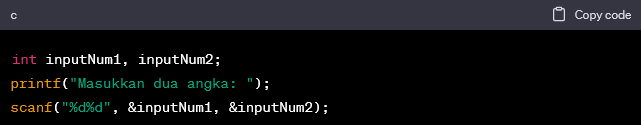
**Input Output Berformat**

Input-Output berformat dalam bahasa C mengacu pada teknik di mana Anda mengontrol tampilan dan pemrosesan data yang masuk dan keluar dari program dengan menggunakan format tertentu. Ini melibatkan penggunaan format specifier dalam perintah printf untuk menampilkan data dengan format tertentu, dan dalam perintah **scanf** untuk membaca data dengan format yang sesuai.



**Input-Output Tidak Berformat dalam Bahasa C:**

Input-Output tidak berformat dalam bahasa C melibatkan penggunaan perintah printf dan scanf tanpa format specifier atau manipulasi format data yang lebih lanjut. Dalam hal ini, data ditampilkan atau dibaca sesuai dengan urutan mereka dalam kode.



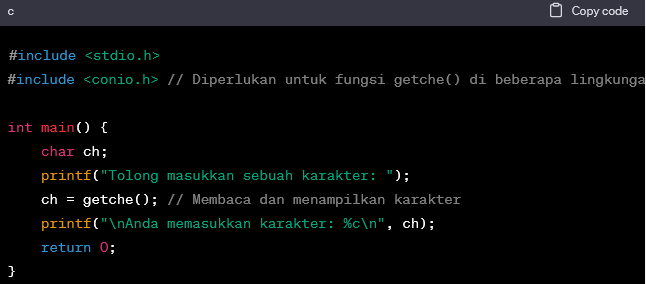
**gets() dan puts():**

* gets(): Fungsi gets() digunakan untuk membaca satu baris teks dari input standar (biasanya keyboard) dan menyimpannya sebagai string. Ini adalah perintah input.
* puts(): Fungsi puts() digunakan untuk menampilkan string ke layar (output standar) dengan menambahkan karakter newline ("\n") pada akhirnya. Ini adalah perintah output.
* Format: Perintah puts() termasuk dalam kategori berformat karena Anda dapat mengontrol format tampilan teks yang akan ditampilkan dengan cara menambahkan karakter newline.



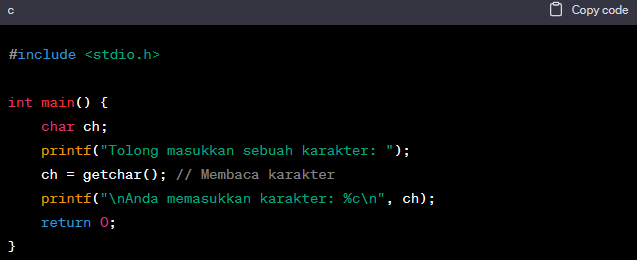
**getche():**

* + **getche():** Fungsi getche() digunakan untuk membaca satu karakter dari input standar (biasanya keyboard) dan menampilkannya ke layar tanpa perlu menekan tombol "Enter". Ini adalah perintah input.
  + **Format**: Perintah getche() termasuk dalam kategori tidak berformat karena tidak ada kontrol format terhadap karakter yang dibaca.



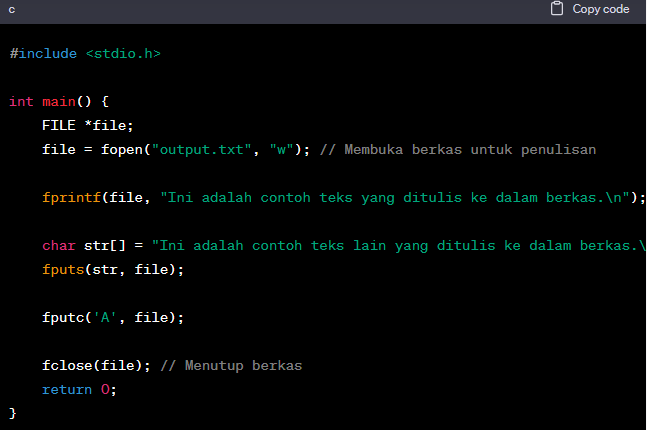
**getchar():**

* + **getchar():** Fungsi getchar() digunakan untuk membaca satu karakter dari input standar (biasanya keyboard) dan mengembalikan nilai integer yang mewakili karakter tersebut. Ini adalah perintah input.
  + **Format**: Perintah getchar() termasuk dalam kategori tidak berformat karena tidak ada kontrol format terhadap karakter yang dibaca.



**fprintf(), fputs(), dan fputc():**

* + **fprintf():** Fungsi fprintf() digunakan untuk menulis data ke dalam berkas dengan format tertentu. Ini adalah perintah output.
  + **fputs():** Fungsi fputs() digunakan untuk menulis string ke dalam berkas. Ini adalah perintah output.
  + **fputc():** Fungsi fputc() digunakan untuk menulis karakter ke dalam berkas. Ini adalah perintah output.
  + **Format**: Perintah-perintah ini termasuk dalam kategori berformat karena Anda dapat mengontrol format data yang akan ditulis ke dalam berkas, terutama dalam hal penentuan tata letak dan pemformatan teks.



Jadi, secara umum, gets(), getche(), getchar() adalah perintah-perintah input tanpa format, sementara puts(), fprintf(), fputs(), dan fputc() adalah perintah-perintah output yang dapat berformat.

1. **Alat dan Bahan**
2. Komputer
3. DevC++
4. Jobsheet
5. **Kasus dalam Pemrograman**
6. Mengetikkan Nama dan Usia

|  |
| --- |
| /\* Nama File : latihan3A.c  Programmer : tuliskan nama dan nim anda  Tgl. pembuatan : tuliskan tanggal hari praktikum anda  Deskripsi : program ini represtasi input output berformat  \*/  #include <stdio.h>  int main() {  char name[50];  int age;  // Input dengan format  printf("Masukkan nama Anda: ");  scanf("%s", name);  printf("Masukkan usia Anda: ");  scanf("%d", &age);  // Menampilkan informasi dengan format  printf("Halo, %s! Anda berusia %d tahun.\n", name, age);  return 0;  } |

1. Menampilkan teks

|  |
| --- |
| /\* Nama File : latihan3B.c  Programmer : tuliskan nama dan nim anda  Tgl. pembuatan : tuliskan tanggal hari praktikum anda  Deskripsi : program ini menampilkan teks  \*/  #include <stdio.h>  int main() {  char text[100];  printf("Masukkan beberapa baris teks :\n");  // Membaca beberapa baris teks dengan gets()  gets(text);  printf("\nAnda memasukkan teks berikut:\n");  // Menampilkan teks dengan puts()  puts(text);  return 0;  } |

1. Membaca karakter getchar()

|  |
| --- |
| /\* Nama File : latihan3C.c  Programmer : tuliskan nama dan nim anda  Tgl. pembuatan : tuliskan tanggal hari praktikum anda  Deskripsi : program ini membaca karakter getchar()  \*/  #include <stdio.h>  int main() {  char ch;  printf("Tolong masukkan sebuah karakter:\n");  ch = getche(); // membaca dan menampilkan karakter  printf("\nAnda memasukkan karakter berikut: %c \n", ch);  return 0;  } |

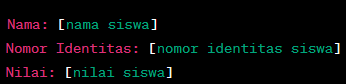
1. Membaca karakter getchar()

|  |
| --- |
| /\* Nama File : latihan3D.c  Programmer : tuliskan nama dan nim anda  Tgl. pembuatan : tuliskan tanggal hari praktikum anda  Deskripsi : program ini melakukan perhitungan diskon  \*/  #include <stdio.h>  int main() {  double harga, diskon, total;  // Meminta pengguna memasukkan harga dan persentase diskon  printf("Harga barang: ");  scanf("%lf", &harga);  printf("Persentase diskon (dalam persen): ");  scanf("%lf", &diskon);  // Menghitung total harga setelah diskon  total = harga \* (1 - diskon / 100.0);  // Menyimpan detail pembelian ke dalam berkas "nota.txt" menggunakan fprintf  FILE \*file = fopen("nota.txt", "w");  if (file != NULL) {  fprintf(file, "Harga Barang : Rp. %.2lf\n", harga);  fprintf(file, "Persentase Diskon : Rp. %.2lf%%\n", diskon);  fprintf(file, "Total Harga Setelah Diskon : Rp. %.2lf\n", total);  fclose(file);  printf("Detail pembelian telah disimpan ke dalam berkas 'nota.txt'.\n");  } else {  printf("Gagal membuka berkas untuk penyimpanan detail pembelian.\n");  return 1; // Keluar dengan kode kesalahan  }  return 0; // Keluar program  } |

1. Membaca karakter getchar()

|  |
| --- |
| /\* Nama File : latihan3E.c  Programmer : tuliskan nama dan nim anda  Tgl. pembuatan : tuliskan tanggal hari praktikum anda  Deskripsi : program ini melakukan perhitungan cicilan kredit barang  \*/  #include <stdio.h>  #include <stdlib.h>  int main() {  char input[100]; // Buffer untuk menyimpan input  char output[100]; // Buffer untuk menyimpan output  double harga, hutang, bunga, cicilan, uangMuka, sisa, pokok;  int bulan, kaliKe;  // Input tanpa format menggunakan fgets()  printf("Masukkan harga barang : ");  fgets(input, sizeof(input), stdin);  harga = atof(input); // Mengubah teks menjadi double  printf("Masukkan besar uang muka : ");  fgets(input, sizeof(input), stdin);  uangMuka = atof(input); // Mengubah teks menjadi double  printf("Masukkan jumlah bulan kredit : ");  fgets(input, sizeof(input), stdin);  bulan = atoi(input); // Mengubah teks menjadi integer  printf("Masukkan Cicilan kali ke berapa : ");  fgets(input, sizeof(input), stdin);  kaliKe = atoi(input); // Mengubah teks menjadi integer  // Menghitung pokok hutang  hutang = harga - uangMuka;  // Menghitung bunga perbulan  bunga = (0.09 \* hutang) / 12;  // Menghitung pokok bulanan  pokok = hutang / bulan;  // Menghitung besar cicilan perbulan  cicilan = pokok + bunga;  // Menghitung sisa hutang  sisa = hutang - pokok \* kaliKe;  // Menampilkan informasi tanpa pemformatan menggunakan puts()  puts("\nHarga Barang : Rp.");  puts(input);  puts("Uang Muka: Rp.");  puts(input);  printf("Banyaknya bulan cicilan : %d kali cicilan\n", bulan);  puts("Pokok Hutang: Rp.");  puts(input);  printf("Besar bunga bulanan : Rp. %.2f\n", bunga);  puts("Besar Cicilan per bulan : Rp.");  puts(input);  printf("Cicilan ke-: %d\n", kaliKe);  puts("Sisa Pokok Hutang : Rp.");  puts(input);  return 0;  } |

1. **Evaluasi**
2. Buatlah program dalam bahasa C yang memungkinkan pengguna untuk memasukkan data siswa, termasuk nama, nomor identitas, dan nilai. Program harus memungkinkan pengguna untuk memasukkan data untuk beberapa siswa dan menyimpannya dalam berkas "data\_siswa.txt" dengan menggunakan fprintf(). Setiap baris dalam berkas harus berisi data satu siswa dengan format:



1. Buatlah program dalam bahasa C yang meminta pengguna memasukkan sebuah kalimat. Program harus menghitung dan menampilkan jumlah karakter dalam kalimat tersebut tanpa format tambahan. Misalnya, jika pengguna memasukkan kalimat "Hari ini adalah hari yang cerah", program harus menampilkan:



1. **Kesimpulan**

Setelah menyelsaikan jobsheet ini mahasiswa mampu melakukan penanganan input dan output berformat dan tidak berformat dalam penyelesaian kasus program bahasa C.

1. **Referensi**

|  |
| --- |
| Utama |
| * Rinaldi Munir. 2016. *Algoritma dan Pemrograman*. Bandung. Informatika ITB * Noel Kalicharan. 2015. *Learn to Program with C*. New York, Springer-Science * Harry H. Chaudhary. 2014. *C Programming Step by Step*. LLC USA. Amazong Inc. |
| Pendukung |
| * Mike McGrath. 2015. *Coding for Beginners*. Leamington Spa. Easy Step Limited. * Dan Gookin. 2014. *Beginning Programming with C for Dummies*. New Jersey. John Wiley & Sons. * [www.tutorialspoint.com](http://www.tutorialspoint.com) * [www.javatpoint.com](http://www.javatpoint.com) * www.programiz.com |