**Kasus2 Input Output Berformat**

Berikut adalah kasus-kasus soal terkait dengan **Penanganan Input Output Berformat (Format Specifier)** tanpa menggunakan alur decisional maupun looping, beserta penyelesaiannya:

**Soal 1: Menghitung Luas dan Keliling Persegi Panjang**

**Kasus:**

Sebuah aplikasi arsitektur ingin menghitung luas dan keliling persegi panjang berdasarkan panjang dan lebar yang dimasukkan pengguna. Nilai yang dimasukkan harus ditampilkan dengan format dua angka di belakang koma.

**Penjelasan Kasus:**

* Input berupa panjang dan lebar persegi panjang (float).
* Output berupa luas dan keliling (dengan format dua desimal).

**Analisis Kebutuhan:**

* Diperlukan variabel panjang, lebar, luas, dan keliling.
* Operasi matematika yang dibutuhkan adalah perkalian untuk luas dan penjumlahan untuk keliling.

**Langkah-langkah Penyelesaian:**

1. Baca panjang dan lebar persegi panjang.
2. Hitung luas dan keliling.
3. Tampilkan hasil dengan dua angka di belakang koma.

**Pseudocode:**

*Algoritma HitungLuasKelilingPersegiPanjang (input, output)*

*{Menghitung luas dan keliling persegi panjang dengan format dua desimal}*

*Deklarasi*

*Variabel:*

*panjang, lebar, luas, keliling: real*

*Deskripsi*

*start*

*// Membaca input panjang dan lebar persegi panjang*

*read(panjang, lebar)*

*// Menghitung luas*

*luas = panjang \* lebar*

*// Menghitung keliling*

*keliling = 2 \* (panjang + lebar)*

*// Menampilkan hasil dengan format dua desimal*

*write("Luas: ", luas, " Keliling: ", keliling)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*float panjang, lebar, luas, keliling;*

*// Membaca panjang dan lebar persegi panjang*

*printf("Masukkan panjang: ");*

*scanf("%f", &panjang);*

*printf("Masukkan lebar: ");*

*scanf("%f", &lebar);*

*// Menghitung luas dan keliling*

*luas = panjang \* lebar;*

*keliling = 2 \* (panjang + lebar);*

*// Menampilkan hasil dengan format dua desimal*

*printf("Luas: %.2f\n", luas);*

*printf("Keliling: %.2f\n", keliling);*

*return 0;*

*}*

**Penjelasan Program:**

Program membaca input panjang dan lebar, menghitung luas dan keliling, lalu menampilkan hasil dengan format dua angka di belakang koma.

**Soal 2: Konversi Suhu dari Celsius ke Fahrenheit dan Kelvin**

**Kasus:**

Sebuah aplikasi cuaca ingin mengonversi suhu dari Celsius ke Fahrenheit dan Kelvin. Hasil konversi harus ditampilkan dengan format satu angka di belakang koma.

**Penjelasan Kasus:**

* Input berupa suhu dalam Celsius (float).
* Output berupa suhu dalam Fahrenheit dan Kelvin.

**Analisis Kebutuhan:**

* Diperlukan variabel celsius, fahrenheit, dan kelvin.
* Operasi yang dilakukan adalah konversi suhu.

**Langkah-langkah Penyelesaian:**

1. Baca suhu dalam Celsius.
2. Hitung suhu dalam Fahrenheit dan Kelvin.
3. Tampilkan hasil dengan format satu angka di belakang koma.

**Pseudocode:**

*Algoritma KonversiSuhu (input, output)*

*{Mengonversi suhu dari Celsius ke Fahrenheit dan Kelvin dengan format satu desimal}*

*Deklarasi*

*Variabel:*

*celsius, fahrenheit, kelvin: real*

*Deskripsi*

*start*

*// Membaca suhu dalam Celsius*

*read(celsius)*

*// Mengonversi ke Fahrenheit dan Kelvin*

*fahrenheit = (celsius \* 9 / 5) + 32*

*kelvin = celsius + 273.15*

*// Menampilkan hasil dengan format satu desimal*

*write("Fahrenheit: ", fahrenheit, " Kelvin: ", kelvin)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*float celsius, fahrenheit, kelvin;*

*// Membaca suhu dalam Celsius*

*printf("Masukkan suhu dalam Celsius: ");*

*scanf("%f", &celsius);*

*// Mengonversi ke Fahrenheit dan Kelvin*

*fahrenheit = (celsius \* 9 / 5) + 32;*

*kelvin = celsius + 273.15;*

*// Menampilkan hasil dengan format satu desimal*

*printf("Fahrenheit: %.1f\n", fahrenheit);*

*printf("Kelvin: %.1f\n", kelvin);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca suhu dalam Celsius, lalu menghitung suhu yang setara dalam Fahrenheit dan Kelvin, dan menampilkan hasilnya dengan satu angka di belakang koma.

**Soal 3: Menghitung Umur dari Tahun Lahir**

**Kasus:**

Program ini akan menghitung umur seseorang berdasarkan tahun lahir yang dimasukkan. Tahun lahir akan ditampilkan dalam format integer tanpa desimal.

**Penjelasan Kasus:**

* Input berupa tahun lahir (integer).
* Output berupa umur dalam tahun saat ini.

**Analisis Kebutuhan:**

* Diperlukan variabel tahunLahir dan umur.
* Operasi yang dilakukan adalah pengurangan tahun saat ini dengan tahun lahir.

**Langkah-langkah Penyelesaian:**

1. Baca tahun lahir.
2. Hitung umur berdasarkan tahun saat ini.
3. Tampilkan umur.

**Pseudocode:**

*Algoritma HitungUmur (input, output)*

*{Menghitung umur berdasarkan tahun lahir}*

*Deklarasi*

*Konstanta:*

*TAHUN\_SEKARANG: integer = 2024*

*Variabel:*

*tahunLahir, umur: integer*

*Deskripsi*

*start*

*// Membaca tahun lahir*

*read(tahunLahir)*

*// Menghitung umur*

*umur = TAHUN\_SEKARANG - tahunLahir*

*// Menampilkan umur*

*write("Umur: ", umur)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*const int TAHUN\_SEKARANG = 2024;*

*int tahunLahir, umur;*

*// Membaca tahun lahir*

*printf("Masukkan tahun lahir: ");*

*scanf("%d", &tahunLahir);*

*// Menghitung umur*

*umur = TAHUN\_SEKARANG - tahunLahir;*

*// Menampilkan umur*

*printf("Umur: %d tahun\n", umur);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca tahun lahir, menghitung umur berdasarkan tahun saat ini (2024), lalu menampilkan hasil dalam format bilangan bulat.

**Soal 4: Menghitung Harga Akhir dengan Diskon**

**Kasus:**

Program ini bertujuan menghitung harga akhir setelah penerapan diskon. Harga dan diskon dimasukkan oleh pengguna, hasil akhir ditampilkan dengan format dua desimal.

**Penjelasan Kasus:**

* Input berupa harga awal dan persentase diskon.
* Output berupa harga akhir setelah diskon.

**Analisis Kebutuhan:**

* Diperlukan variabel hargaAwal, diskon, dan hargaAkhir.
* Operasi yang dilakukan adalah pengurangan harga berdasarkan diskon.

**Langkah-langkah Penyelesaian:**

1. Baca harga awal dan diskon.
2. Hitung harga akhir setelah diskon.
3. Tampilkan harga akhir.

**Pseudocode:**

*Algoritma HitungHargaAkhir (input, output)*

*{Menghitung harga akhir setelah diskon dengan format dua desimal}*

*Deklarasi*

*Variabel:*

*hargaAwal, diskon, hargaAkhir: real*

*Deskripsi*

*start*

*// Membaca harga awal dan diskon*

*read(hargaAwal, diskon)*

*// Menghitung harga akhir*

*hargaAkhir = hargaAwal - (hargaAwal \* diskon / 100)*

*// Menampilkan harga akhir dengan format dua desimal*

*write("Harga Akhir: ", hargaAkhir)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*float hargaAwal, diskon, hargaAkhir;*

*// Membaca harga awal dan diskon*

*printf("Masukkan harga awal: ");*

*scanf("%f", &hargaAwal);*

*printf("Masukkan diskon (%%): ");*

*scanf("%f", &diskon);*

*// Menghitung harga akhir*

*hargaAkhir = hargaAwal - (hargaAwal \* diskon / 100);*

*// Menampilkan harga akhir dengan format dua desimal*

*printf("Harga Akhir: %.2f\n", hargaAkhir);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini menghitung harga akhir setelah penerapan diskon, menampilkan hasilnya dalam format dua desimal.

**Soal 5: Mengonversi Nilai Uang dalam Rupiah ke Dolar**

**Kasus:**

Sebuah aplikasi keuangan ingin mengonversi nilai uang dari Rupiah (IDR) ke Dolar AS (USD). Nilai tukar harus ditampilkan dengan empat angka di belakang koma.

**Penjelasan Kasus:**

* Input berupa nilai Rupiah.
* Output berupa nilai dalam Dolar AS.
* Nilai tukar tetap adalah 15.000 IDR = 1 USD.

**Analisis Kebutuhan:**

* Diperlukan variabel rupiah, dolar, dan rate (konstanta).
* Operasi yang dilakukan adalah pembagian.

**Langkah-langkah Penyelesaian:**

1. Baca nilai Rupiah.
2. Hitung nilai dalam Dolar AS.
3. Tampilkan nilai dengan format empat angka di belakang koma.

**Pseudocode:**

*Algoritma KonversiRupiahKeDolar (input, output)*

*{Mengonversi nilai Rupiah ke Dolar dengan format empat desimal}*

*Deklarasi*

*Konstanta:*

*RATE: real = 15000*

*Variabel:*

*rupiah, dolar: real*

*Deskripsi*

*start*

*// Membaca nilai Rupiah*

*read(rupiah)*

*// Mengonversi ke Dolar*

*dolar = rupiah / RATE*

*// Menampilkan hasil dengan format empat desimal*

*write("Nilai dalam Dolar: ", dolar)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*const float RATE = 15000;*

*float rupiah, dolar;*

*// Membaca nilai Rupiah*

*printf("Masukkan nilai Rupiah: ");*

*scanf("%f", &rupiah);*

*// Mengonversi ke Dolar*

*dolar = rupiah / RATE;*

*// Menampilkan hasil dengan format empat desimal*

*printf("Nilai dalam Dolar: %.4f\n", dolar);*

*return 0;*

*}*

**Penjelasan Program:**

Program membaca nilai Rupiah, mengonversinya ke Dolar AS menggunakan kurs tetap, dan menampilkan hasilnya dengan format empat angka di belakang koma.

**Soal 6: Menghitung Rata-rata Tiga Nilai Ujian**

**Kasus:**

Program ini menghitung rata-rata dari tiga nilai ujian. Hasil rata-rata ditampilkan dengan format tiga desimal.

**Penjelasan Kasus:**

* Input berupa tiga nilai ujian.
* Output berupa rata-rata nilai ujian.

**Analisis Kebutuhan:**

* Diperlukan variabel nilai1, nilai2, nilai3, dan rataRata.
* Operasi yang dilakukan adalah penjumlahan dan pembagian.

**Langkah-langkah Penyelesaian:**

1. Baca nilai-nilai ujian.
2. Hitung rata-rata.
3. Tampilkan rata-rata dengan format tiga angka di belakang koma.

**Pseudocode:**

*Algoritma HitungRataRataNilaiUjian (input, output)*

*{Menghitung rata-rata nilai ujian dengan format tiga desimal}*

*Deklarasi*

*Variabel:*

*nilai1, nilai2, nilai3, rataRata: real*

*Deskripsi*

*start*

*// Membaca nilai ujian*

*read(nilai1, nilai2, nilai3)*

*// Menghitung rata-rata*

*rataRata = (nilai1 + nilai2 + nilai3) / 3*

*// Menampilkan rata-rata dengan format tiga desimal*

*write("Rata-rata: ", rataRata)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*float nilai1, nilai2, nilai3, rataRata;*

*// Membaca nilai ujian*

*printf("Masukkan nilai ujian pertama: ");*

*scanf("%f", &nilai1);*

*printf("Masukkan nilai ujian kedua: ");*

*scanf("%f", &nilai2);*

*printf("Masukkan nilai ujian ketiga: ");*

*scanf("%f", &nilai3);*

*// Menghitung rata-rata*

*rataRata = (nilai1 + nilai2 + nilai3) / 3;*

*// Menampilkan hasil dengan format tiga desimal*

*printf("Rata-rata: %.3f\n", rataRata);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca tiga nilai ujian dari pengguna, menghitung rata-rata, dan menampilkan hasilnya dengan format tiga angka di belakang koma.

Berikut tambahan 4 buah soal terkait **Penanganan Input Output Berformat (Format Specifier)** tanpa penggunaan alur decisional maupun looping:

**Soal 7: Menghitung Jarak dalam Kilometer dan Mil**

**Kasus:**

Sebuah program akan menghitung jarak dalam kilometer dan mengonversinya ke mil. Hasilnya harus ditampilkan dengan format tiga angka di belakang koma.

**Penjelasan Kasus:**

* Input berupa jarak dalam kilometer (float).
* Output berupa jarak dalam kilometer dan mil.
* Rasio konversi yang digunakan: 1 kilometer = 0.621371 mil.

**Analisis Kebutuhan:**

* Diperlukan variabel kilometer, mil.
* Operasi yang dilakukan adalah perkalian untuk konversi kilometer ke mil.

**Langkah-langkah Penyelesaian:**

1. Baca jarak dalam kilometer.
2. Konversikan ke mil.
3. Tampilkan jarak dalam kilometer dan mil dengan format tiga angka di belakang koma.

**Pseudocode:**

*Algoritma HitungJarak (input, output)*

*{Menghitung jarak dalam kilometer dan konversi ke mil dengan format tiga desimal}*

*Deklarasi*

*Konstanta:*

*KONVERSI: real = 0.621371*

*Variabel:*

*kilometer, mil: real*

*Deskripsi*

*start*

*// Membaca jarak dalam kilometer*

*read(kilometer)*

*// Mengonversi ke mil*

*mil = kilometer \* KONVERSI*

*// Menampilkan hasil dengan format tiga desimal*

*write("Kilometer: ", kilometer, " Mil: ", mil)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*const float KONVERSI = 0.621371;*

*float kilometer, mil;*

*// Membaca jarak dalam kilometer*

*printf("Masukkan jarak dalam kilometer: ");*

*scanf("%f", &kilometer);*

*// Mengonversi ke mil*

*mil = kilometer \* KONVERSI;*

*// Menampilkan hasil dengan format tiga desimal*

*printf("Kilometer: %.3f km\n", kilometer);*

*printf("Mil: %.3f mi\n", mil);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca jarak dalam kilometer, mengonversi jarak tersebut ke mil, dan menampilkan hasilnya dengan format tiga angka di belakang koma.

**Soal 8: Menghitung Total Pembelian dengan Pajak**

**Kasus:**

Sebuah aplikasi kasir ingin menghitung total harga pembelian barang setelah ditambahkan pajak sebesar 10%. Hasil akhir harus ditampilkan dengan format dua angka di belakang koma.

**Penjelasan Kasus:**

* Input berupa harga barang (float).
* Output berupa total harga setelah ditambahkan pajak.
* Pajak yang digunakan adalah 10% dari harga barang.

**Analisis Kebutuhan:**

* Diperlukan variabel hargaBarang, totalHarga.
* Operasi yang dilakukan adalah penambahan 10% dari harga barang.

**Langkah-langkah Penyelesaian:**

1. Baca harga barang.
2. Hitung total harga setelah ditambahkan pajak.
3. Tampilkan hasil dengan format dua angka di belakang koma.

**Pseudocode:**

*Algoritma HitungTotalPembelian (input, output)*

*{Menghitung total pembelian barang dengan pajak 10% dengan format dua desimal}*

*Deklarasi*

*Variabel:*

*hargaBarang, totalHarga: real*

*Deskripsi*

*start*

*// Membaca harga barang*

*read(hargaBarang)*

*// Menghitung total harga dengan pajak*

*totalHarga = hargaBarang + (hargaBarang \* 0.1)*

*// Menampilkan hasil dengan format dua desimal*

*write("Total Harga: ", totalHarga)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*float hargaBarang, totalHarga;*

*// Membaca harga barang*

*printf("Masukkan harga barang: ");*

*scanf("%f", &hargaBarang);*

*// Menghitung total harga dengan pajak*

*totalHarga = hargaBarang + (hargaBarang \* 0.1);*

*// Menampilkan hasil dengan format dua desimal*

*printf("Total Harga: %.2f\n", totalHarga);*

*return 0;*

*}*

**Penjelasan Program:**

Program membaca harga barang, menghitung total harga dengan menambahkan pajak sebesar 10%, dan menampilkan hasilnya dengan format dua angka di belakang koma.

**Soal 9: Menghitung Luas Segitiga**

**Kasus:**

Sebuah aplikasi geometri ingin menghitung luas segitiga berdasarkan panjang alas dan tinggi yang dimasukkan oleh pengguna. Hasil perhitungan harus ditampilkan dengan format dua angka di belakang koma.

**Penjelasan Kasus:**

* Input berupa panjang alas dan tinggi segitiga (float).
* Output berupa luas segitiga.
* Rumus yang digunakan adalah: Luas = 1/2 \* alas \* tinggi.

**Analisis Kebutuhan:**

* Diperlukan variabel alas, tinggi, dan luas.
* Operasi yang dilakukan adalah perkalian dan pembagian.

**Langkah-langkah Penyelesaian:**

1. Baca panjang alas dan tinggi.
2. Hitung luas segitiga.
3. Tampilkan hasil dengan format dua angka di belakang koma.

**Pseudocode:**

*Algoritma HitungLuasSegitiga (input, output)*

*{Menghitung luas segitiga dengan format dua desimal}*

*Deklarasi*

*Variabel:*

*alas, tinggi, luas: real*

*Deskripsi*

*start*

*// Membaca panjang alas dan tinggi segitiga*

*read(alas, tinggi)*

*// Menghitung luas segitiga*

*luas = 0.5 \* alas \* tinggi*

*// Menampilkan hasil dengan format dua desimal*

*write("Luas Segitiga: ", luas)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*float alas, tinggi, luas;*

*// Membaca panjang alas dan tinggi segitiga*

*printf("Masukkan panjang alas: ");*

*scanf("%f", &alas);*

*printf("Masukkan tinggi: ");*

*scanf("%f", &tinggi);*

*// Menghitung luas segitiga*

*luas = 0.5 \* alas \* tinggi;*

*// Menampilkan hasil dengan format dua desimal*

*printf("Luas Segitiga: %.2f\n", luas);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca panjang alas dan tinggi, menghitung luas segitiga menggunakan rumus, dan menampilkan hasil dengan format dua angka di belakang koma.

**Soal 10: Menghitung Gaji Karyawan dengan Bonus**

**Kasus:**

Sebuah perusahaan ingin menghitung gaji akhir seorang karyawan setelah ditambahkan bonus sebesar 5%. Gaji akhir harus ditampilkan dengan format dua angka di belakang koma.

**Penjelasan Kasus:**

* Input berupa gaji pokok karyawan.
* Output berupa gaji akhir setelah ditambahkan bonus.
* Bonus adalah 5% dari gaji pokok.

**Analisis Kebutuhan:**

* Diperlukan variabel gajiPokok, gajiAkhir.
* Operasi yang dilakukan adalah penambahan 5% dari gaji pokok.

**Langkah-langkah Penyelesaian:**

1. Baca gaji pokok.
2. Hitung gaji akhir dengan bonus.
3. Tampilkan hasil dengan format dua angka di belakang koma.

**Pseudocode:**

*Algoritma HitungGajiKaryawan (input, output)*

*{Menghitung gaji akhir karyawan dengan bonus 5% dengan format dua desimal}*

*Deklarasi*

*Variabel:*

*gajiPokok, gajiAkhir: real*

*Deskripsi*

*start*

*// Membaca gaji pokok karyawan*

*read(gajiPokok)*

*// Menghitung gaji akhir dengan bonus*

*gajiAkhir = gajiPokok + (gajiPokok \* 0.05)*

*// Menampilkan hasil dengan format dua desimal*

*write("Gaji Akhir: ", gajiAkhir)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*float gajiPokok, gajiAkhir;*

*// Membaca gaji pokok karyawan*

*printf("Masukkan gaji pokok: ");*

*scanf("%f", &gajiPokok);*

*// Menghitung gaji akhir dengan bonus*

*gajiAkhir = gajiPokok + (gajiPokok \* 0.05);*

*// Menampilkan hasil dengan format dua desimal*

*printf("Gaji Akhir: %.2f\n", gajiAkhir);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca gaji pokok karyawan, menambahkan bonus 5%, lalu menampilkan gaji akhir dengan format dua angka di belakang koma.

Berikut adalah kasus-kasus terkait **Penanganan Input dan Output yang tidak menggunakan format specifier** seperti getchar(), gets(), putchar(), puts(), tanpa menggunakan alur decisional maupun looping:

**Soal 1: Membaca dan Menampilkan Karakter**

**Kasus:**

Sebuah program diminta untuk membaca satu karakter dari input pengguna menggunakan fungsi getchar() dan kemudian menampilkannya kembali menggunakan putchar().

**Penjelasan Kasus:**

* Input berupa satu karakter yang dimasukkan oleh pengguna.
* Output adalah karakter yang sama yang ditampilkan.

**Analisis Kebutuhan:**

* Diperlukan variabel karakter untuk menyimpan karakter yang diinputkan.
* Fungsi getchar() untuk membaca input dan putchar() untuk menampilkan karakter.

**Langkah-langkah Penyelesaian:**

1. Baca satu karakter dari pengguna.
2. Tampilkan karakter yang dibaca menggunakan putchar().

**Pseudocode:**

*Algoritma BacaDanTampilKarakter (input, output)*

*{Membaca satu karakter dan menampilkannya kembali}*

*Deklarasi*

*Variabel:*

*karakter: char*

*Deskripsi*

*start*

*// Membaca satu karakter dari input*

*karakter = getchar()*

*// Menampilkan karakter*

*putchar(karakter)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*char karakter;*

*// Membaca satu karakter*

*karakter = getchar();*

*// Menampilkan karakter yang dibaca*

*putchar(karakter);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca satu karakter dari input menggunakan getchar(), lalu menampilkan karakter yang sama menggunakan putchar().

**Soal 2: Membaca dan Menampilkan String**

**Kasus:**

Sebuah aplikasi meminta pengguna untuk memasukkan satu kalimat menggunakan gets(), kemudian menampilkan kalimat yang sama menggunakan puts().

**Penjelasan Kasus:**

* Input berupa satu string (kalimat) yang dimasukkan oleh pengguna.
* Output berupa string yang sama yang ditampilkan.

**Analisis Kebutuhan:**

* Diperlukan array kalimat[] untuk menyimpan string.
* Fungsi gets() digunakan untuk membaca string, dan puts() untuk menampilkan string.

**Langkah-langkah Penyelesaian:**

1. Baca string dari pengguna menggunakan gets().
2. Tampilkan string tersebut menggunakan puts().

**Pseudocode:**

*Algoritma BacaDanTampilString (input, output)*

*{Membaca satu kalimat dan menampilkannya kembali}*

*Deklarasi*

*Variabel:*

*kalimat: array[100] of char*

*Deskripsi*

*start*

*// Membaca kalimat dari input*

*gets(kalimat)*

*// Menampilkan kalimat*

*puts(kalimat)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*char kalimat[100];*

*// Membaca kalimat dari input*

*gets(kalimat);*

*// Menampilkan kalimat*

*puts(kalimat);*

*return 0;*

*}*

**Penjelasan Program:**

Program membaca sebuah string dari input menggunakan gets(), lalu menampilkan string tersebut menggunakan puts().

**Soal 3: Membaca Dua Karakter Berurutan**

**Kasus:**

Program diminta untuk membaca dua karakter berurutan dari input pengguna menggunakan dua kali pemanggilan getchar(), dan menampilkan kedua karakter tersebut menggunakan putchar().

**Penjelasan Kasus:**

* Input berupa dua karakter yang dimasukkan satu per satu oleh pengguna.
* Output adalah kedua karakter yang ditampilkan secara berurutan.

**Analisis Kebutuhan:**

* Diperlukan variabel char1 dan char2 untuk menyimpan dua karakter.
* Fungsi getchar() digunakan untuk membaca masing-masing karakter, dan putchar() untuk menampilkannya.

**Langkah-langkah Penyelesaian:**

1. Baca karakter pertama menggunakan getchar().
2. Baca karakter kedua menggunakan getchar().
3. Tampilkan kedua karakter menggunakan putchar().

**Pseudocode:**

*Algoritma BacaDuaKarakter (input, output)*

*{Membaca dua karakter berurutan dan menampilkannya}*

*Deklarasi*

*Variabel:*

*char1, char2: char*

*Deskripsi*

*start*

*// Membaca dua karakter dari input*

*char1 = getchar()*

*char2 = getchar()*

*// Menampilkan kedua karakter*

*putchar(char1)*

*putchar(char2)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*char char1, char2;*

*// Membaca dua karakter berurutan*

*char1 = getchar();*

*char2 = getchar();*

*// Menampilkan kedua karakter*

*putchar(char1);*

*putchar(char2);*

*return 0;*

*}*

**Penjelasan Program:**

Program membaca dua karakter berurutan menggunakan getchar() dan menampilkan kedua karakter tersebut menggunakan putchar().

**Soal 4: Menampilkan String Statis**

**Kasus:**

Program akan menampilkan string statis yang sudah ditetapkan di dalam kode menggunakan fungsi puts().

**Penjelasan Kasus:**

* Tidak ada input dari pengguna.
* Output adalah string yang ditetapkan dalam kode dan ditampilkan ke layar.

**Analisis Kebutuhan:**

* Diperlukan string statis yang ditetapkan dalam kode.
* Fungsi puts() digunakan untuk menampilkan string.

**Langkah-langkah Penyelesaian:**

1. Tentukan string statis.
2. Tampilkan string tersebut menggunakan puts().

**Pseudocode:**

*Algoritma TampilStringStatis (input, output)*

*{Menampilkan string statis ke layar}*

*Deklarasi*

*Variabel:*

*pesan: array[20] of char = "Selamat datang!"*

*Deskripsi*

*start*

*// Menampilkan string statis*

*puts(pesan)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*// Menampilkan string statis*

*puts("Selamat datang!");*

*return 0;*

*}*

**Penjelasan Program:**

Program ini langsung menampilkan string statis "Selamat datang!" ke layar menggunakan puts().

**Soal 5: Membaca dan Menampilkan Satu Kata**

**Kasus:**

Sebuah program meminta pengguna untuk memasukkan satu kata menggunakan gets() dan kemudian menampilkannya kembali menggunakan puts().

**Penjelasan Kasus:**

* Input berupa satu kata dari pengguna.
* Output adalah kata yang sama yang ditampilkan.

**Analisis Kebutuhan:**

* Diperlukan array kata[] untuk menyimpan kata yang dimasukkan.
* Fungsi gets() digunakan untuk membaca kata, dan puts() untuk menampilkannya.

**Langkah-langkah Penyelesaian:**

1. Baca satu kata dari pengguna menggunakan gets().
2. Tampilkan kata tersebut menggunakan puts().

**Pseudocode:**

*Algoritma BacaDanTampilSatuKata (input, output)*

*{Membaca satu kata dan menampilkannya kembali}*

*Deklarasi*

*Variabel:*

*kata: array[50] of char*

*Deskripsi*

*start*

*// Membaca satu kata dari input*

*gets(kata)*

*// Menampilkan kata*

*puts(kata)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*char kata[50];*

*// Membaca satu kata dari input*

*gets(kata);*

*// Menampilkan kata*

*puts(kata);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini meminta pengguna memasukkan satu kata, lalu menampilkan kata yang sama menggunakan puts().

**Soal 6: Membaca dan Menampilkan Tiga Karakter**

**Kasus:**

Program diminta untuk membaca tiga karakter satu per satu dari input pengguna menggunakan getchar(), dan kemudian menampilkannya kembali satu per satu menggunakan putchar().

**Penjelasan Kasus:**

* Input berupa tiga karakter dari pengguna.
* Output berupa tiga karakter yang ditampilkan satu per satu.

**Analisis Kebutuhan:**

* Diperlukan tiga variabel char1, char2, dan char3 untuk menyimpan masing-masing karakter.
* Fungsi getchar() digunakan untuk membaca setiap karakter, dan putchar() untuk menampilkannya.

**Langkah-langkah Penyelesaian:**

1. Baca karakter pertama menggunakan getchar().
2. Baca karakter kedua menggunakan getchar().
3. Baca karakter ketiga menggunakan getchar().
4. Tampilkan ketiga karakter menggunakan putchar().

**Pseudocode:**

*Algoritma BacaTigaKarakter (input, output)*

*{Membaca tiga karakter satu per satu dan menampilkannya kembali}*

*Deklarasi*

*Variabel:*

*char1, char2, char3: char*

*Deskripsi*

*start*

*// Membaca tiga karakter*

*char1 = getchar()*

*char2 = getchar()*

*char3 = getchar()*

*// Menampilkan tiga karakter*

*putchar(char1)*

*putchar(char2)*

*putchar(char3)*

*stop*

**Implementasi C:**

*#include <stdio.h>*

*int main() {*

*char char1, char2, char3;*

*// Membaca tiga karakter*

*char1 = getchar();*

*char2 = getchar();*

*char3 = getchar();*

*// Menampilkan tiga karakter*

*putchar(char1);*

*putchar(char2);*

*putchar(char3);*

*return 0;*

*}*

**Penjelasan Program:**

Program ini membaca tiga karakter secara berurutan dari input pengguna, kemudian menampilkannya kembali satu per satu menggunakan putchar().