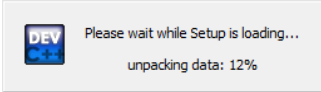
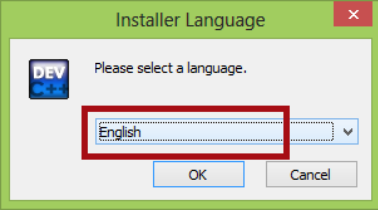
1. Install software Dev C++, buatlah program Hello World dengan bahasa c, Dimana program mencetak Nama dan NIM, Berikan screenshoot dari step penginstallan program dan beri penjelasan program yang dibuat

**Cara penginstalan:**

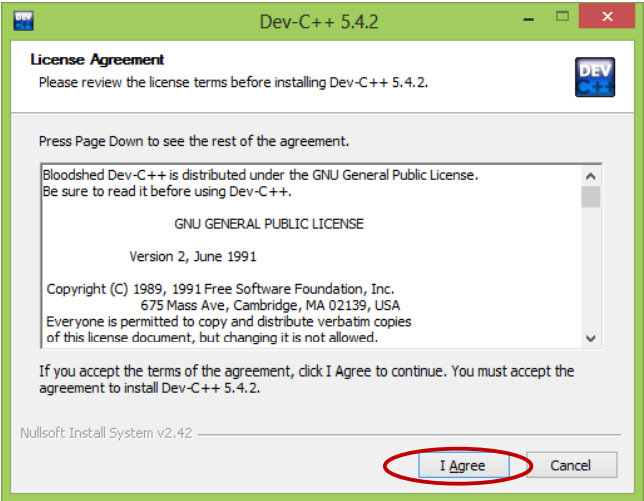
1. Download di website resmi dev C++
2. Buka file downloadnya dan tunggu loadingnya



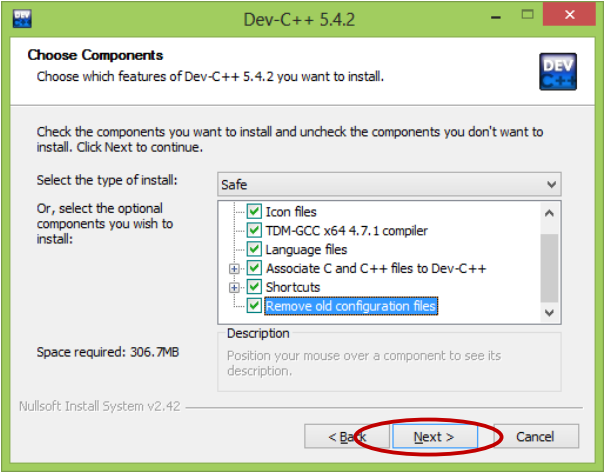
1. Pilih Bahasa



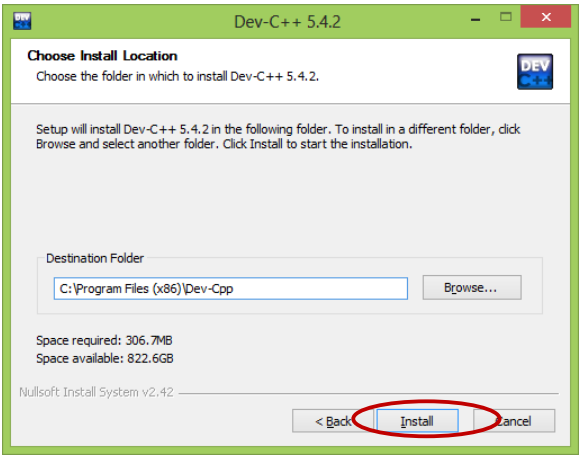
1. Baca panduan dan pilih *I Agree*



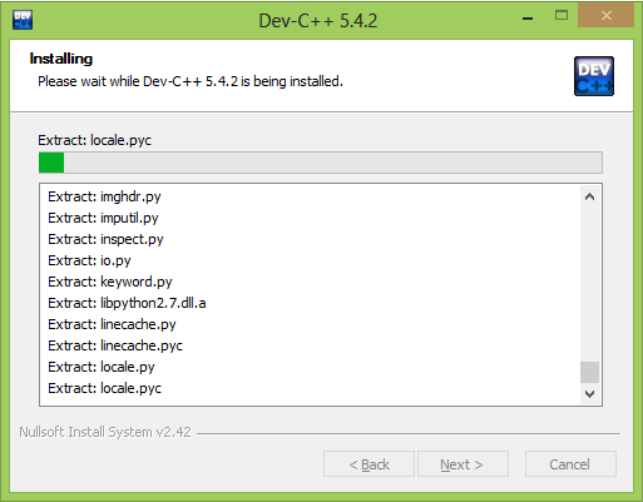
1. Next Langsung, settingan default sudah cukup



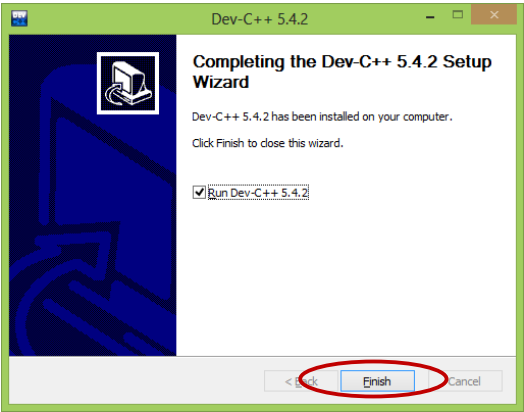
1. Pilih lokasi penginstalan



1. Loading…



1. Selesai



**Program Mencetak Nama dan NIM menggunakan Bahasa C**

#include <stdio.h>

int main() {

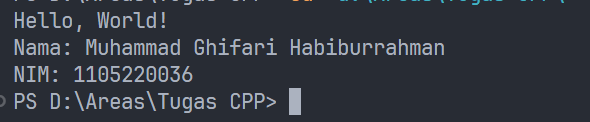
    printf("Hello, World!\n");

    printf("Nama: Muhammad Ghifari Habiburrahman\n");

    printf("NIM: 1105220036\n");

    return 0;

}

****

1. Buatlah algoritma kasus berikut, buatlah notasi algoritmik serta flowchart programnya:
2. Menampilkan bilangan genap/ganjil dari angka yang diinputkan oleh user

**Notasi Algoritmik:**

#include <stdio.h>

int main() {

    int num;

*// Meminta pengguna untuk memasukkan angka*

    printf("Masukkan sebuah angka: ");

    scanf("%d", &num);

*// Memeriksa apakah angka tersebut genap atau ganjil dan menampilkannya*

    if (num % 2 == 0) {

        printf("%d adalah bilangan genap.\n", num);

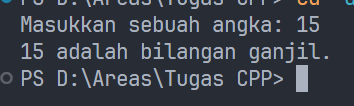
    } else {

        printf("%d adalah bilangan ganjil.\n", num);

    }

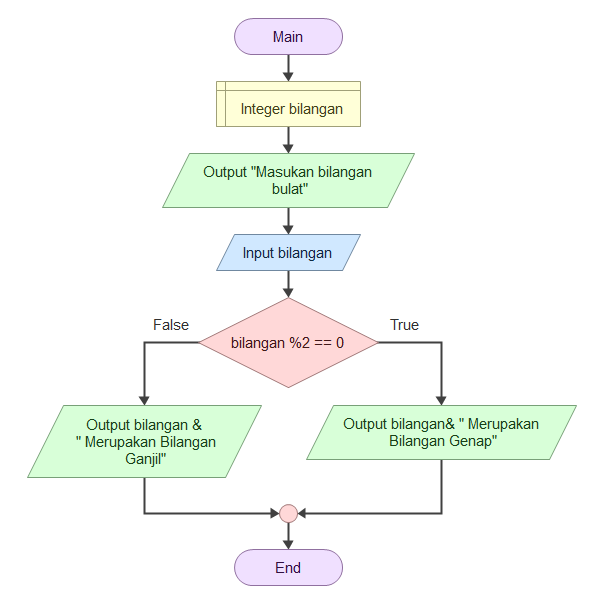
    return 0;

}



**Flowchart:**

Output:



1. Menghitung hambatan total pada rangkaian seri dan paralel

**Notasi Algoritmik:**

#include <stdio.h>

*// Fungsi untuk menghitung hambatan total pada rangkaian seri*

float hitung\_hambatan\_seri(float \**resistors*, int *num\_resistors*) {

    float total\_resistance = 0;

    for (int i = 0; i < num\_resistors; ++i) {

        total\_resistance += resistors[i];

    }

    return total\_resistance;

}

*// Fungsi untuk menghitung hambatan total pada rangkaian paralel*

float hitung\_hambatan\_paralel(float \**resistors*, int *num\_resistors*) {

    float total\_resistance = 0;

    for (int i = 0; i < num\_resistors; ++i) {

        total\_resistance += 1 / resistors[i];

    }

    return 1 / total\_resistance;

}

int main() {

    int num\_resistors;

    printf("Masukkan jumlah resistor: ");

    scanf("%d", &num\_resistors);

    float resistors[num\_resistors];

    printf("Masukkan nilai hambatan resistor:\n");

    for (int i = 0; i < num\_resistors; ++i) {

        printf("Resistor %d: ", i + 1);

        scanf("%f", &resistors[i]);

    }

*// Menghitung hambatan total untuk rangkaian seri*

    float total\_seri = hitung\_hambatan\_seri(resistors, num\_resistors);

    printf("Hambatan total pada rangkaian seri: %.2f ohm\n", total\_seri);

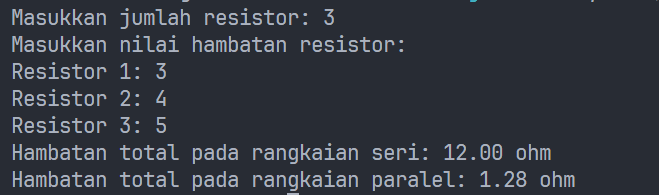
*// Menghitung hambatan total untuk rangkaian paralel*

    float total\_paralel = hitung\_hambatan\_paralel(resistors, num\_resistors);

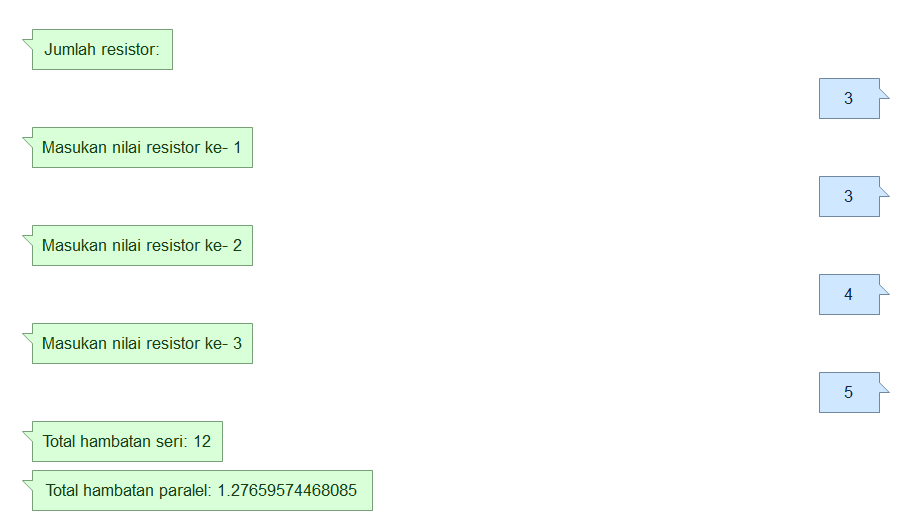
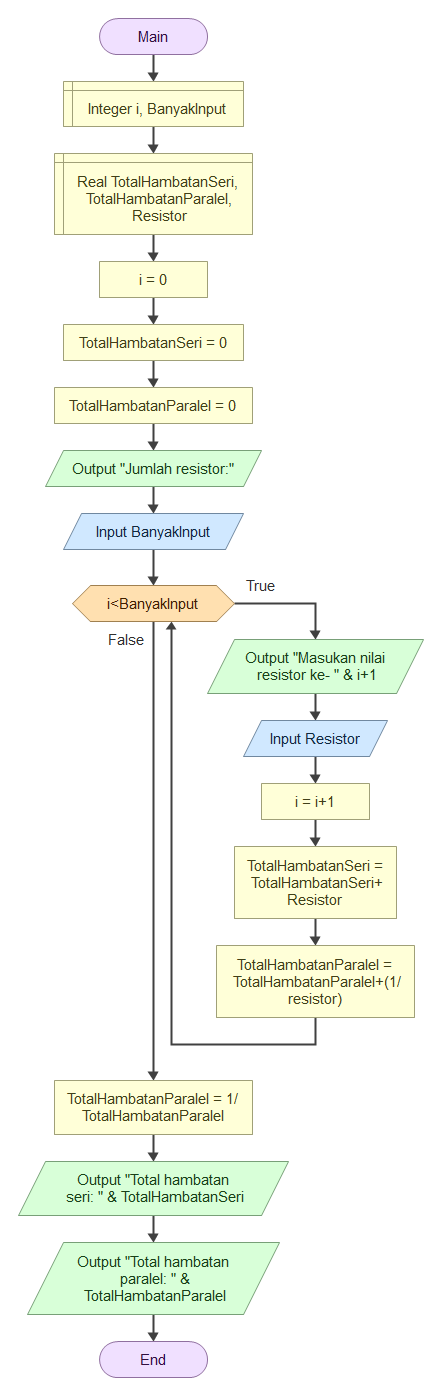
    printf("Hambatan total pada rangkaian paralel: %.2f ohm\n", total\_paralel);

    return 0;

}



**Flowchart:**

****

Output:

1. Buatlah sebuah kasus, beserta deskripsinya, notasi algoritmik dan flowchartnya.
2. Kasus

Penilaian Mahasiswa

Deskripsi:

Sebuah program untuk menghitung nilai akhir mahasiswa berdasarkan nilai ujian, tugas, dan kehadiran. Nilai akhir dihitung berdasarkan bobot masing-masing komponen.

1. Notasi Algoritmik

#include <stdio.h>

int main() {

    float nilaiUjian, nilaiTugas, jumlahKehadiran, nilaiAkhir;

*// Input nilai ujian*

    printf("Masukkan nilai ujian: ");

    scanf("%f", &nilaiUjian);

*// Input nilai tugas*

    printf("Masukkan nilai tugas: ");

    scanf("%f", &nilaiTugas);

*// Input jumlah kehadiran*

    printf("Masukkan jumlah kehadiran: ");

    scanf("%f", &jumlahKehadiran);

*// Hitung nilai akhir*

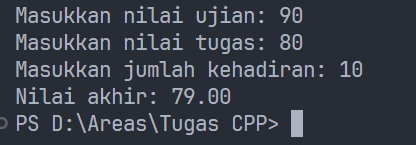
    nilaiAkhir = (nilaiUjian \* 0.6) + (nilaiTugas \* 0.3) + (jumlahKehadiran \* 0.1);

*// Tampilkan nilai akhir*

    printf("Nilai akhir: %.2f\n", nilaiAkhir);

    return 0;

}



1. Flowchart

