**TUGAS PRAKTIKUM 13**

**DASAR FUNGSI**



**Nama : Devafilla Rizqy Santosa**

**Kelas : 1 D4 Teknik Informatika A**

**NRP : 3120600026**

1. Tugas Pendahuluan.
   1. A. Listing

#include <stdio.h>

#include <stdlib.h>

void menu();

int main()

{

int angka, bil;

printf("Menampilkan menu dengan input angka : ");

scanf("%d", &angka);

for (bil=0; bil<angka; bil++)

menu();

}

void menu()

{

printf("Macam macam menu: \n");

printf("1. Fungsi triangular\n");

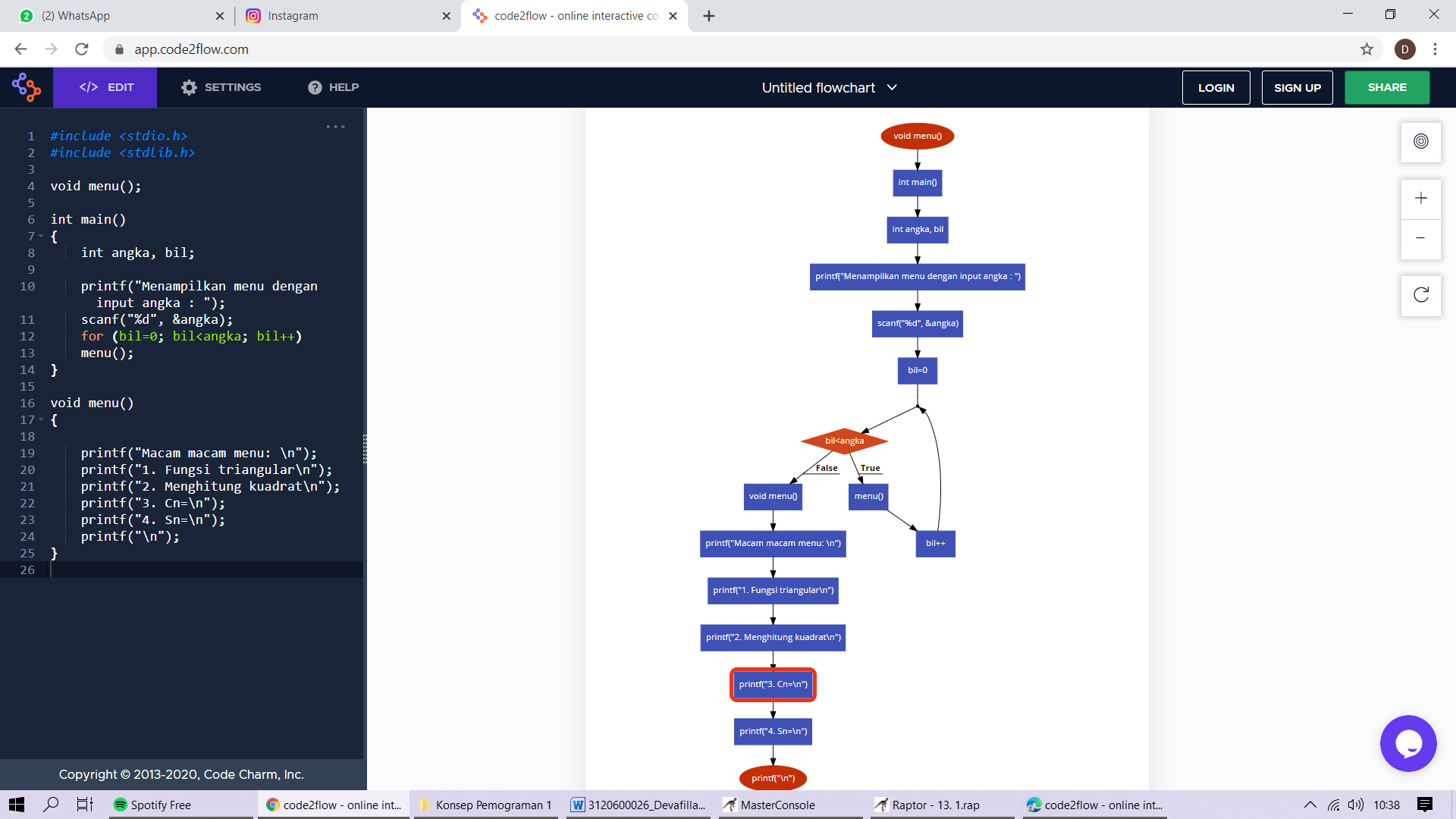
printf("2. Menghitung kuadrat\n");

printf("3. Cn=\n");

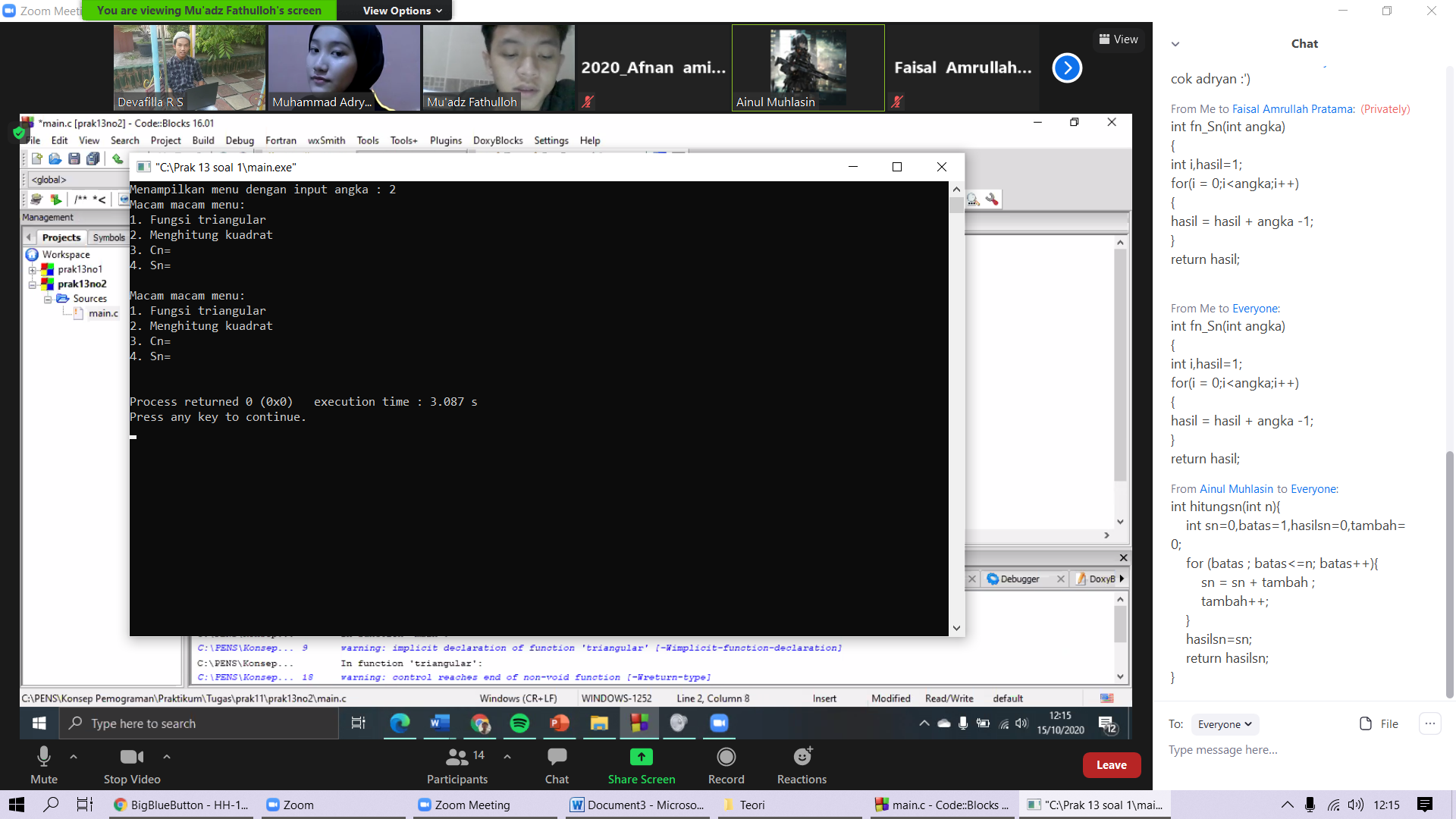
printf("4. Sn=\n");

printf("\n");

}



1. Output



1. Analisa

Sebuah program yang menampilkan berapa kali menu sejumlah input dari user dengan menggunakan fungsi tanpa return value. Dengan menggunkan menu() yang dikombinasi dengan loop for, maka fungsi menu() ditulis Kembali menjadi sebuah sub main yang nantinya berhubungan dengan int main.

* 1. A. Listing

#include <stdio.h>

#include <stdlib.h>

int triangular();

int main()

{

int bil;

printf("Menghitung jumlah triangualar\n");

printf("Masukan angka = ");

scanf("%d",&bil);

triangular(bil);

}

int triangular(bil)

{

int i, hasil=0;

for(i=1; i<=bil; bil--)

{

printf(" %d +",bil);

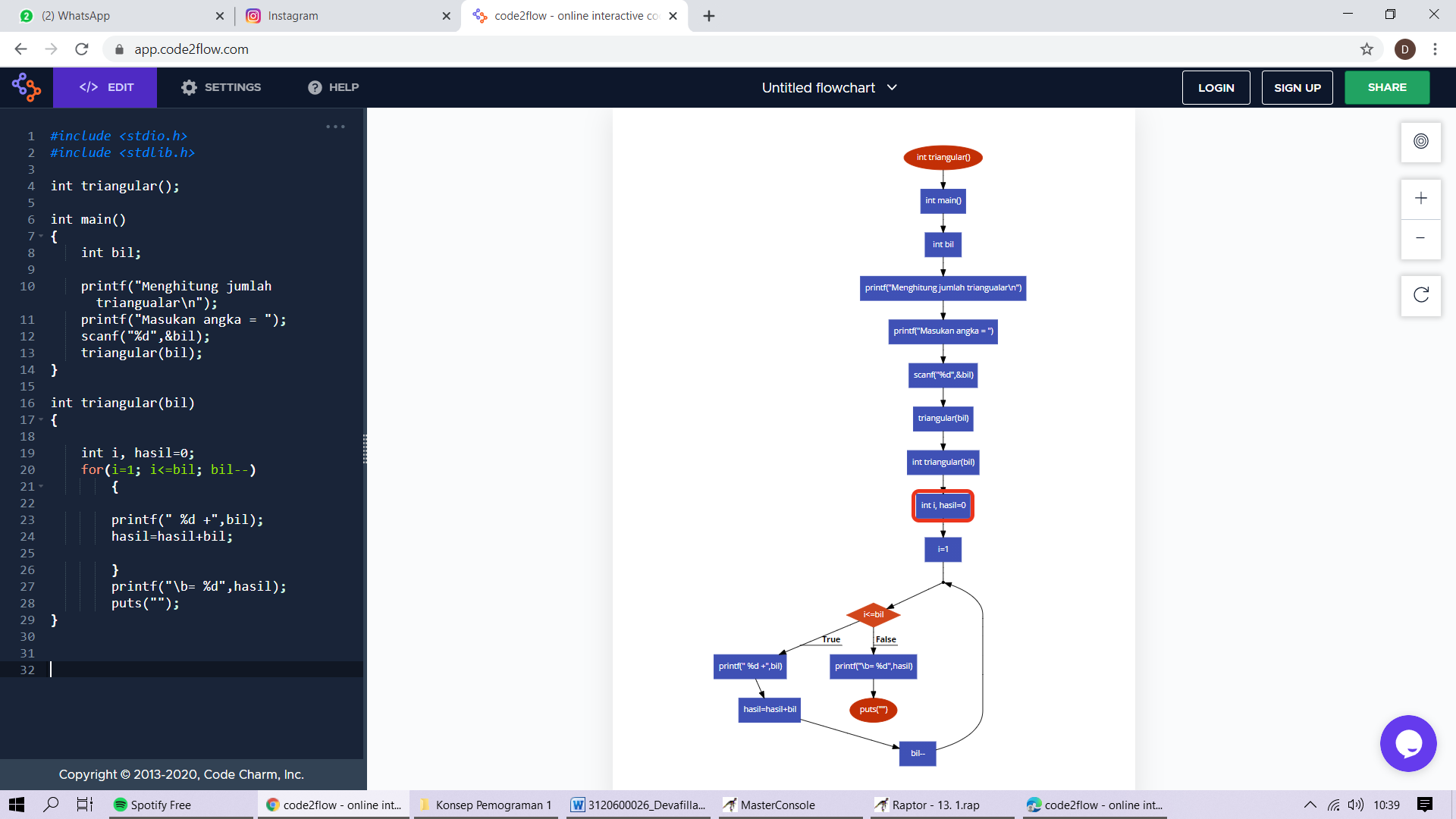
hasil=hasil+bil;

}

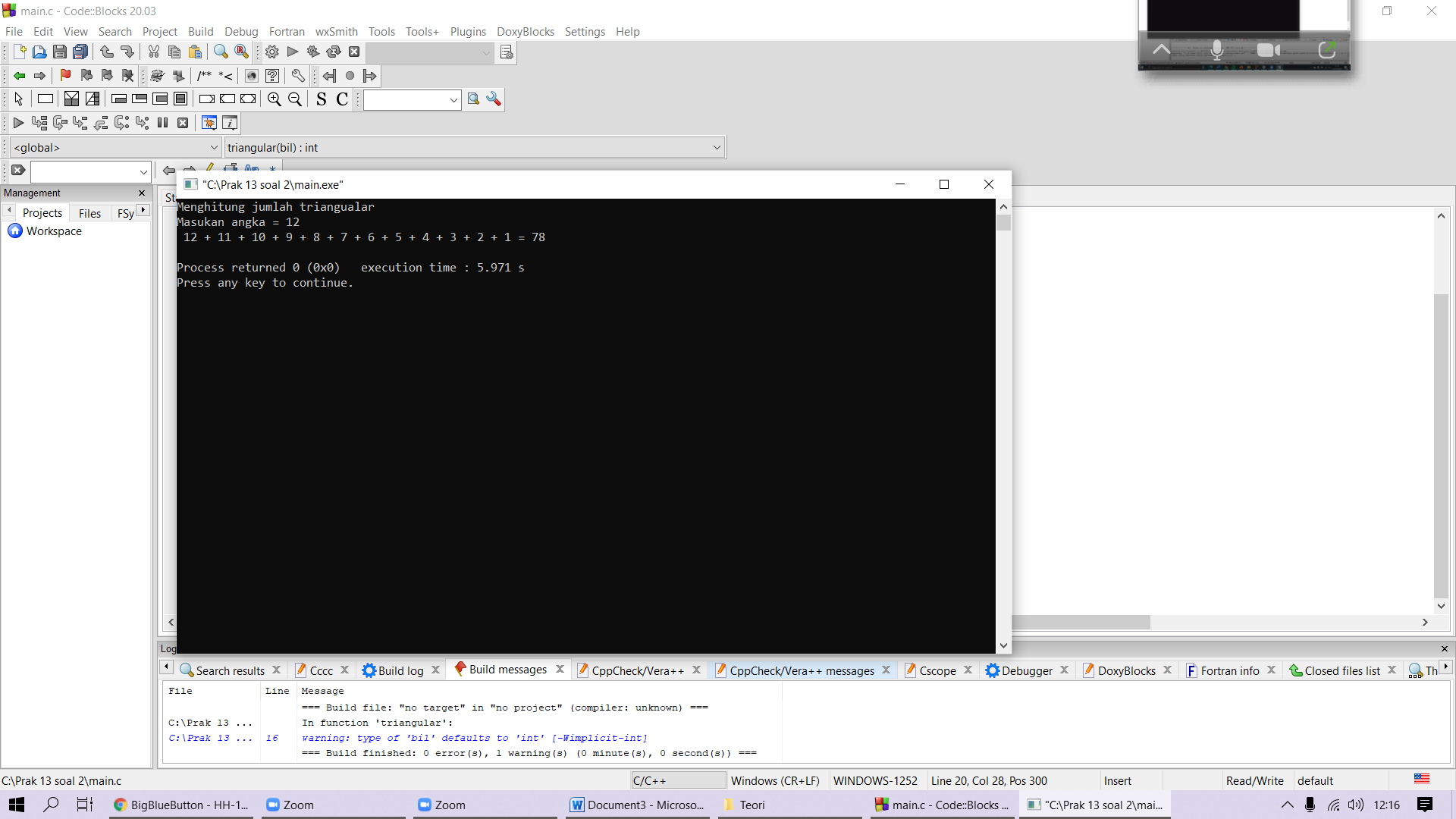
printf("\b= %d",hasil);

puts("");

}



1. Output



1. Analisa

Sebuah program yang dapat menampilkan penghitungan triangular dengan menggunakan fungsi tanpa return value. Saat menginputkan bil lalu dilanjutkan dengan menggunakan fungsi triangular(bil) untuk yang nantinya dapat dilanjutkan di sub main. Lalu untuk sub main digunakan rumus untuk menghitung triangular.

* 1. A. Listing

#include <stdio.h>

#include <stdlib.h>

float kuadrat(float);

int main()

{

float bil, hasil;

printf("Input angka kuadrat : ");

scanf("%f",&bil);

hasil = kuadrat(bil);

printf("Hasil kuadrat %.2f",hasil);

puts("");

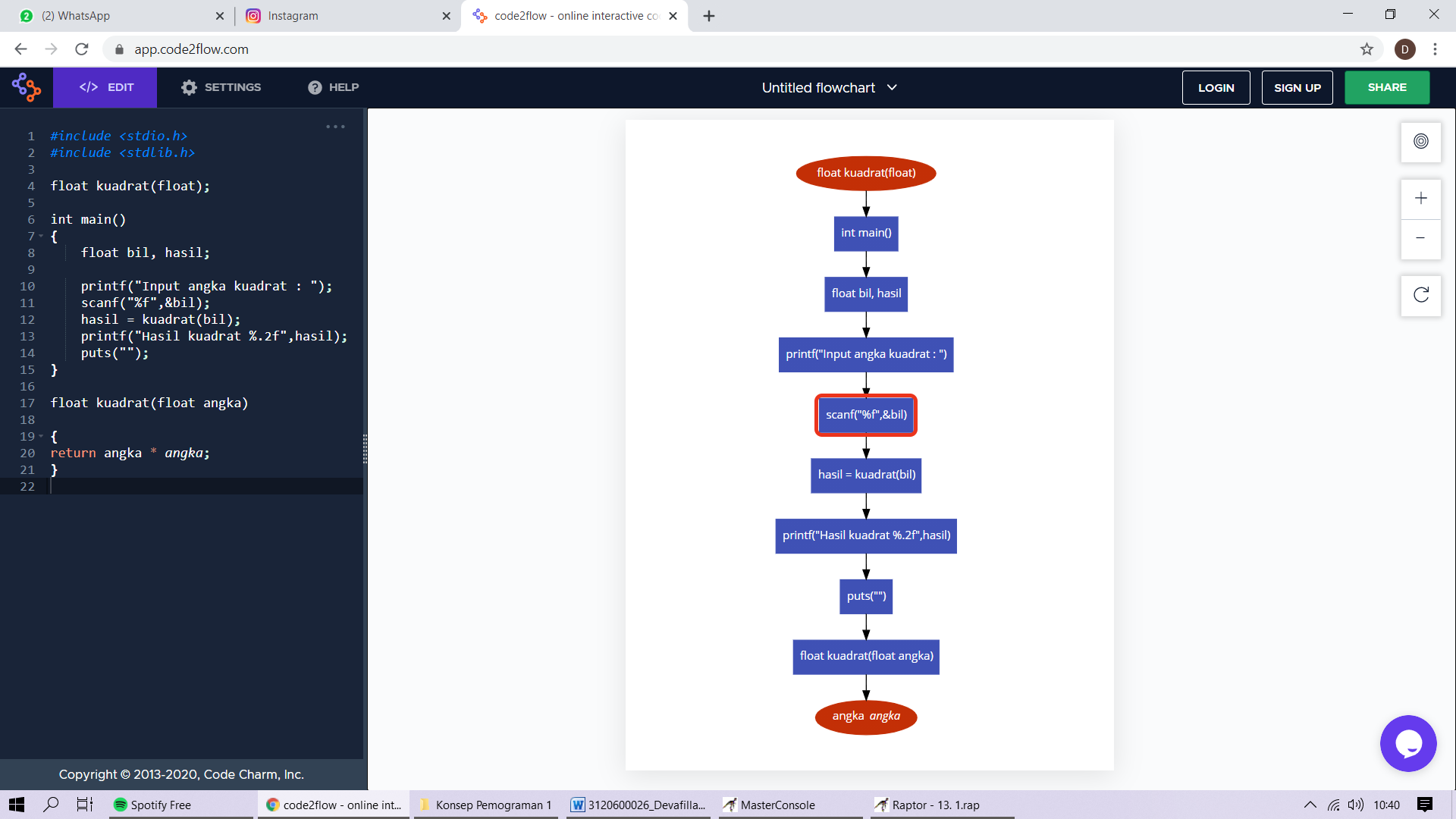
}

float kuadrat(float angka)

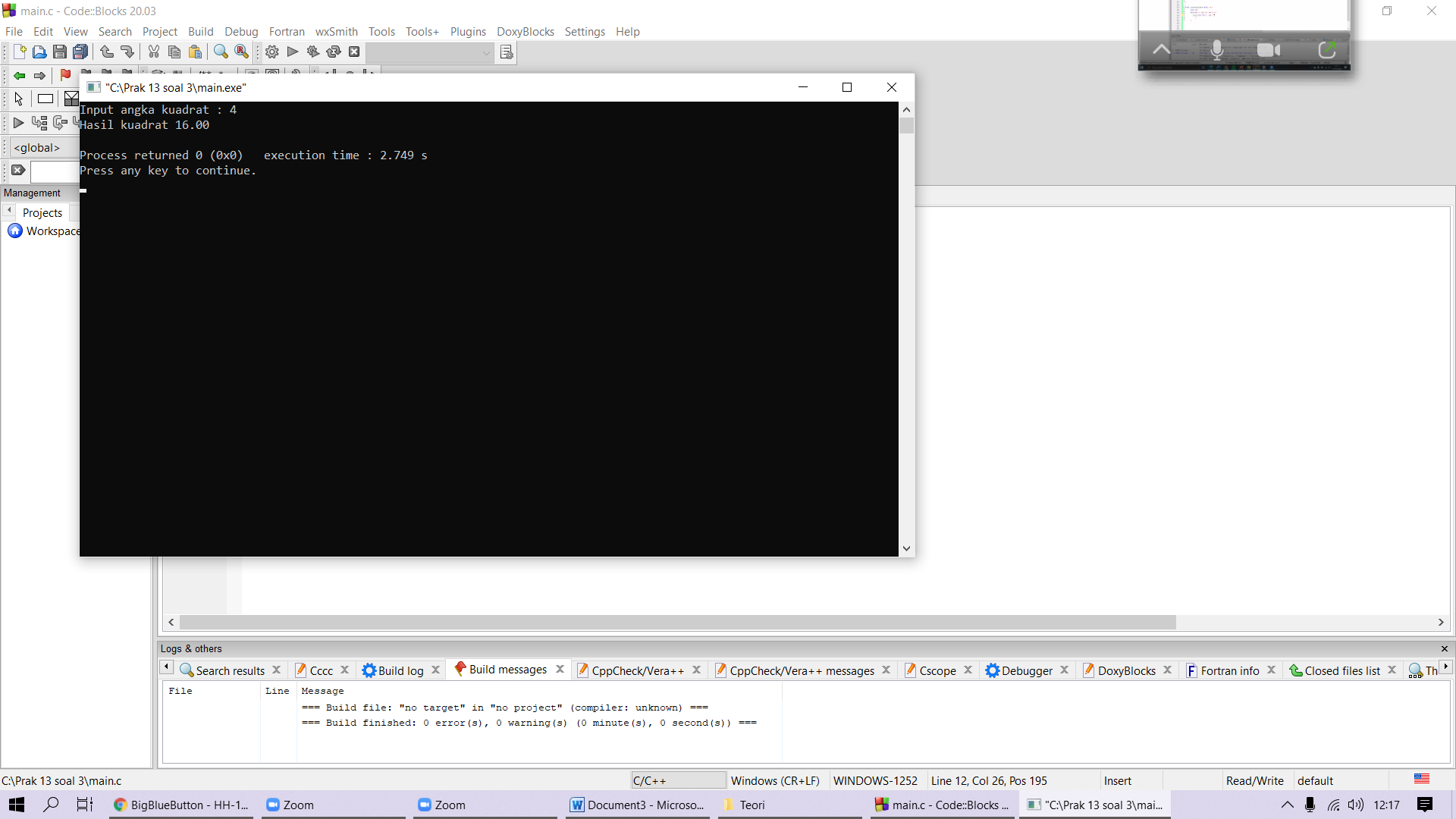
{

return angka \* angka;

}



B. Output



C.Analisa

Sebuah program yang dapat menampilkan bilangan kuadrat dengan menggunakan fungsi dan return value. Dengan menggunakan scanf dan hasil=fungsi kuadrat(bil) dan proses nantinya akan di lakukan di sub main. Lalu untuk sub main diisi dengan rumus untuk mencari hasil kuadrat dari bilangan input, untuk mengembalikan hasil proses digunakan return hasil. Dan di bagian main digunakan printf hasil maka nantinya hasil akan dapat keluar.

* 1. .1A. Listing

#include <stdio.h>

#include <stdlib.h>

int Cn(int);

int main()

{

int bil, hasil;

printf("Masukkan bilangan : ");

scanf("%d",&bil);

hasil = Cn (bil);

printf("Hasil Cn : %d", hasil);

printf("\n");

}

int Cn(int angka)

{

int n, hasil=1;

for(n=0; n<angka; n++)

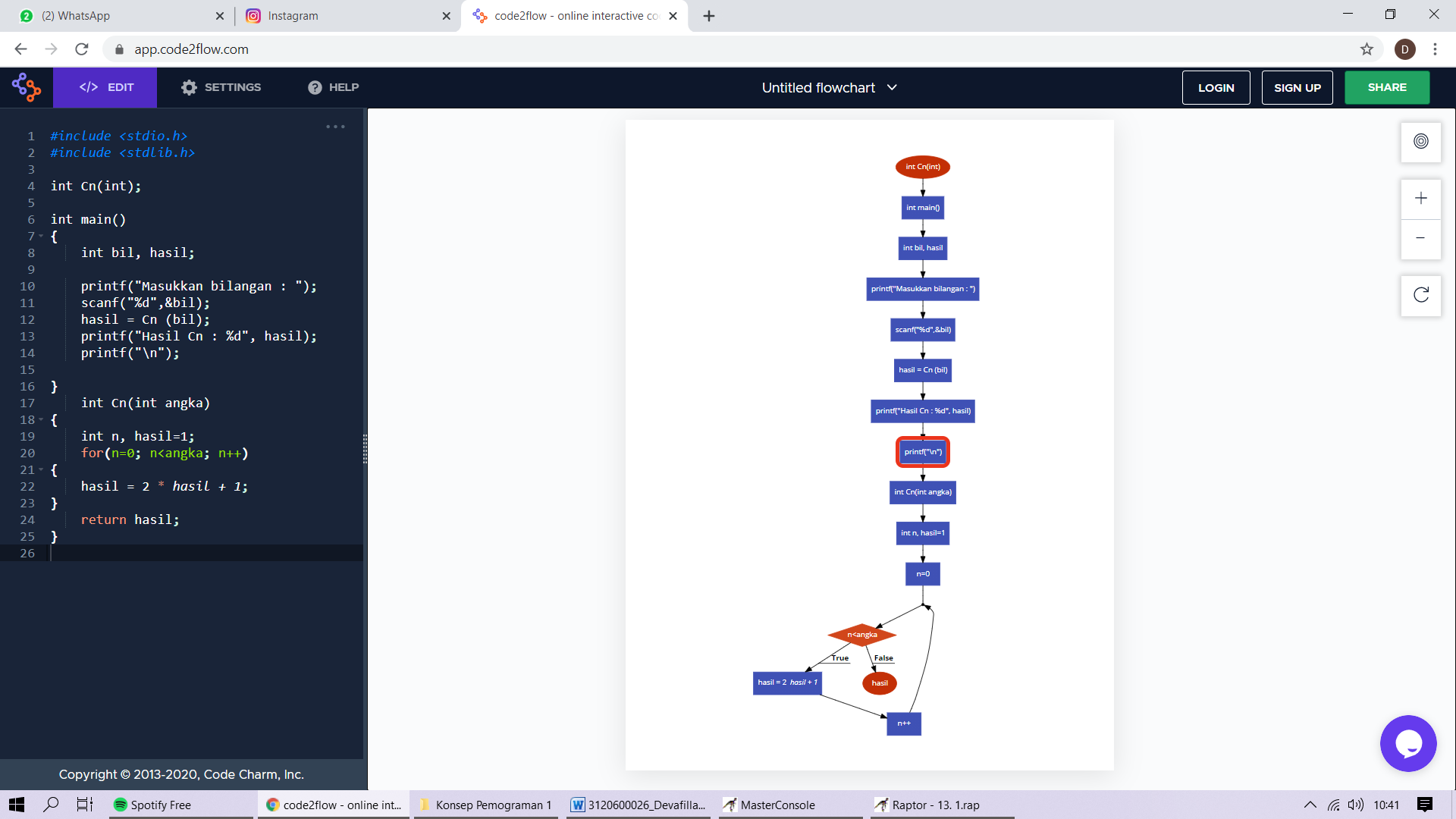
{

hasil = 2 \* hasil + 1;

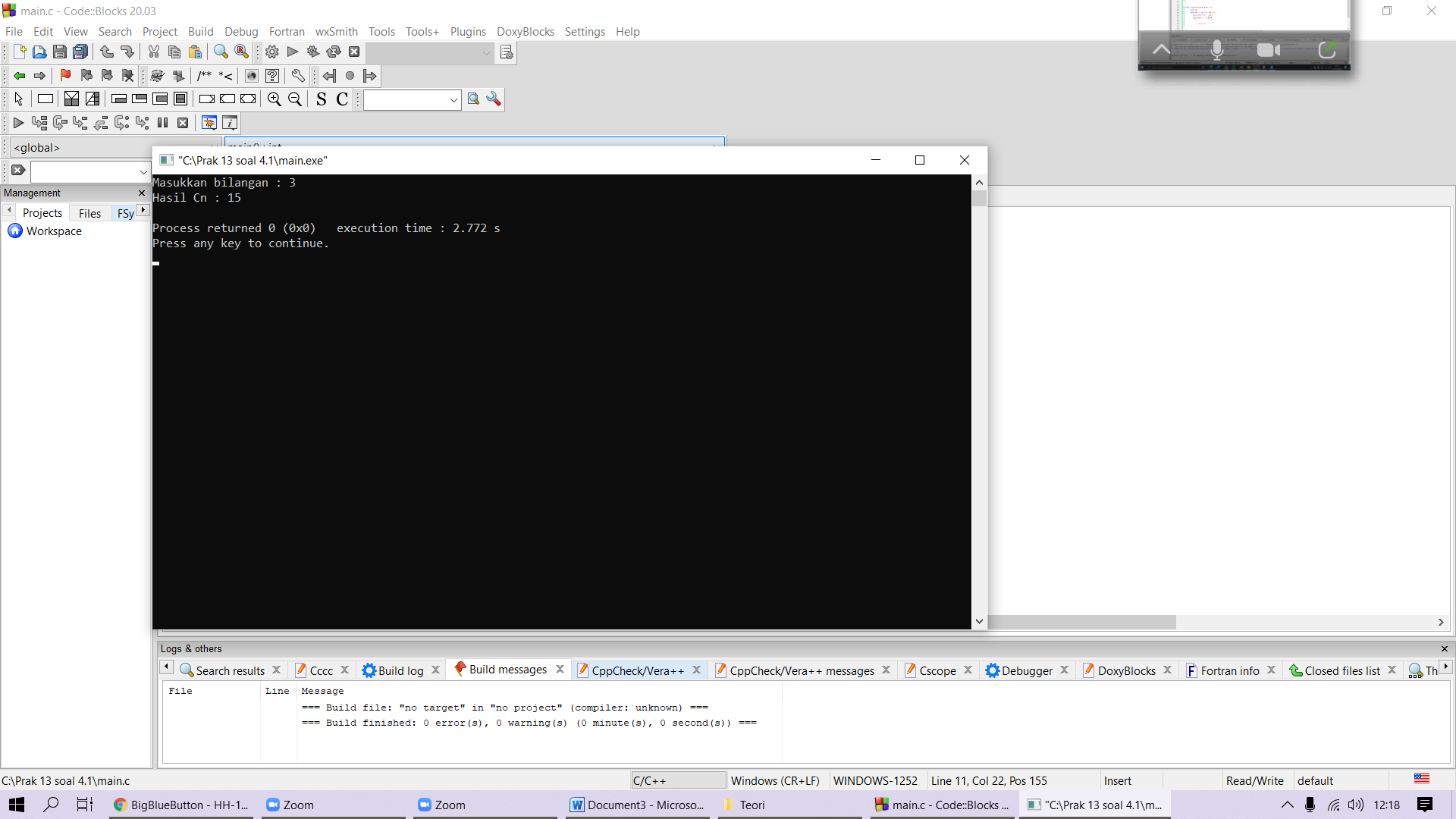
}

return hasil;

}



B. Output



C.Analisa

Sebuah program yang dapat menghitung fungsi di atas dengan return value. Maka program menggunakan scanf dengan fungsi hasilcn(n) dan hasilsn(n) yang nantinya dilakukan proses penghitugan di sub main. Untuk sub main setelah dilakukan proses menghitung lalu di gunakan return hasilcn dan return hasilsn yang nantinya hasil proses penghitugan akan dibawa ke main untuk dapat ditampilkan nilainya.

* 1. .2A. Listing

#include <stdio.h>

#include <stdlib.h>

int Sn(int);

int main()

{

int bil, hasil;

printf("Masukkan bilangan : ");

scanf("%d",&bil);

hasil = Sn(bil);

printf("Hasil Sn : %d", hasil);

printf("\n");

}

int Sn(int angka)

{

int n, hasil=0, tambah=0;

for(n=0; n<angka; n++)

{

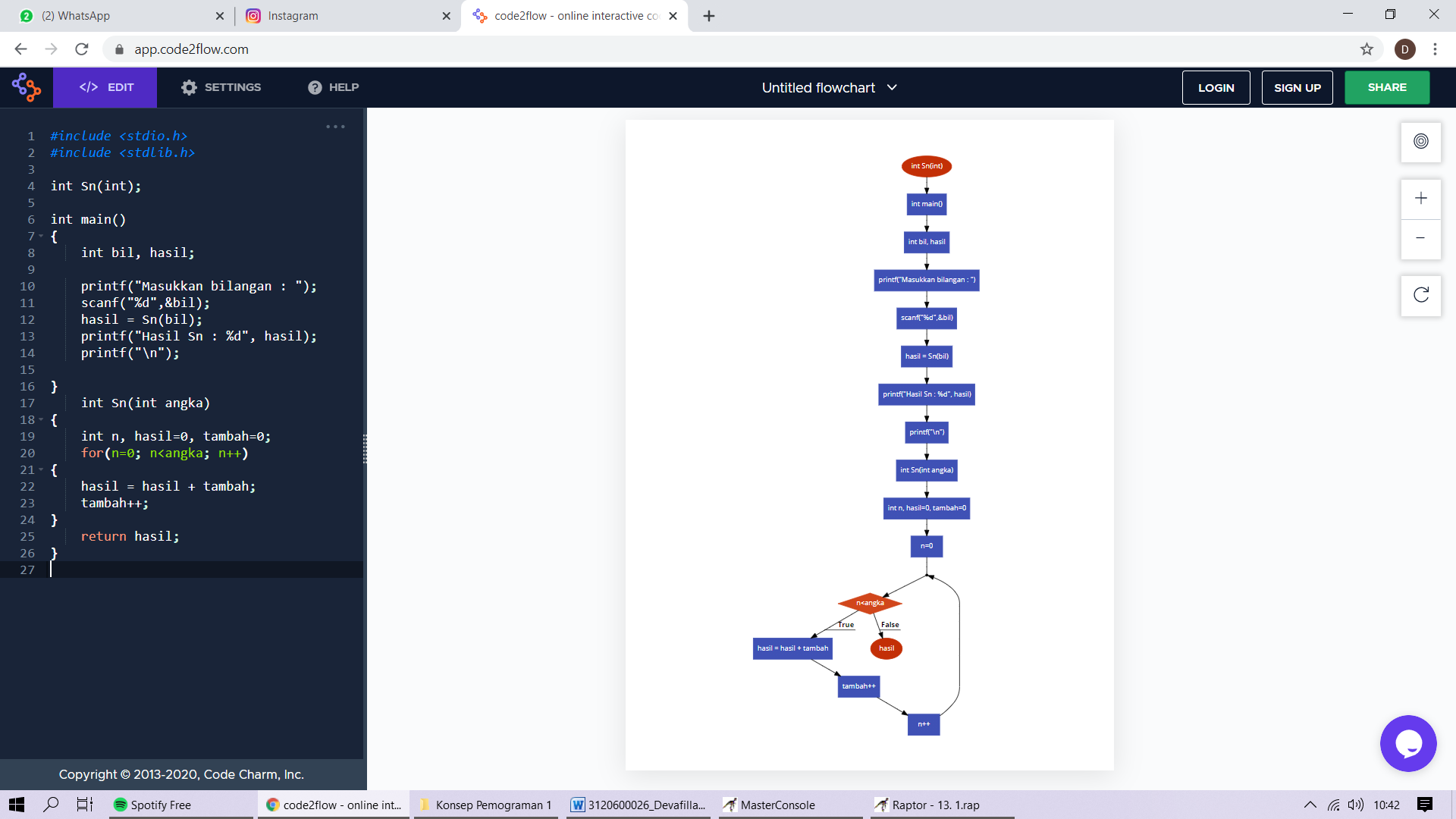
hasil = hasil + tambah;

tambah++;

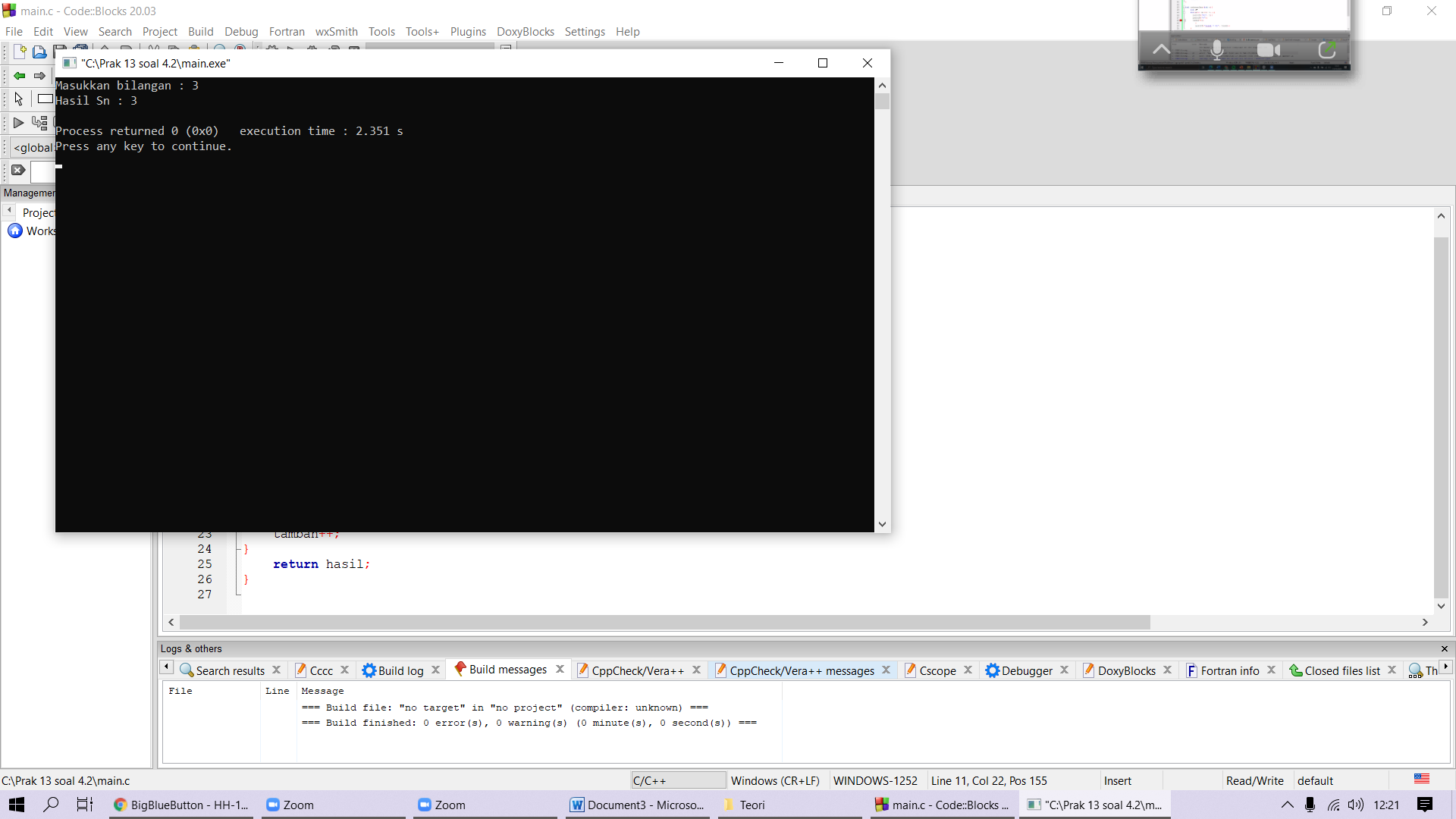
}

return hasil;

}



B. Output



C.Analisa

Sebuah program yang dapat menghitung fungsi di atas dengan return value. Maka program menggunakan scanf dengan fungsi hasilcn(n) dan hasilsn(n) yang nantinya dilakukan proses penghitugan di sub main. Untuk sub main setelah dilakukan proses menghitung lalu di gunakan return hasilcn dan return hasilsn yang nantinya hasil proses penghitugan akan dibawa ke main untuk dapat ditampilkan nilainya.

1. Kesimpulan

Fungsi dalam C merupakan sekumpulan code yang dibungkus ke dalam sebuah scope untuk menjalankan tugas tertentu. Fungsi memungkinkan kita untuk menghemat penulisan kode yang mana sering ditulis berulang-ulang, namun bisa dibungkus ke dalam sebuah fungsi dan kita tinggal memanggilnya saja.