

**LAPORAN PRAKTIKUM 11 dan 13**

“Nested Looping Break Continue”

dan

“Fungsi Dasar”

Mata kuliah Praktikum Konsep Pemrograman



Nama dosen pengampu:

Ibu Entin Martiana Kusumaningtyas S.Kom, M.kom.

Dikerjakan oleh

Nama : Muhammad Rafi Rizaldi

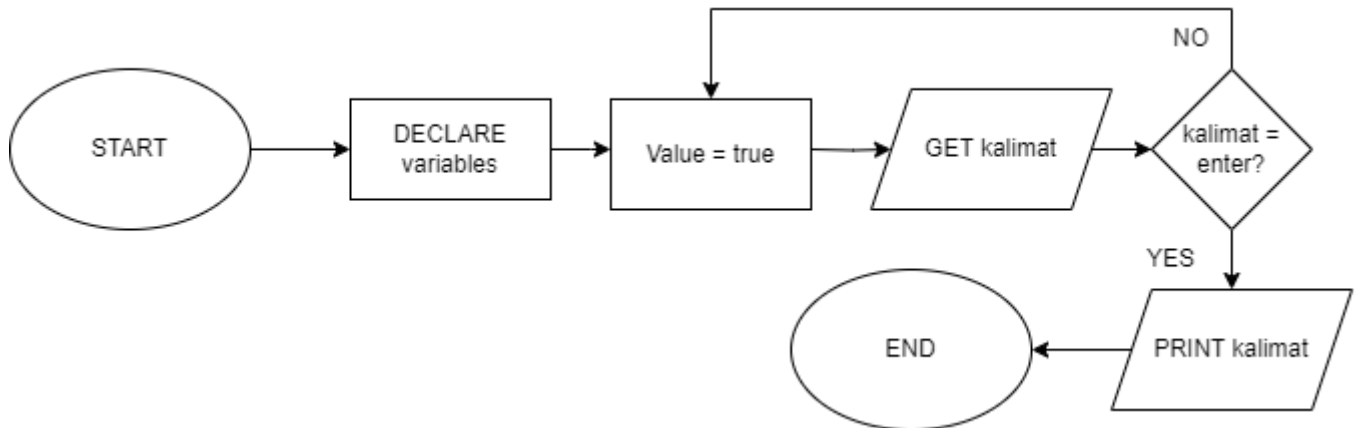
NRP : 3123600001

Kelas : 1 D4 IT A

## PRAKTIKUM 11 :

1. Dengan menggunakan pernyataan break, buatlah sebuah program yang dapat menampilkan semua karakter yang diketikkan dan program berakhir ketika ditekan tombol enter.

Flowchart :



Input :

```
#include<stdio.h>
int main(){
    char kalimat;
    printf("Masukkan kalimat : ");
    while (1)
    {
        scanf("%c", &kalimat);
        if (kalimat == '\n') break;
        printf("%c", kalimat);
    }
}
```

Output :

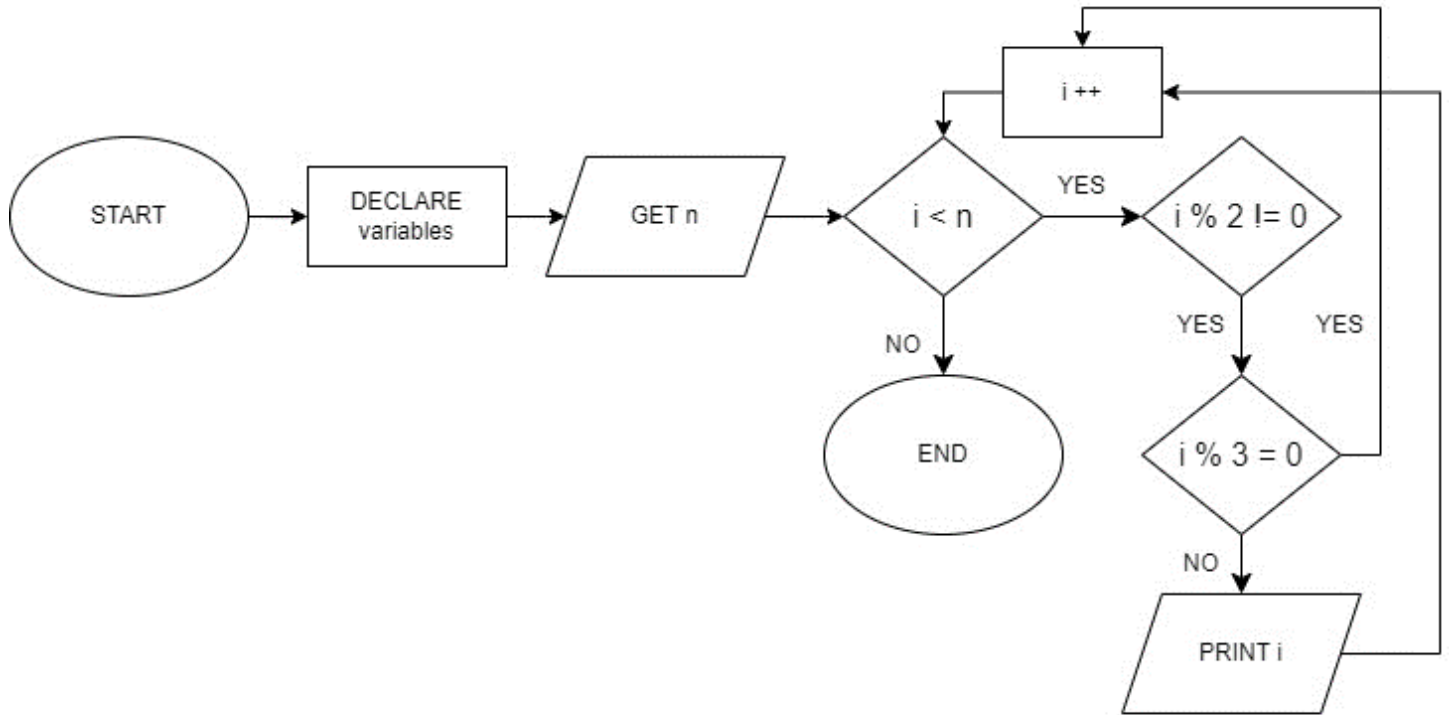
```
Masukkan kalimat : gfgdfgg
gfgdfgg
```

Analisa :

Program C tersebut merupakan program c sederhana yang menggunakan looping while dalam menginputkan kalimat dan looping akan berhenti ketika pengguna menginputkan enter.

2. Dengan menggunakan pernyataan continue, buatlah sebuah program yang dapat menampilkan semua karakter yang dapat menampilkan bilangan ganjil dari 1 sampai < n (n diinputkan), kecuali bilangan ganjil kelipatan 3.

Flowchart :



Input :

```
#include <stdio.h>
int main (){
    int n, i;
    printf("masukkan batas n = ");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {
        if (i % 2 != 0)
        {
            if (i % 3 == 0) continue;
            printf("%d ", i);
        }
    }
}
```

Output :

```
masukkan banyaknya n = 14
1 5 7 11 13
```

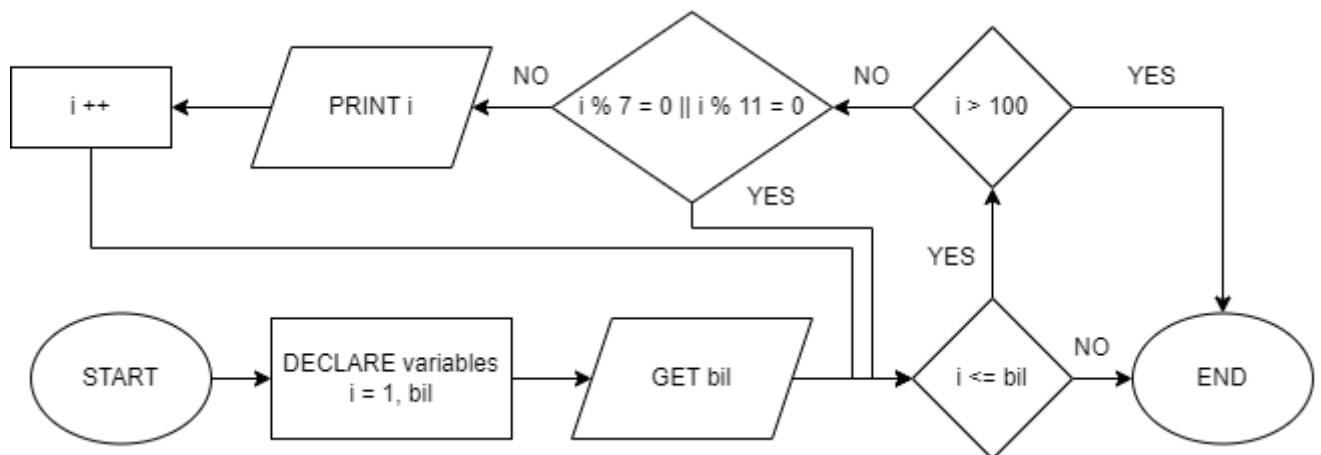
Analisa :

Program C ini bertujuan untuk memungkinkan pengguna untuk menginputkan angka `n`, setelah itu mencetak semua angka ganjil dalam rentang dari 0 hingga `n-1` yang bukan merupakan kelipatan dari 3. Ini dilakukan dengan menggunakan loop `for` yang menghitung dan mencetak angka-angka tersebut. Dengan kata lain, program ini memberi pengguna kemampuan untuk melihat angka-angka ganjil yang sesuai dengan aturan tersebut berdasarkan masukan yang mereka berikan.

Program

3. Dengan menggunakan pernyataan break dan continue, buatlah program untuk membuat program dengan input n, dan output, bilangan ganjil kecuali kelipatan 7 dan 11 mulai dari 1 sampai < n atau bilangan tersebut < 100

Flowchart :



Input :

```
#include <stdio.h>
int main(){
    int bil, i;
    printf("masukkan bilangan batas = ");
    scanf("%d", &bil);
    for ( i = 1; i <= bil; i++)
    {
        if (i > 100) break;
        if (i % 7 == 0 || i % 11 == 0) continue;
        if (i % 30 == 0)
        {
            printf("\n");
        }
        printf("%d ", i);
    }
}
```

```
}  
}
```

Output :

```
masukkan bilangan batas = 50  
1 2 3 4 5 6 8 9 10 12 13 15 16 17 18 19 20 23 24 25 26 27 29  
30 31 32 34 36 37 38 39 40 41 43 45 46 47 48 50
```

Analisa :

Kode ini adalah program dalam bahasa C yang meminta pengguna untuk memasukkan sebuah bilangan bulat `bil`. Setelah menerima input tersebut, program akan mencetak angka-angka dari 1 hingga `bil`, dengan beberapa aturan tertentu. Pertama, jika angka melebihi 100, program akan berhenti mencetak. Selanjutnya, jika angka tersebut adalah kelipatan dari 7 atau 11, program akan mengabaikannya dan melanjutkan ke angka berikutnya. Jika angka tersebut adalah kelipatan dari 30, program akan mencetak angka tersebut di baris baru. Hasil akhirnya adalah daftar angka yang dicetak dengan mematuhi aturan-aturan tersebut sesuai dengan input yang diberikan oleh pengguna..

4. Buatlah program untuk menerima daftar nilai mahasiswa sbb :

Input : - Jumlah data (n)

- Nilai ke-1 s/d Nilai ke-n

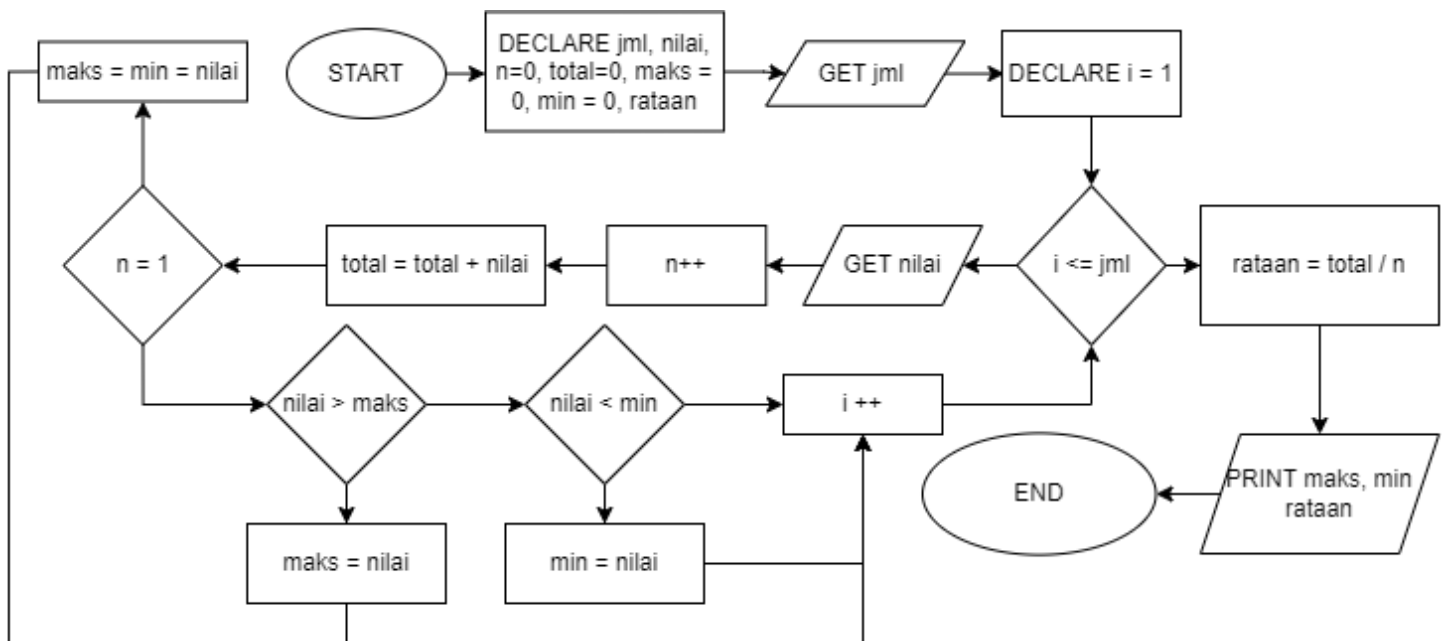
Output : - Nilai minimal

- Nilai maksimal

- Nilai rata-rata (rata-rata = nilai total / jumlah data)

(Petunjuk : Gunakan loop for dan seleksi kondisi dengan if)

Flowchart :



Input :

```

#include<stdio.h>
int main (){
    int jml, nilai, n=0, total=0, maks=0, min= 0, rataan;
    printf("masukkan nilai n = ");
    scanf("%d", &jml);
    for (int i = 1; i <= jml; i++)
    {
        printf("Masukkan nilai %d ", i);
        scanf("%d", &nilai);
        n++;
        total += nilai;
        if (n == 1)
        {
            maks=min=nilai;
        }else {
            if (nilai > maks)
            {
                maks = nilai;
            }
            if (nilai < min)
            {
                min = nilai;
            }
        }
    }

    rataan = total / n;
    printf("Nilai maksimal adalah %d \n", maks);
    printf("Nilai minimal adalah %d \n", min);
    printf("Nilai rata rata adalah %d", rataan);
}
  
```

Output :

```

masukkan nilai n = 4
Masukkan nilai 1 97
Masukkan nilai 2 87
Masukkan nilai 3 88
Masukkan nilai 4 98
Nilai maksimal adalah 98
Nilai minimal adalah 87
Nilai rata rata adalah 92

```

Analisa :

Program C dalam kode ini meminta pengguna untuk memasukkan jumlah nilai ("jml"). Kemudian, program mengambil nilai-nilai tersebut dan melakukan perhitungan statistik sederhana. Program menggunakan variabel "n" untuk menghitung jumlah nilai yang dimasukkan, "total" untuk mengumpulkan jumlah total dari semua nilai, "maks" dan "min" untuk menyimpan nilai maksimum dan minimum, dan "rataan" untuk menghitung nilai rata-rata. Saat nilai pertama dimasukkan, program menginisialisasi "maks" dan "min" dengan nilai tersebut. Program ini digunakan untuk melakukan analisis statistik sederhana terhadap sejumlah nilai yang dimasukkan oleh pengguna.

5. Dengan menggunakan pernyataan nested loop , buatlah program sebagai berikut :

Input : n

Output :

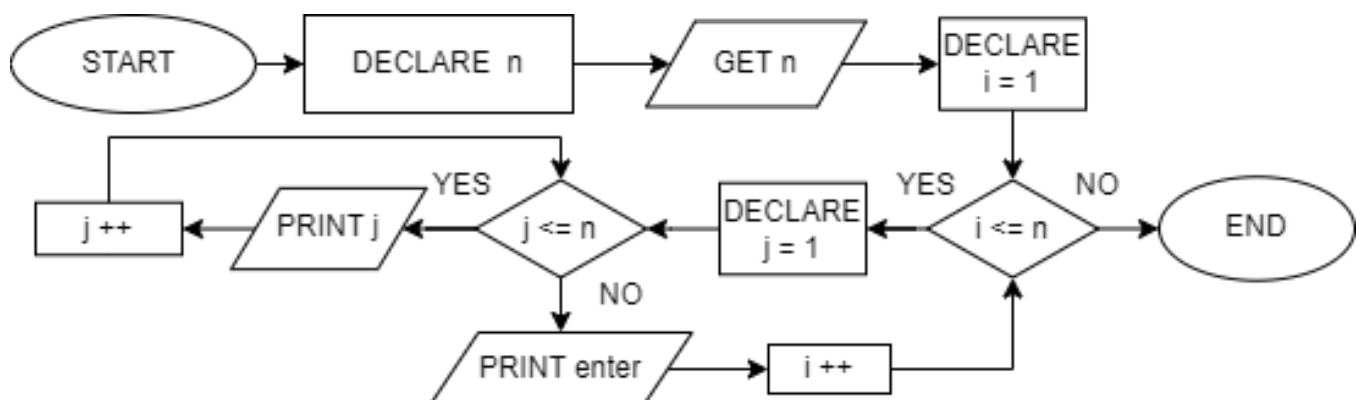
```

1 2 3 4 5 ... n
1 2 3 4 5 ... n
1 2 3 4 5 ... n
.....
1 2 3 4 5 ... n

```

} n kali

Flowchart :



Input :

```

#include <stdio.h>
int main(){
    int n;
    printf("Masukkan n = ");
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)

```

```

{
    for (int j = 1; j <= n; j++)
    {
        printf("%d ", j);
    }
    printf("\n");
}

```

Output :

```

Masukkan n = 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8

```

Analisa :

Program bahasa C ini mengharapkan input bilangan bulat positif "n" dari pengguna.

Kemudian, program ini mencetak pola segitiga angka dari 1 hingga n, dengan setiap baris berisi angka dari 1 hingga n, dan setiap baris diakhiri dengan baris baru. Ini dilakukan dengan dua loop bersarang, dengan loop luar mengontrol baris-baris pola dan loop dalam mencetak angka dalam setiap baris. Hasilnya adalah pola segitiga angka dengan n baris, dengan setiap baris berisi urutan angka dari 1 hingga n.

6. Dengan menggunakan pernyataan nested loop , buatlah program sebagai berikut :

Input : n

Output :

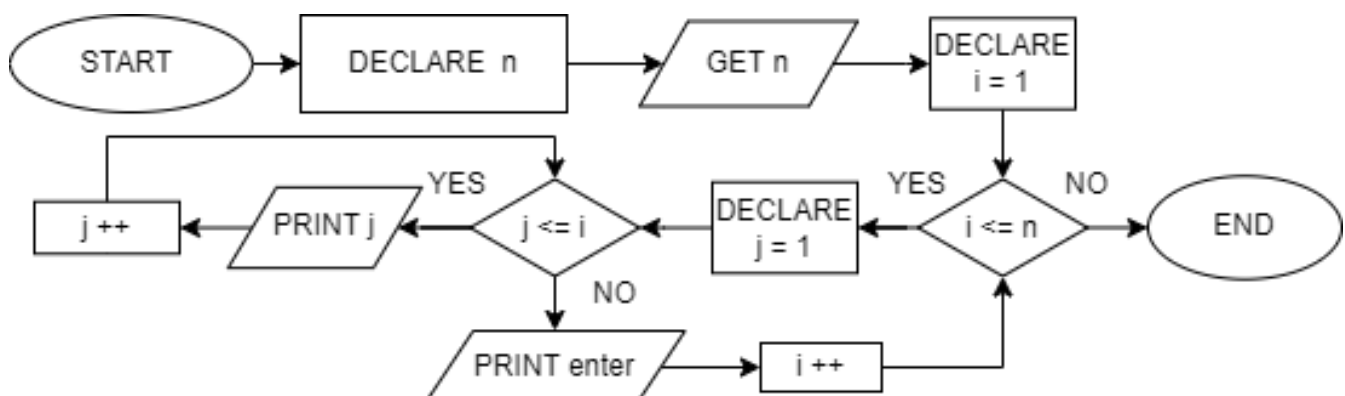
```

1
2 2
3 3 3
.....
n n n n n ... n

```

} n kali

Flowchart :



Input :

```
#include <stdio.h>
```



```

int main(){
    int n;
    printf("masukkan n = ");
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            printf("%d", i);
        }
        printf("\n");
    }
}

```

Output :

```

masukkan n = 8
1
22
333
4444
55555
666666
7777777
88888888

```

Analisa :

Program C dalam kode tersebut meminta pengguna untuk memasukkan bilangan bulat "n" dan kemudian mencetak pola segitiga dengan angka yang berulang dari 1 hingga "n". Ada dua loop bersarang dalam program ini, loop luar mengatur baris-baris pola (dari 1 hingga "n"), dan loop dalam mengatur pencetakan angka dalam setiap baris. Selama pencetakan, program mencetak angka yang sesuai dengan nilai "i", atau baris saat ini, sebanyak Hasilnya adalah pola segitiga dengan angka yang berulang di setiap baris sesuai dengan nomor barisnya; contohnya, baris pertama berisi "1", baris kedua berisi "22", baris ketiga berisi "333", dan seterusnya sesuai dengan input pengguna.

7. Pada akhir setiap 4 buah program diatas tambahkan tanya “apakah anda ingin keluar (y/t)?”, pertanyaan tersebut hanya bisa di jawab dengan huruf ‘y’ (y kecil) dan ‘t’(t kecil). Dan akan keluar dari program setelah dijawab dengan ‘y’ (y kecil)

7.3 Input :

```

#include <stdio.h>
int main()
{
    int bil, i;
    char jwb;
    while (1)
    {
        printf("masukkan bilangan batas = ");
        scanf("%d", &bil);
    }
}

```

```

for (i = 1; i <= bil; i++)
{
    if (i > 100)
        break;
    if (i % 7 == 0 || i % 11 == 0)
        continue;
    if (i % 30 == 0)
    {
        printf("\n");
    }
    printf("%d ", i);

}
fflush(stdin);
printf("\napakah anda ingin keluar ? ");
scanf("%c", &jwb);
if (jwb == 'y' ) break;
if(jwb == 't') continue;
}
}

```

Output :

```

masukkan bilangan batas = 77
1 2 3 4 5 6 8 9 10 12 13 15 16 17 18 19 20 23 24 25 26 27 29
30 31 32 34 36 37 38 39 40 41 43 45 46 47 48 50 51 52 53 54 57 58 59
60 61 62 64 65 67 68 69 71 72 73 74 75 76
apakah anda ingin keluar ? t
masukkan bilangan batas = 40
1 2 3 4 5 6 8 9 10 12 13 15 16 17 18 19 20 23 24 25 26 27 29
30 31 32 34 36 37 38 39 40
apakah anda ingin keluar ? y

```

7.4 Input :

```

#include <stdio.h>
int main()
{
    char jwb;
    while (1)
    {
        int jml, rataa;
        int nilai, n = 0, total = 0, maks = 0, min = 0;
        printf("masukkan nilai n = ");
        scanf("%d", &jml);
        fflush(stdin);
        for (int i = 1; i <= jml; i++)
        {

            printf("Masukkan nilai %d ", i);
            scanf("%d", &nilai);
            n++;
            total += nilai;
            if (n == 1)
            {
                maks = min = nilai;
            }
        }
    }
}

```

```

    }
    else
    {
        if (nilai > maks)
        {
            maks = nilai;
        }
        if (nilai < min)
        {
            min = nilai;
        }
    }
    fflush(stdin);
}
rataan = total / n;
printf("Nilai maksimal adalah %d \n", maks);
printf("Nilai minimal adalah %d \n", min);
printf("Nilai rata rata adalah %d\n\n", rataan);

printf("apakah anda ingin keluar ? ");
scanf(" %c", &jwb);
if (jwb == 'y' ) break;
if(jwb == 't') continue;
}
}

```

Output :

```

Masukkan nilai 3 43
Nilai maksimal adalah 43
Nilai minimal adalah 23
Nilai rata rata adalah 33

apakah anda ingin keluar ? t
masukkan nilai n = 3
Masukkan nilai 1 5
Masukkan nilai 2 6
Masukkan nilai 3 7
Nilai maksimal adalah 7
Nilai minimal adalah 5
Nilai rata rata adalah 6

apakah anda ingin keluar ? y

```

7.5 Input :

```

#include <stdio.h>
int main()
{
    int n;
    char jwb;
    while (1)
    {
        /* code */
    }
}

```

```

printf("Masukkan n = ");
scanf("%d", &n);
for (int i = 1; i <= n; i++)
{
    for (int j = 1; j <= n; j++)
    {
        printf("%d ", j);
    }
    printf("\n");
}
printf("anda ingin keluar ");
scanf(" %c", &jwb);
if (jwb == 'y' ) break;
if(jwb == 't') continue;
}
}

```

Output :

```

Masukkan n = 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
anda ingin keluar t
Masukkan n = 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
anda ingin keluar y

```

7.6 Input :

```

#include <stdio.h>
int main()
{
    int n;
    char jwb;
    while (1)
    {
        printf("masukkan n = ");
        scanf("%d", &n);
        for (int i = 1; i <= n; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                printf("%d", i);
            }
            printf("\n");
        }
        printf("anda ingin keluar? ");
    }
}

```

```
scanf(" %c", &jwb);  
if (jwb == 'y' ) break;  
if(jwb == 't') continue;  
}  
}
```

Output :

```
masukkan n = 6  
1  
22  
333  
4444  
55555  
666666  
anda ingin keluar? t  
masukkan n = 5  
1  
22  
333  
4444  
55555  
anda ingin keluar? y
```

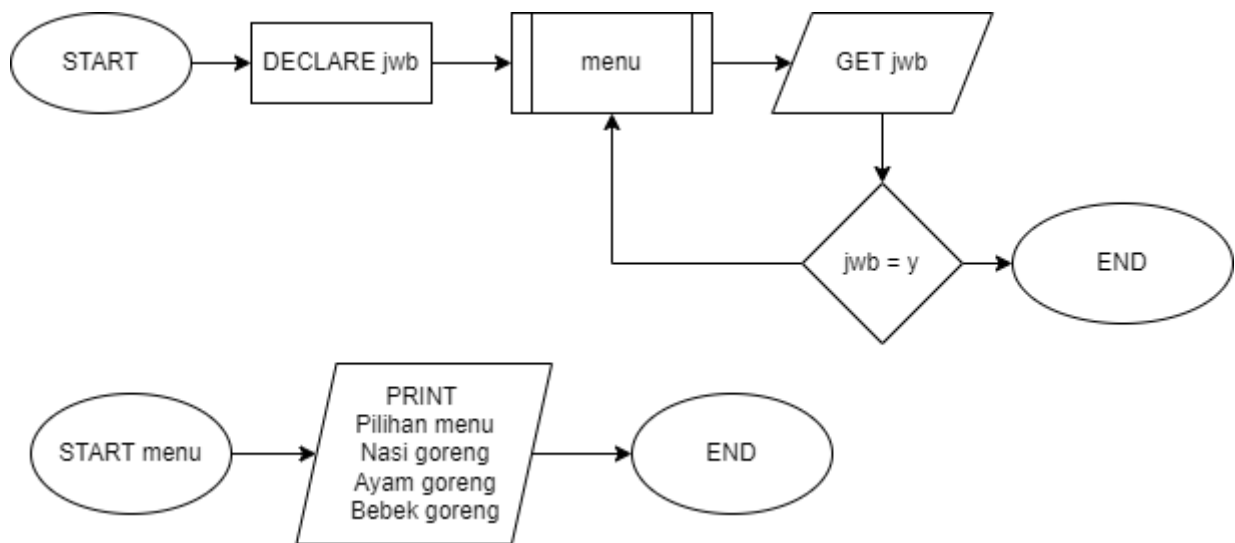
[Source Code Praktikum 11](#)

## PRAKTIKUM 13 :

### Percobaan :

1. a. Buatlah sebuah fungsi yang berfungsi untuk menampilkan sebuah string (di layar) = "Pilihan Menu" (misalkan nama fungsinya = menu). Fungsi tersebut tidak memiliki nilai kembalian (return value) dan juga tidak menerima parameter masukan apapun.
- b. Tulislah prototipe fungsi untuk fungsi tersebut.
- c. Buat function main untuk memanggil function menu() secara berulang-ulang, dengan jumlah perulangan yang merupakan input dari user.

Flowchart :



Input :

```
#include<stdio.h>
void menu();//prototype
int main(){
    char jwb;
    do
    {
        menu();
        printf( "apakah anda ingin mengulang lagi? ");
        scanf(" %c", &jwb);
    } while (jwb == 'y');
}
//
void menu(){
    printf("Pilihan menu :\n");
    printf("Nasi goreng\n");
    printf("ayam goreng\n");
    printf("bebek goreng\n");
}
```

Output :

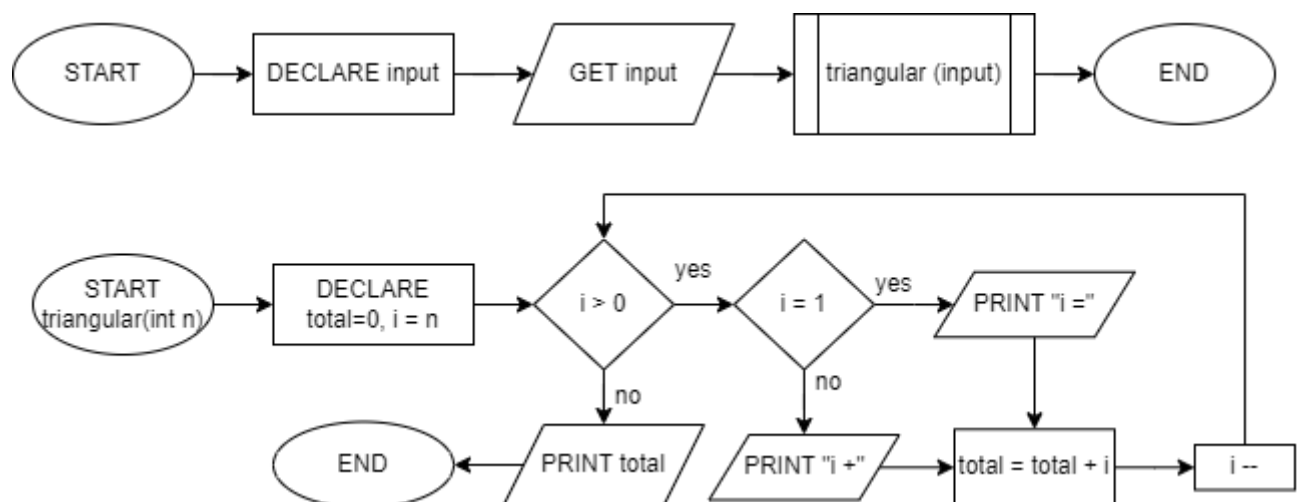
```
Pilihan menu :
Nasi goreng
ayam goreng
bebek goreng
apakah anda ingin mengulang lagi? y
Pilihan menu :
Nasi goreng
ayam goreng
bebek goreng
apakah anda ingin mengulang lagi? t
```

Analisa :

Kode ini adalah program sederhana dalam bahasa C yang menggunakan fungsi menu() untuk mencetak daftar menu makanan. Fungsi ini dianggil oleh program utama dalam loop do-while, yang meminta pengguna untuk memilih menu makanan. Setelah mencetak menu, program menanyakan kepada pengguna apakah mereka ingin mengulangi proses memilih makanan dengan pertanyaan "Apakah Anda ingin mengulang lagi?" dan akan berlanjut selama pengguna menjawab "ya". Kode ini digunakan untuk membuat program menu sederhana yang memungkinkan pengguna memilih makanan dan mengulangi proses pemilihan kapan pun mereka mau.

2. a. Buatlah sebuah fungsi untuk menghitung jumlah triangular n (misal nama fungsinya = triangular). Fungsi tersebut memiliki sebuah parameter berupa bilangan int (n) yang akan dicari triangularnya serta tidak memiliki nilai kembalian (return value)
- b. Tulislah prototipe fungsi untuk fungsi tersebut.
- c. Buat function main untuk memanggil function triangular() tersebut dengan nilai n yang merupakan input dari user.

Flowchart :



Input :

```
#include <stdio.h>
```

```

void triangular(int n);
int main()
{
    int input;
    printf("masukkan nilai n = ");
    scanf("%d", &input);
    triangular(input);
}
void triangular(int n)
{
    int total=0;
    for (int i = n; i > 0; i--)
    {
        if (i == 1)
        {
            printf("%d = ", i);
        }
        else
        {
            printf("%d + ", i);
        }
        total += i;
    }
    printf("total = %d", total);
}

```

Output :

```

masukkan nilai n = 7
7 + 6 + 5 + 4 + 3 + 2 + 1 = 28

```

Analisa :

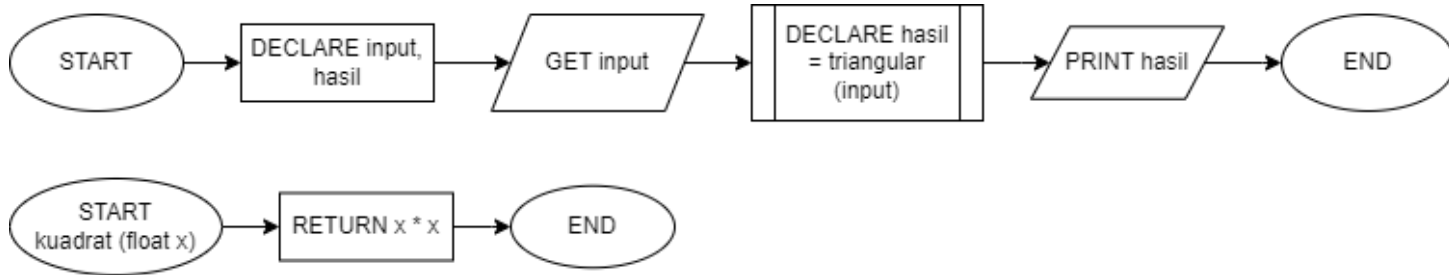
Program bahasa C ini meminta pengguna untuk memasukkan bilangan bulat positif n. Program memanggil fungsi triangular(input) dengan nilai n sebagai argumen setelah menerima input tersebut. Fungsi triangular(int n) menghitung total dari serangkaian angka yang dimulai dengan nilai n dan mencetak barisan angka tersebut dalam format yang memasukkan penjumlahan. Program mencetak angka dari n hingga 1, dengan tanda + di akhir setiap angka kecuali angka terakhir yang diakhiri dengan =. Kemudian, jumlah total angka dicetak setelahnya. Hasil akhir adalah barisan angka dan total penjumlahannya yang diwakili sesuai dengan nilai n yang dimasukkan oleh pengguna.

3. a. Buatlah sebuah fungsi untuk menghitung nilai bilangan kuadrat (misal nama fungsinya = kuadrat). Fungsi tersebut memiliki sebuah parameter bertipe float, yaitu bilangan yang akan dikuadratkan serta memiliki sebuah return value bertipe float, yaitu hasil kuadratnya
- b. Tulislah prototipe fungsi untuk fungsi tersebut.
- c. Buat function main untuk memanggil function kuadrat() tersebut dengan



bilangan x yang akan dicari kuadratnya merupakan input dari user.

Flowchart :



Input :

```
#include<stdio.h>
float kuadrat(float x);

int main(){
    float input, hasil;
    printf("Masukkan bilangan = ");
    scanf("%f", &input);
    hasil = kuadrat(input);
    printf("hasil = %.2f", hasil );
}

float kuadrat(float x){
    return x * x;
}
```

Output :

```
Masukkan bilangan = 3
hasil = 9.00
```

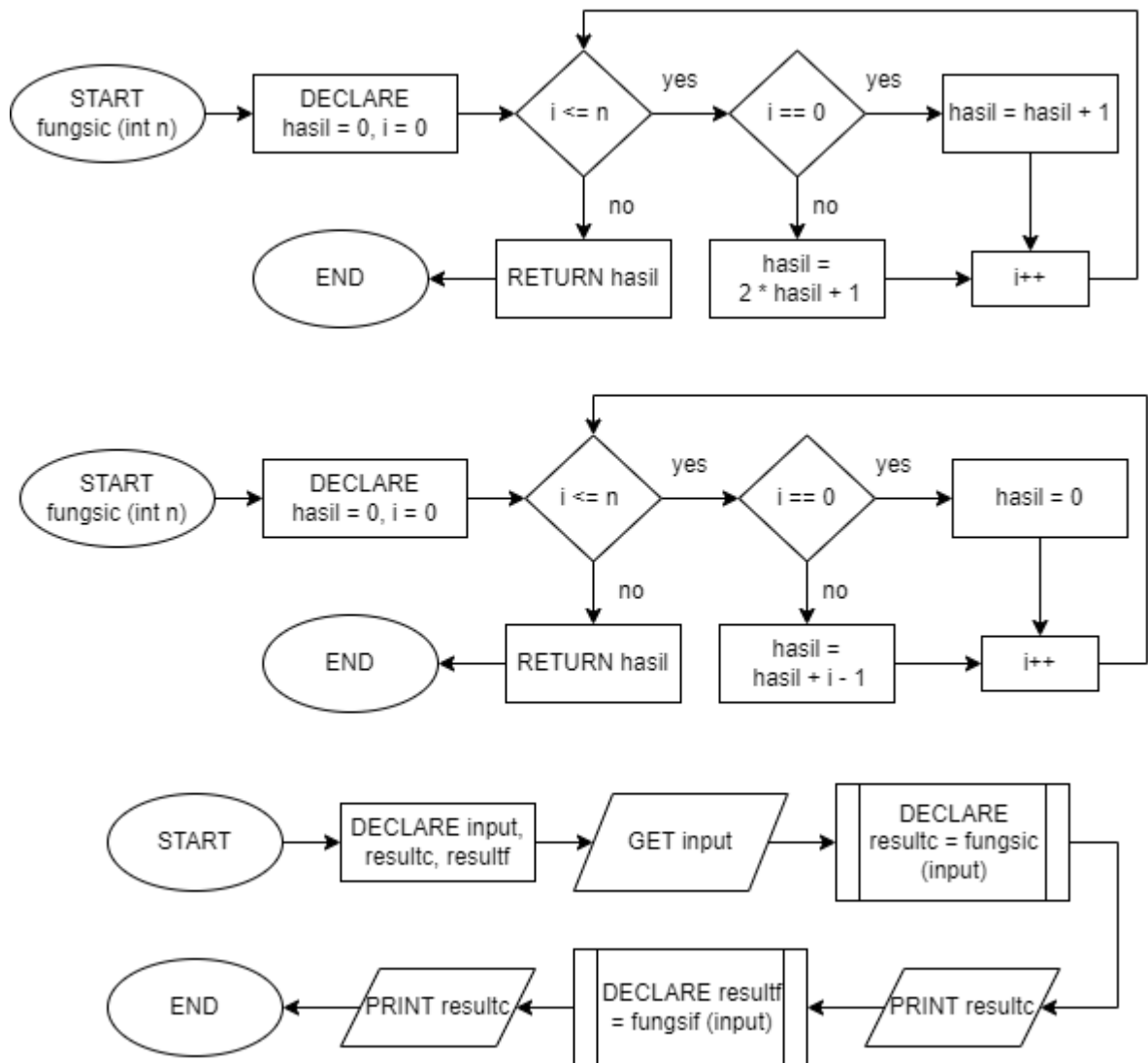
Analisa :

Kode ini adalah program dalam bahasa C yang mengharapkan pengguna untuk memasukkan bilangan desimal (float) dan kemudian menghitung kuadrat dari bilangan tersebut menggunakan fungsi kuadrat(float x). Pengguna diminta untuk memasukkan bilangan, dan kemudian bilangan tersebut diteruskan ke fungsi `kuadrat` untuk dihitung kuadratnya. Hasil perhitungan tersebut kemudian dicetak dalam format dua angka desimal menggunakan printf. Program ini sangat sederhana, dan fokus pada penggunaan fungsi untuk melakukan perhitungan matematis sederhana.

4. Dengan menggunakan fungsi, buatlah program untuk mendapatkan keluaran dari fungsi di bawah ini, dimana inputnya berupa bilangan untuk n

- $C_n = 2 C_{n-1} + 1$  jika  $C_0 = 1$
- $S_n = S_{n-1} + n - 1$  jika  $S_1 = 0$

Flowchart :



Input :

```

#include <stdio.h>
int fungsic(int n);
int fungsis(int n);
int main()
{
    int input, resultc, resultf;
    printf("masukkan n = ");
    scanf("%d", &input);
    resultc = fungsic(input);
    printf("Hasil dari Cn = 2Cn-1 + 1 dari n = %d adalah %d\n", input, resultc);
    resultf = fungsis(input);
    printf("Hasil dari Sn = Sn-1 + n - 1 dari n = %d adalah %d\n", input, resultf);
}
int fungsic(int n)
{
    int hasil = 0;
    for (int i = 0; i <= n; i++)
    {
        if (i == 0)
        {
            hasil += 1;
        }
        else
        {

```

```

        hasil = 2 * hasil + 1;
    }
}
return hasil;
}
int fungsic(int n)
{
    int hasil = 0;
    for (int i = 1; i <= n; i++)
    {
        if (i == 1)
        {
            hasil = 0;
        }
        else
        {
            hasil = hasil + i - 1;
        }
    }
    return hasil;
}
}

```

Output :

```

masukkan n = 4
Hasil dari  $C_n = 2C_{n-1} + 1$  dari  $n = 4$  adalah 31
Hasil dari  $S_n = S_{n-1} + n - 1$  dari  $n = 4$  adalah 6

```

Analisa :

Kode ini adalah program dalam bahasa C yang menghitung dua deret aritmatika berbeda, yaitu  $C_n = 2C_{n-1} + 1$  dan  $S_n = S_{n-1} + n - 1$ , berdasarkan bilangan bulat positif  $n$  yang dimasukkan oleh pengguna. Program memiliki dua fungsi, fungsic untuk menghitung deret pertama dan fungsic untuk menghitung deret kedua. Dalam fungsi fungsic, program menggunakan loop for untuk menghitung nilai deret pertama dengan memulai dari 1 hingga  $n$ , mengikuti rumus rekursif. Sementara dalam fungsi fungsic, program menghitung deret kedua menggunakan loop for yang dimulai dari 1 hingga  $n$  dengan rumus yang sesuai. Hasil dari kedua deret tersebut kemudian dicetak dengan pesan yang menjelaskan hasil perhitungan sesuai dengan input pengguna. Terdapat juga komentar yang memberikan alternatif implementasi menggunakan rekursi untuk menghitung deret pertama.

[Source Code Praktikum 13](#)