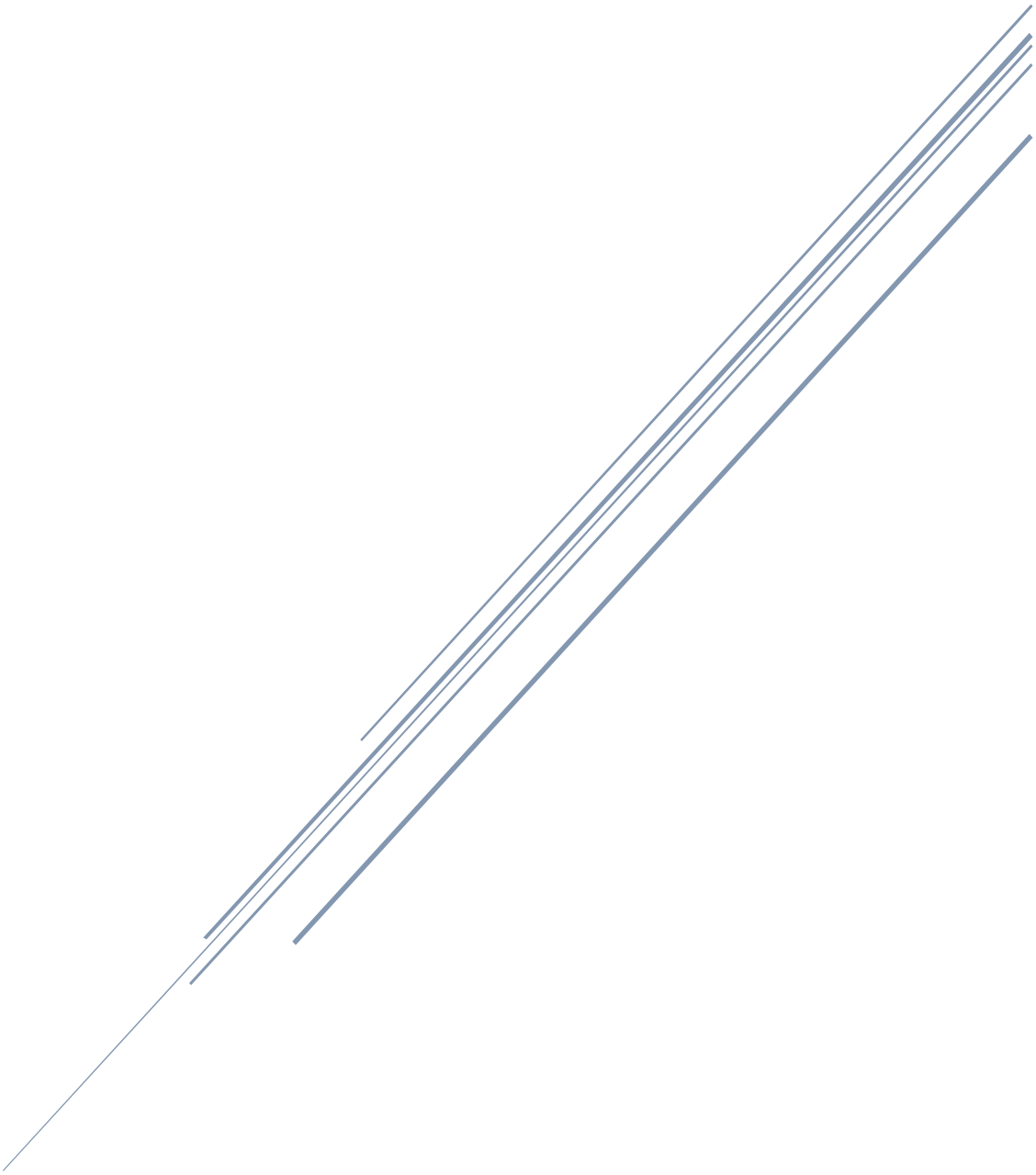
FLOWCHART 2

MODUL #2



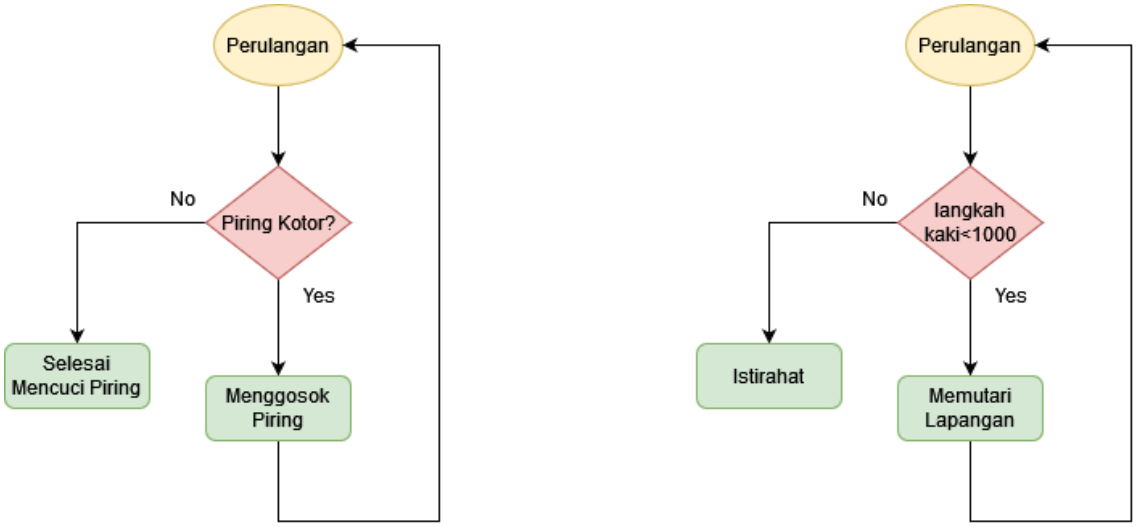
Asisten Dasar Pemrograman 2021/2022

TUJUAN

1. Memahami cara kerja penggunaan flowchart lebih mendalam
2. Memahami penggunakan konsep perulangan dalam flowchart
3. Mengimplementasikan kasus pada dunia tanya kedalam flowchart

Pengantar

Dalam kegiatan sehari hari, kita sebagai manusia sering melakukan kegiatan atau rutinitas yang sama berkali-kali yang bisa disebut juga dengan “mengulang” . Seperti contoh ketika kita sedang melakukan aktivitas mencuci piring, kita akan terus menggosok atau menyikat piring tersebut hingga bersih. Adapun contoh yang lain, ketika kita berlari memutari lapangan sekolah, kita akan terus berlari hingga target kalori yang terbakar atau jumlah langkah kaki sudah terpenuhi. Jika kedua kasus nyata tersebut diimplementasikan dalam sebuah diagram alir atau bagan arus (flowchart) akan berbentuk sebagai berikut:



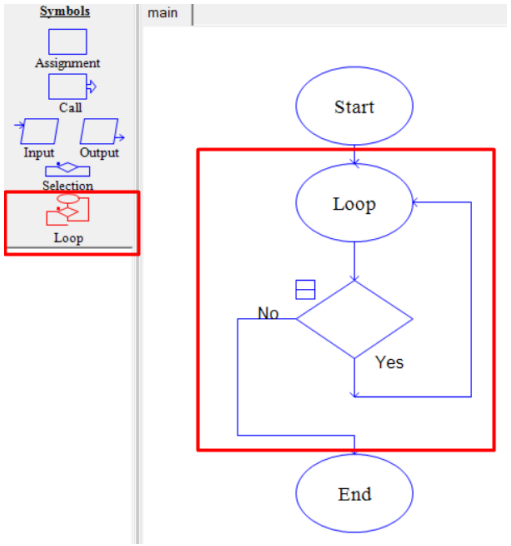
Pengertian Perulangan

Perulangan merupakan sebuah aksi yang dilakukan secara berulang-ulang atau terus – menerus hingga kondisi atau tujuan yang ditentukan telah tercapai. Dalam pemrograman perulangan sangat membantu karena tidak perlu menulis code / algoritma yang sama secara terus menerus, sehingga penulisan code menjadi lebih efisien.

Perulangan dalam flowchart

1. Symbol

Gambar disamping merupakan *symbol* dari perulangan yang bersifat *default* (belum diisi dengan **kondisi** atau **aksi**).

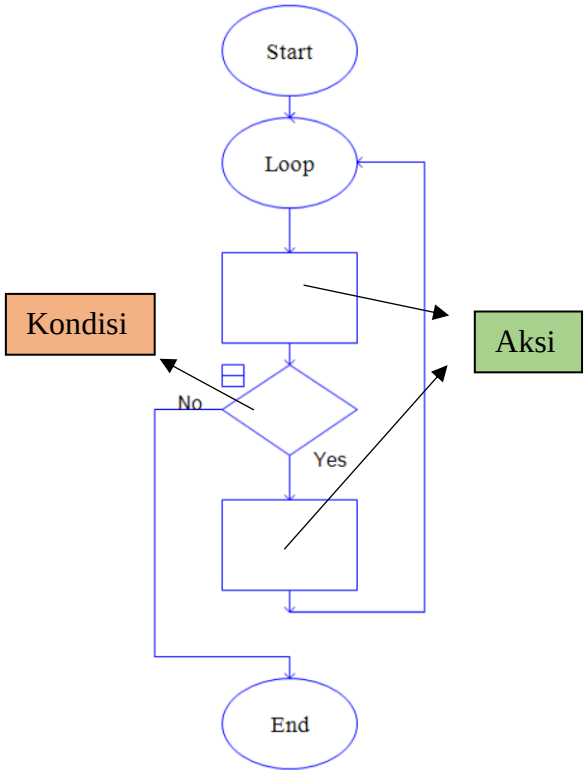


Symbol Perulangan Pada Flowchart

2. Komponen

Kondisi : Melakukan pengecekan jika bernilai *true*/**Yes** maka aksi akan dijalankan, sedangkan jika bernilai *false*/**No** maka perulangan akan berhenti

Aksi : Merupakan proses/kegiatan yang akan terus dijalankan selama **kondisi** perulangan bernilai *true*/**Yes**. **Aksi** dapat terletak pada sebelum/sesudah pengkodisian yang akan dijelaskan lebih detail pada **4. Jenis perulangan**



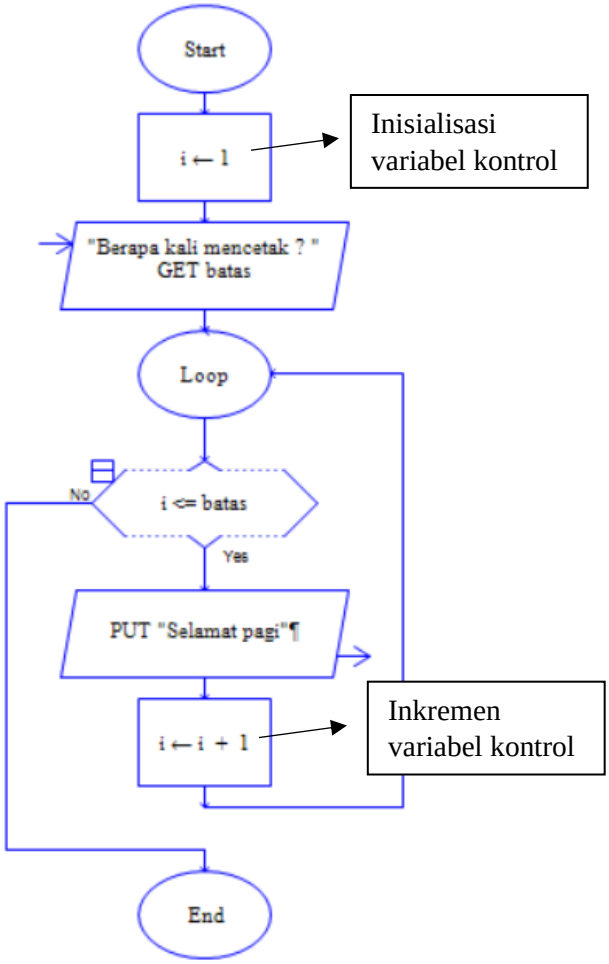
3. Struktur

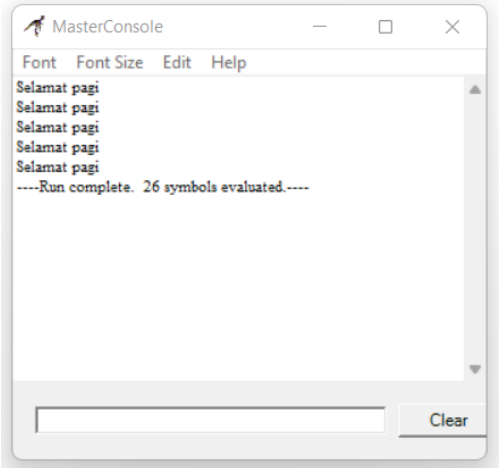
Variabel kontrol :
Merupakan variabel yang mengontrol jalannya perulangan, dimana setiap putaran perulangan variabel kontrol akan melakukan pengkodisian apakah perulangan bernilai *true* / *false*. Variabel kontrol memiliki *value* awal sebagai mulainya sebuah perulangan.

Inisialisasi :
Pengisian nilai / *value* awal pada suatu variable.

Inkremen = Penambahan jumlah **variabel kontrol** sesuai dengan yang ditentukan dalam setiap putaran perulangan.
Contoh :
i = i + 1, memilliki arti bahwa setiap putaran perulangan, variabel i akan bertambah 1.

Dekremen = Pengurangan jumlah **variabel kontrol** sesuai dengan yang ditentukan dalam setiap putaran perulangan.
Contoh :
i = i - 1, memilliki arti bahwa setiap putaran perulangan variabel i akan berkurang 1.



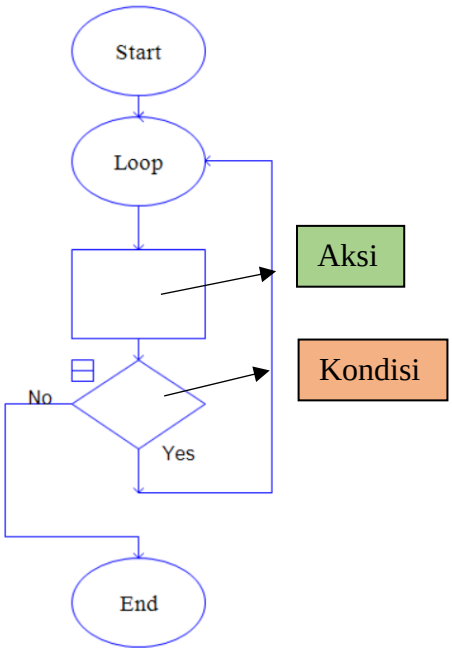


Mencetak “Selamat pagi” sebanyak 5 kali

4. Jenis Perulangan

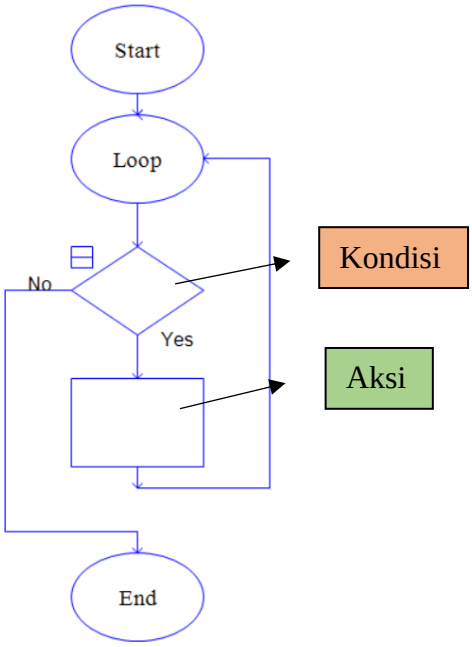
Do – While (Aksi – Kondisi) / Post - test

Merupakan jenis perulangan dimana **proses / aksi** akan dijalankan terlebih dahulu selanjutnya dilakukan proses **pengkodisian** apakah bernilai true/false.



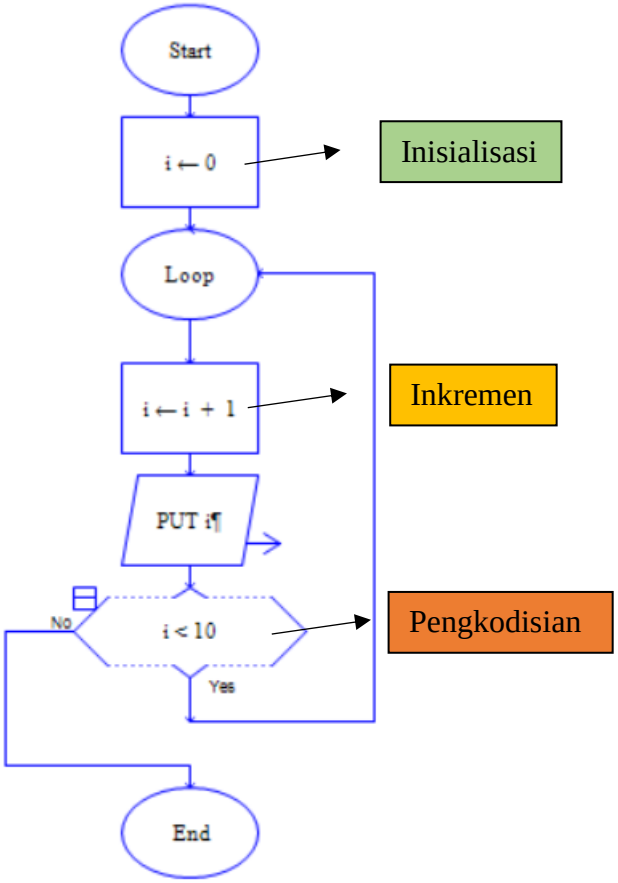
While – Do (Kondisi – Aksi) / Pretest

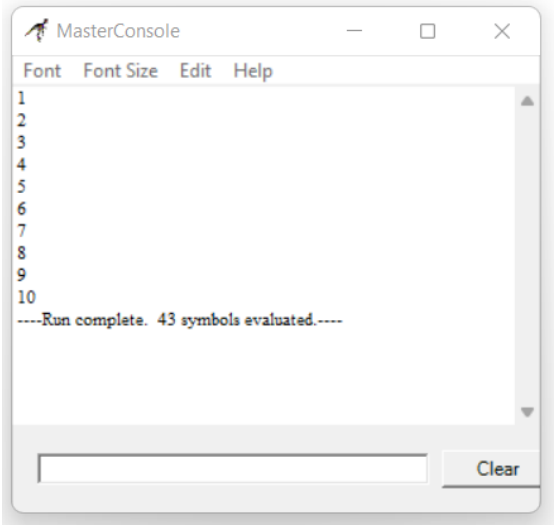
Merupakan jenis perulangan dimana akan dilakukan **pengkodisian** terlebih dahulu apakah bernilai **true / false**, jika true maka selanjutnya akan dilakukan aksi/proses.



For

Merupakan jenis perulangan dimana sudah mengetahui jumlah perulangan yang akan dilakukan. Perulangan for wajib menggunakan variabel kontrol serta meng inisialisasi variabel kontrol tersebut, lalu menentukan jenis dari perulangan (inkremen / dekremen) dan yang terakhir memberi pengkodisian yang menyatakan kapan perulangan akan berakhir.

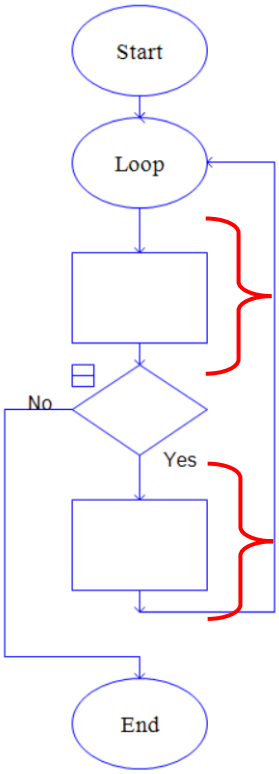




Output (Menampilkan bilangan 1 - 10)

Tambahan :

Hindari perulangan yang bersifat **unstructured loop**, dimana terdapat **dua aksi** yaitu sebelum dan sesudah pengkodisian, gunakan **do while / while – do.**

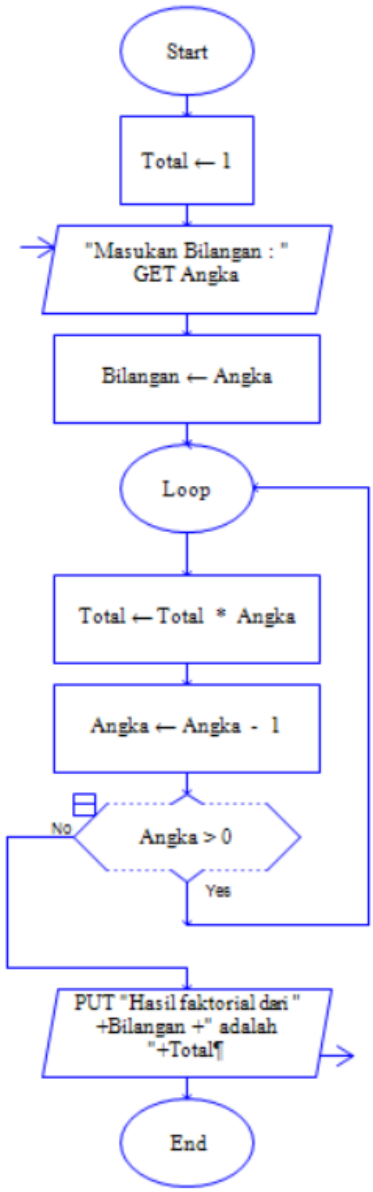


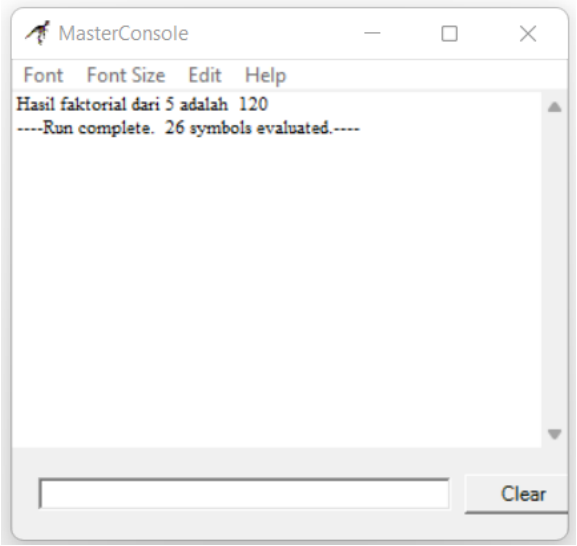
2 Aksi, sebelum dah sesudah pengkodisian

Guided

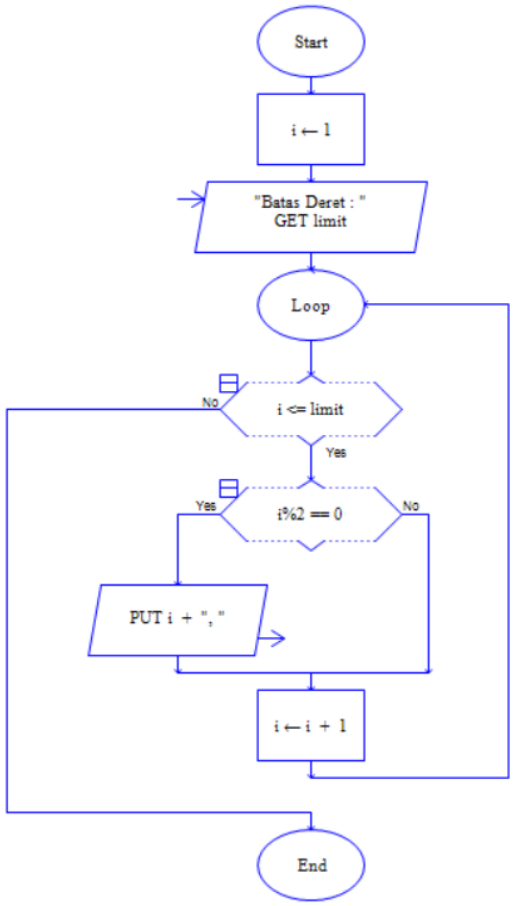
Setelah praktikkan memahami modul flowhart 2, praktikkan diharapkan dapat memplajari guided dibawah, supaya tidak mengalami kesulitan saat mengerjakkan unguided nanti. Penjelasan guided terdapat pada akhir halaman, jangan lupa perhatikan ketentuan pengerjaan dan format pengumpulan, jika ada yang ingin ditanyakan silahkan menghubungi asisten. Terima kasih!.

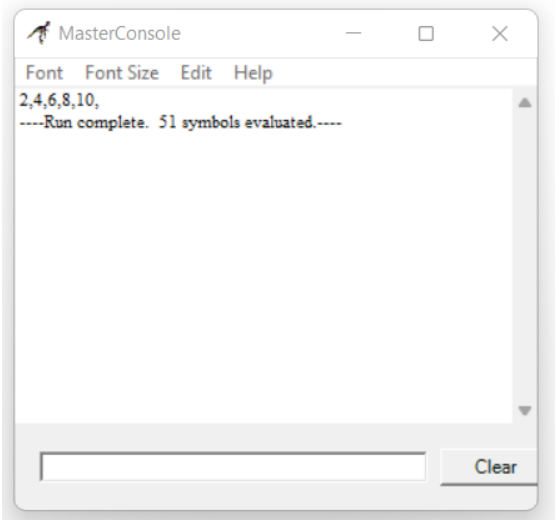
1. Membuat program dimana dapat menghitung faktorial dari angka yang diinputkan, contoh 5! Maka 5 x 4 x 3 x 2 x 1 = 120.





2. Membuat program dimana menampilkan billangan genap dari 1 sampai dengan sesuai inputan *user*.





Ketentuan Pengerjaan :

Format penamaan guided :

- **GD2_X_Y_ZZZZZ**.rap

Kedua file .rap diletakkan dalam 1 folder lalu di zip / *archive* dengan format nama :

- **GD2 _Y_ZZZZZ**.zip

Keterangan format penamaan :

X = nomor soal

Y = kelas

Z = 5 digit NPM terakhir

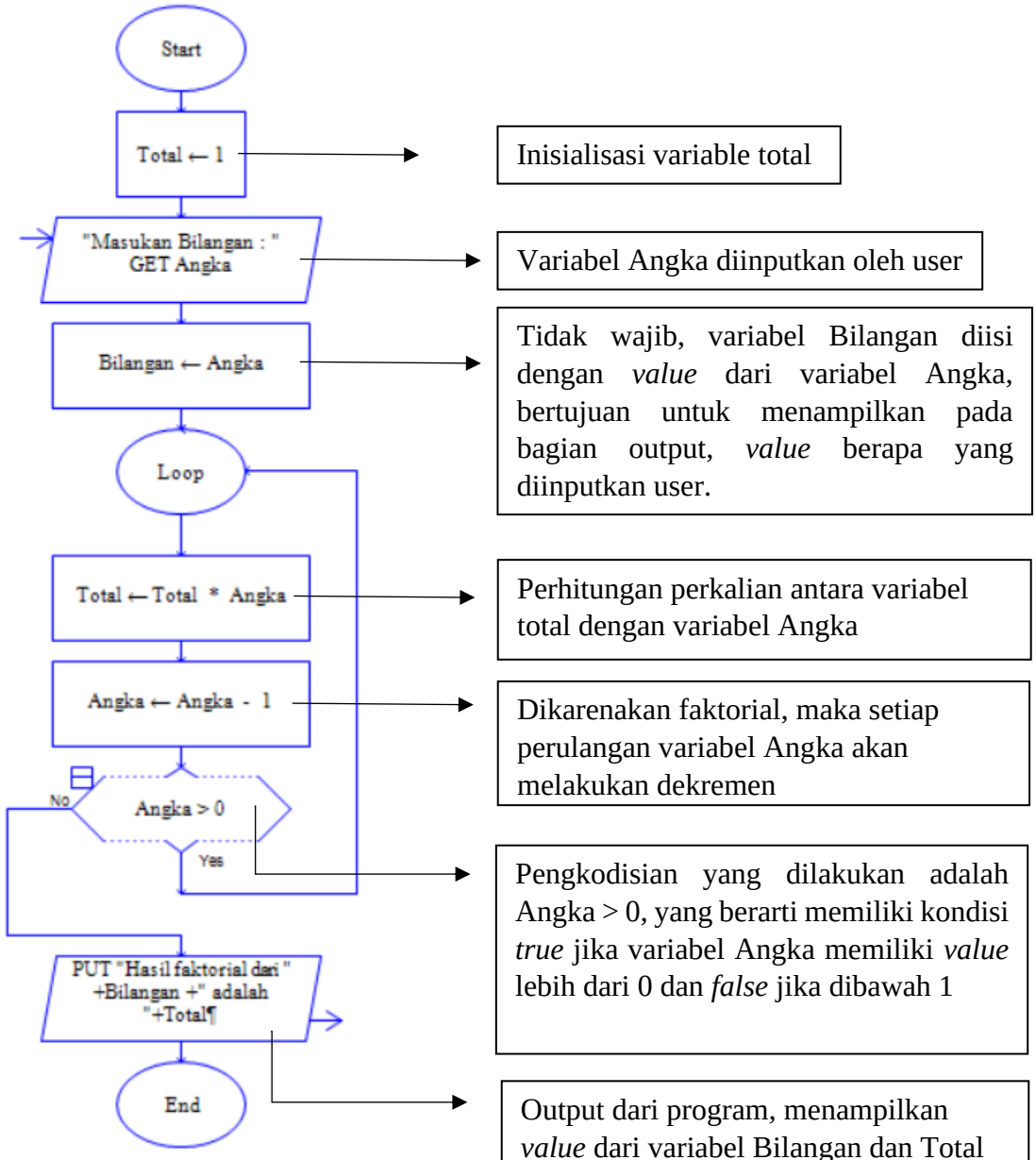
Contoh format penamaan :

Guided 1 = GD2_1_B_10670

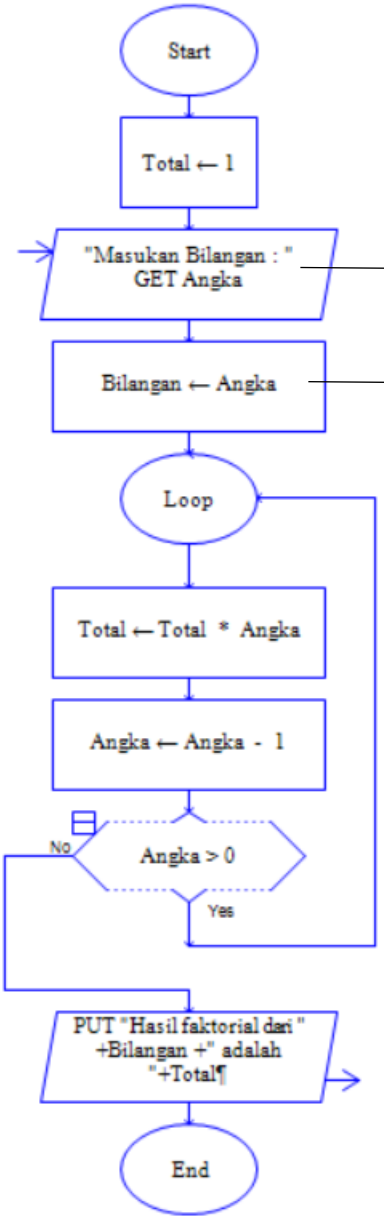
Zip / archive = GD2_B_10670

Pembahasan Guided

1. Pada soal nomor 1 diminta unuk membuat faktorial dari angka yang diinputkan, bisa disimpulkan bahwa angka yang diinputkan adalah variabel kontrol / variabel pengendali yang akan menentukan putaran perulangan lanjut atau berhenti. Mengapa ? karena perulangan akan melakukan aksi sebanyak jumlah angka yang diinputkan, semisal 4! Maka aksi akan dilakukan sebanyak 4 kali (4 x 3 x 2 x 1). Maka pada pengkodisian diberikan angka > 0 yang berarti angka akan berhenti jika memiliki *value* 0, karena faktorial hanya mencapai *value* 1 saja. Varibel total digunakan untuk menampung jumlah dari hasil perkalian faktorial, karena sifatnya perkalian, maka inisialisasi dari variabel total diisi dengan value 1.



Proses Jalannya Algoritma



User menginputkan *value* 4

Karena *value* dari variabel Angka adalah 4, maka variabel Bilangan memiliki *value* 4

Proses perulangan

***Value* awal :**

Angka = 4
Total = 1
Bilangan = 4

Aksi ke - 1 , Total * Angka = 4 → Angka – 1 = 3

Aksi ke – 2 , Total * Angka =12 → Angka – 1 = 2

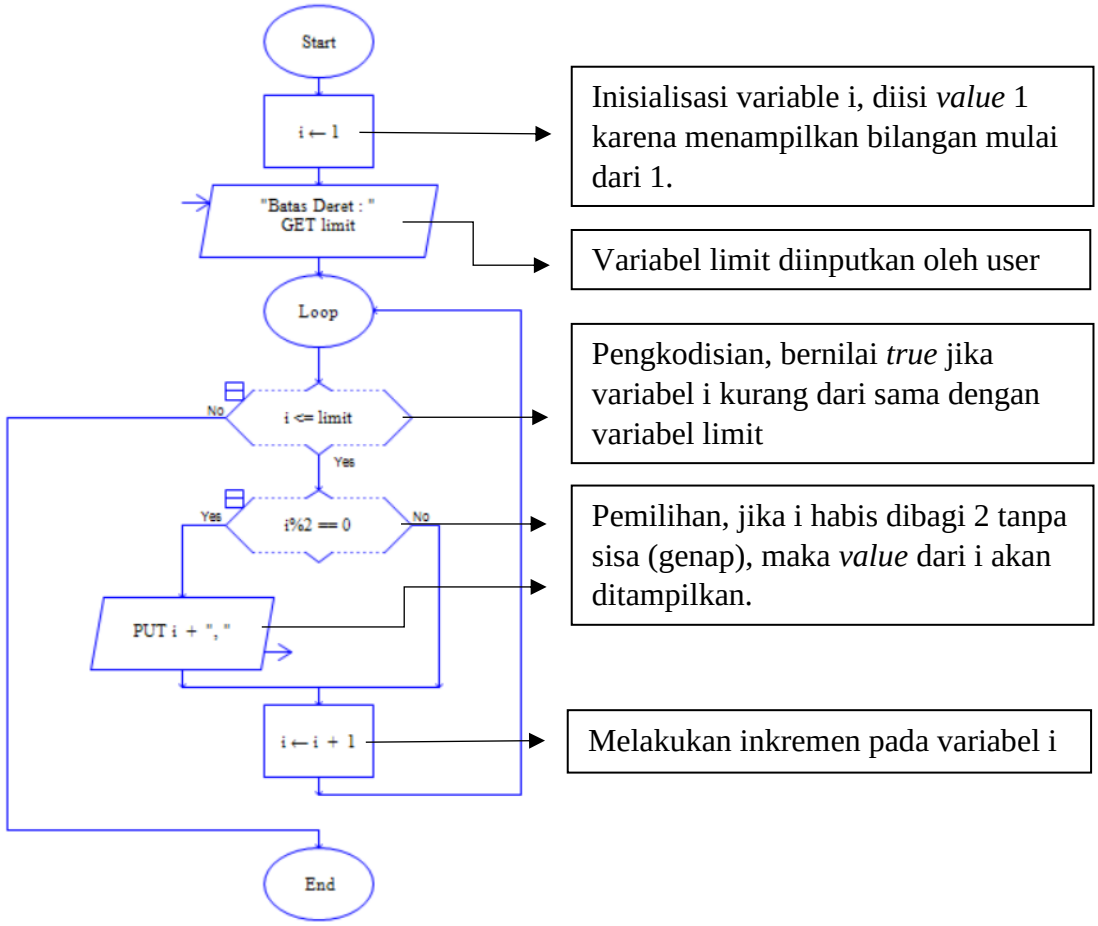
Aksi ke – 3 , Total * Angka =24 → Angka – 1 = 1

Aksi ke – 4 , Total * Angka =24 → Angka – 1 = 0

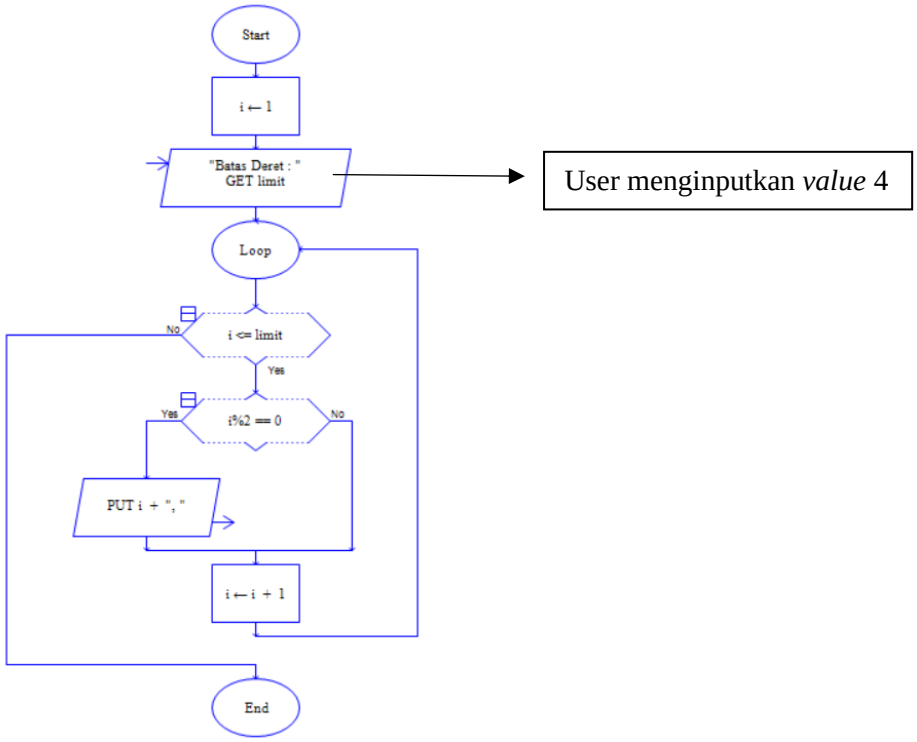
Aksi ke – 5

Value dari variabel Angka 0 maka pengkodisian bernilai *false* dan keluar dari perulangan , sehingga hasil faktorial dari 4 adalah 24.

2. Soal nomor 2 diminta untuk menampilkan bilangan genap mulai dari 1 hingga bilangan yang diminta oleh *user*. Variabel kontrol dalam program ini adalah i yang memiliki 2 fungsi yaitu menampilkan bilangan dan sebagai perbandingan pengkodisian. Pengkodisian ditentukan berdasarkan apakah i kurang dari sama dengan limit, limit sendiri berdasarkan inputan user yang merupakan batas bilangan yang ingin ditampillkan. Karena yang ditampilkan hanya bilangan genap, maka terdapat kondisi pemillihan, dimana jika i%2==0 atau jika i habis dibagi 2 tanpa sisa , maka i merupakan bilangan genap dan selanjutnya menampilkan value dari i. Pada akhir aksi akan dilakukan inkremen pada variabel i.



Proses Jalannya Algoritma



Diketahui:
limit = 4

Aksi ke – 1
i = 1
1 % 2 == 0, *false* maka tidak menampilkan *value*
i + 1 = 2

Aksi ke – 2
i = 2
2 % 2 == 0, *true* maka menampilakan *value*
i + 1 = 3

Aksi ke – 3
i = 3
3 % 2 == 0, *false* maka tidak menampilkan *value*
i + 1 = 4

Aksi ke – 4
i = 4
4 % 2 == 0, *true* maka menampilkan *value*
i + 4 = 5

Sebelum lanjut ke **Aksi ke – 5** pengkodisian sudah menunjukkan *false* karena i > limit, maka output dari program diatas adalah 2,4.