

---

Jobsheet Bahasa Pemrograman

## #03

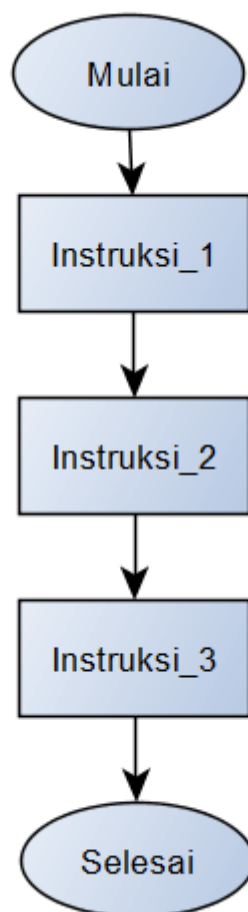
## Struktur Dasar Algoritma

Struktur Dasar Algoritma terdiri dari **Algoritma Sekuensial/Runtunan (Sequence Algorithm)**, **Algoritma Percabangan/Pemilihan (Conditional Algorithm)**, dan **Algoritma Perulangan (Looping Algorithm)**. Dimana **struktur** adalah cara sesuatu disusun atau dibangun dengan pola tertentu.

### Algoritma Sekuensial/Runtunan (Sequence Algorithm)

**Algoritma Sekuensial/Runtunan (Sequence Algorithm)** adalah algoritma yang langkah-langkahnya dikerjakan atau dieksekusi secara **urut** dari awal hingga akhir sesuai dengan urutannya.

Flowchart struktur algoritma sekuensial



Kita bisa lihat contoh gambar flowchart di atas, terlihat kumpulan instruksi tersusun dari atas ke bawah. Program akan mengeksekusi instruksi\_1, setelah instruksi\_1 sudah selesai

---

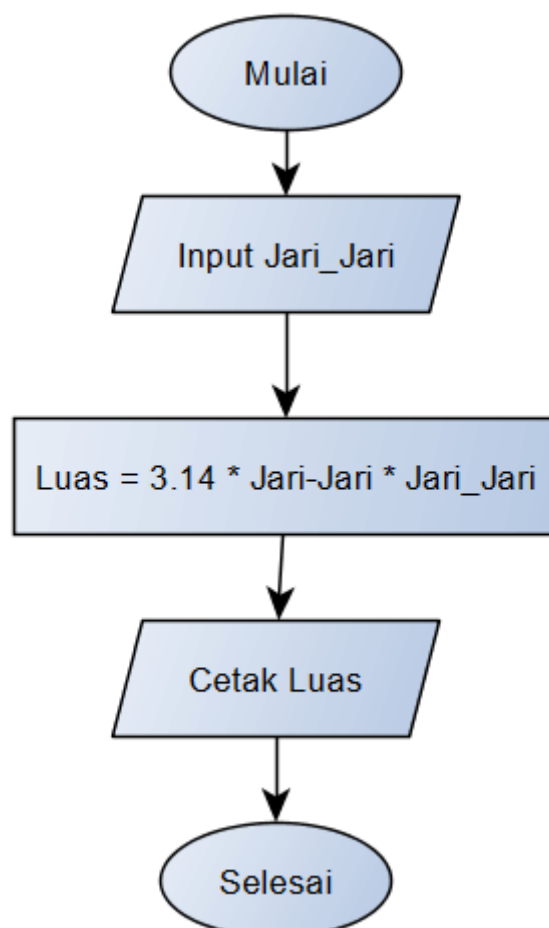
dikerjakan selanjutnya program akan mengeksekusi instruksi\_2. Setelah instruksi\_2 sudah selesai dikerjakan selanjutnya program akan mengeksekusi instruksi\_3. Setelah instruksi\_3 sudah selesai dikerjakan maka selanjutnya program akan berakhir atau selesai.

Kita ambil contoh program Luas Lingkaran, dimana untuk panjang jari-jarinya sudah diketahui besarnya. Dan berikut adalah algoritmanya :

**Deskripsi :**

1. Mulai
2. Input Jari\_Jari
3. Hitung Luas =  $3.14 * \text{Jari\_Jari} * \text{Jari\_Jari}$
4. Cetak Luas
5. Selesai

**Flowchart Luas Lingkaran**



Perhatikan gambar, flowchart Luas Lingkaran memiliki 3 instruksi yang harus dieksekusi.

Instruksi ke-1 : program membaca nilai jari-jari yang telah diinputkan.

Instruksi ke-2 : program melakukan operasi aritmatika rumus luas lingkaran.

Instruksi ke-3 : menampilkan hasil dari perhitungan rumus luas lingkaran.

**Penting** untuk diingat bahwa Semakin **banyak instruksi** pada program maka semakin **banyak waktu** untuk mengeksekusinya. Pada perancangan sebuah program, **efesiensi instruksi** sangatlah **penting** untuk mempercepat proses dan menghemat konsumsi memory pada program.

### ✓ Latihan Soal

1. Jelaskan konsep dari struktur Algoritma Sekuensial/Runtunan.
2. Buat algoritma untuk perhitungan luas kubus dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).
3. Buat contoh Masalah yang memiliki algoritma penyelesaian dalam bentuk struktur Algoritma Sekuensial/Runtunan, dan buat algoritmanya dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).

## ✚ Algoritma Percabangan/Pemilihan (Conditional Algorithm)

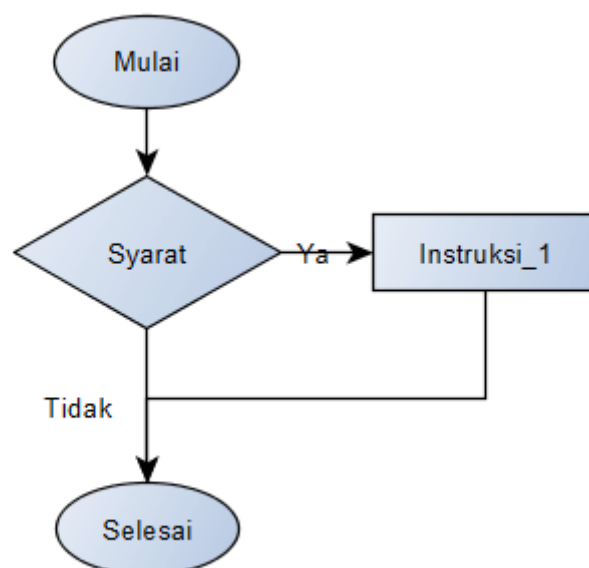
**Algoritma Percabangan/Pemilihan (Conditional Algorithm)** atau dapat disebut juga **Algoritma bersyarat** adalah algoritma yang menjalankan instruksi selanjutnya **apabila syarat** yang ditetapkan sudah **terpenuhi**. Pada struktur ini tidak setiap instruksi akan dikerjakan, instruksi yang dikerjakan hanya **yang memenuhi syarat saja**. Pada bahasa pemrograman struktur ini sering digunakan menggunakan instruksi IF-THEN atau lebih dikenal instruksi jika-maka.

## Macam Bentuk Struktur Algoritma Percabangan/Pemilihan

### A. Struktur IF Sederhana

Bentuk dari struktur IF sederhana adalah IF (Syarat) THEN (Instruksi), simak gambar dibawah ini :

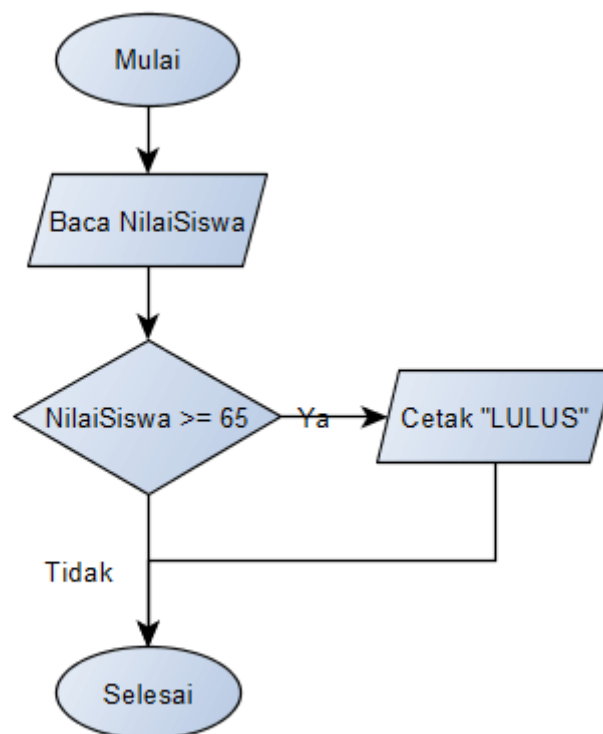
Bentuk flowchart IF Sederhana



Pada gambar flowchart di atas, struktur IF sederhana dapat kita lihat bahwa instruksi\_1 akan dieksekusi jika syarat yang ditentukan bernilai benar dan jika bernilai salah, instruksi\_1 tidak akan dieksekusi. Untuk lebih jelasnya kita simak contoh algoritma kelulusan siswa, program ini akan mengatakan bahwa siswa tersebut lulus jika nilainya  $\geq 65$ .

**Deskripsi :**

1. Mulai
2. Baca nilaisiswa
3. Jika nilaisiswa  $\geq 65$  maka kerjakan langkah 4
4. Cetak "LULUS"
5. Selesai

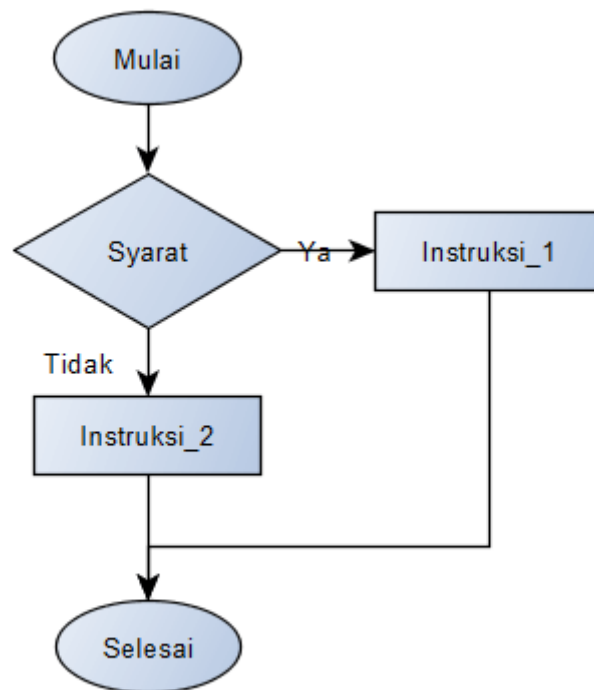
**Flowchart Kelulusan Siswa**

Pada flowchart di atas, nilai siswa yang diinputkan akan diuji apakah nilai tersebut lebih besar atau sama dengan 65. Jika benar maka program akan mencetak "LULUS" kemudian selesai, namun jika salah maka program selesai.

---

**B. Struktur IF...THEN...ELSE...**

Pada struktur ini, terdapat dua kemungkinan instruksi yang akan dikerjakan berdasarkan hasil dari pengujian. Contoh jika syarat yang diujikan memperoleh hasil benar maka instruksi\_1 dikerjakan, namun jika bernilai salah maka instruksi\_2 yang dikerjakan. Untuk lebih jelasnya bisa lihat gambar dibawah ini :

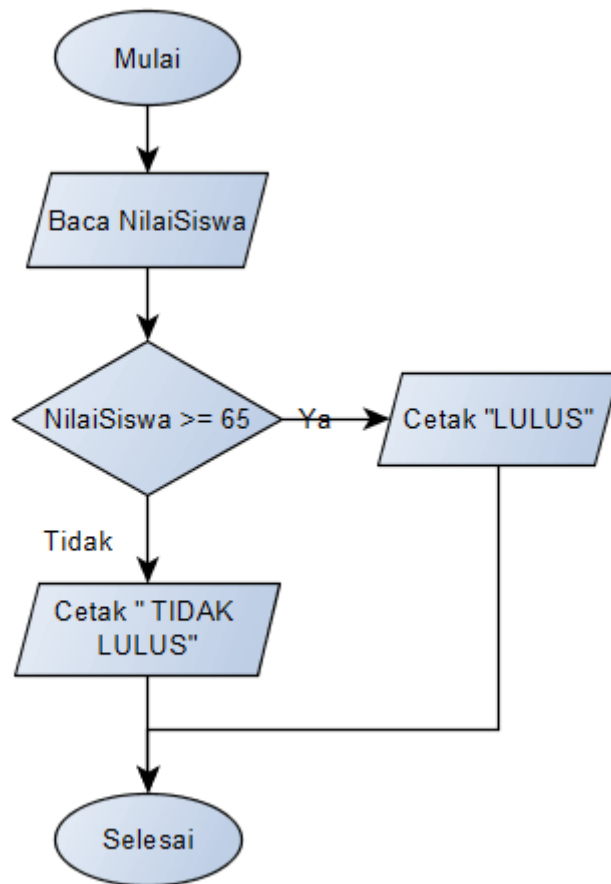
**Bentuk flowchart**

Tidak lengkap rasanya kalau tidak diberikan contoh, mengambil masalah sebelumnya tentang kelulusan siswa. Sekarang kita modifikasi sedikit dimana jika nilai siswa lebih besar atau sama dengan 65 maka dinyatakan lulus dan jika nilai siswa lebih kecil dari 65 maka siswa dinyatakan tidak lulus.

**Deskripsi :**

1. Mulai
2. Baca nilaisiswa
3. Jika nilaisiswa  $\geq 65$  maka kerjakan langkah 4, selain itu kerjakan langkah 5
4. Cetak "LULUS"
5. Cetak "TIDAK LULUS"
6. Selesai

### Flowchart Kelulusan Siswa

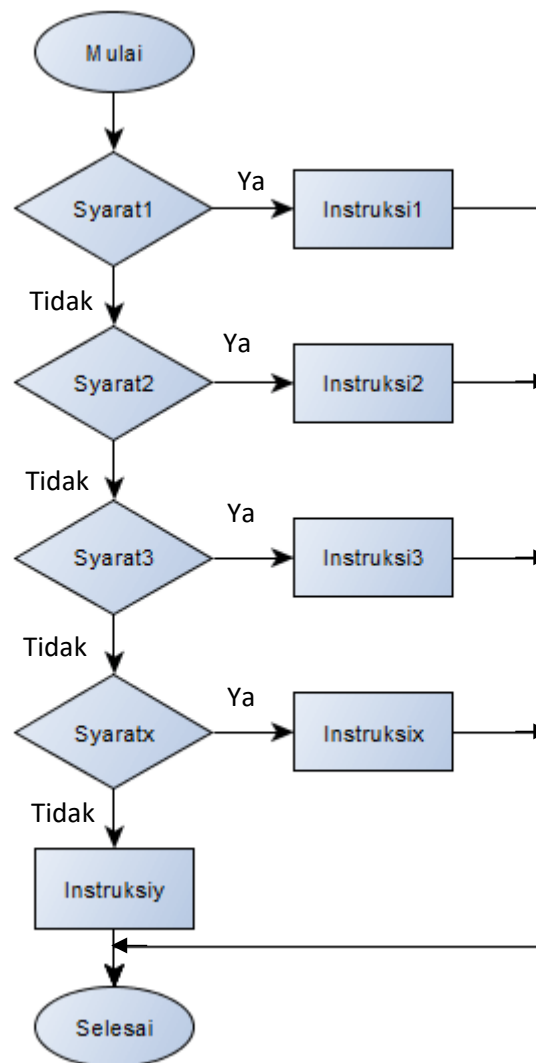


Pada gambar flowchart di atas, setelah nilai diinputkan maka nilai tersebut akan diuji apakah nilai siswa lebih besar atau sama dengan 65. Jika benar maka program akan mencetak "LULUS" lalu selesai, Jika tidak maka akan dicetak "TIDAK LULUS" lalu program selesai.

### C. IF Bersarang

Untuk struktur yang satu ini kita perlu belajar logika dan ketelitian, satu alasan yang pasti adalah struktur ini sering dipakai untuk tes kerja dalam bidang IT terutama pekerjaan yang berkaitan dengan perancangan sistem. Pada struktur ini juga kemungkinan akan banyak instruksi yang dikerjakan berdasarkan hasil pengujian, bisa disimak gambar dibawah ini:

---

**Flowchart struktur IF Bersarang**


Dari flowchart di atas kita bisa pelajari bagaimana struktur dari IF bersarang. Kita jabarkan satu persatu, jika syarat1 yang diuji bernilai benar maka instruksi1 akan dieksekusi, jika syarat1 bernilai salah maka syarat2 diuji, jika syarat2 yang diuji bernilai benar maka instruksi2 akan dieksekusi, jika syarat2 bernilai salah maka syarat3 diuji, jika syarat3 yang diuji bernilai benar maka instruksi3 akan dieksekusi, jika syarat3 bernilai salah maka syaratx diuji, jika syaratx yang diuji bernilai benar maka instruksix akan dieksekusi, jika syaratx bernilai salah atau tidak ada syarat yang terpenuhi maka instruksiy yang dieksekusi.

Untuk lebih jelasnya, kita implementasikan dalam sebuah algoritma konversi nilai siswa, program akan mengkonversi nilai siswa dari semula dalam bentuk angka dirubah menjadi bentuk huruf dengan ketentuan sebagai berikut :

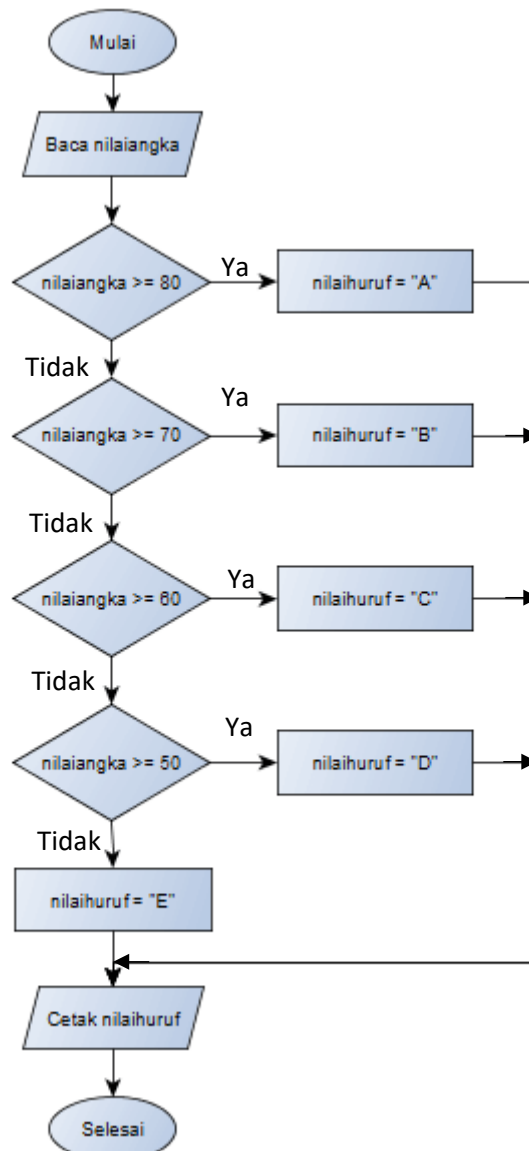
- Jika nilai angka  $\geq 80$  maka nilai huruf sama dengan A
- Jika nilai angka  $\geq 70$  maka nilai huruf sama dengan B
- Jika nilai angka  $\geq 60$  maka nilai huruf sama dengan C
- Jika nilai angka  $\geq 50$  maka nilai huruf sama dengan D

---

– Jika nilai angka < 50 maka nilai huruf sama dengan E

**Deskripsi :**

1. Mulai
2. Baca nilaiangka
3. Jika nilaiangka  $\geq 80$  maka nilaihuruf = "A", selain itu
4. Jika nilaiangka  $\geq 70$  maka nilaihuruf = "B", selain itu
5. Jika nilaiangka  $\geq 60$  maka nilaihuruf = "C", selain itu
6. Jika nilaiangka  $\geq 50$  maka nilaihuruf = "D", selain itu
7. nilaihuruf = "E"
8. Cetak nilaihuruf
9. Selesai

**Flowchart Konversi Nilai Siswa**



---

Dapat dicermati gambar flowchart di atas, hal yang perlu diperhatikan dalam struktur IF bersarang adalah jika salah satu syarat sudah terpenuhi maka syarat-syarat lainnya yang ada sejajar tidak akan diuji lagi. Contohnya kita masukan nilaiangka 60 maka nilaihurufnya adalah "C" dan pengujian nilaiangka lebih besar atau sama dengan 50 tidak akan dijalankan.

### ✓ Latihan Soal

1. Jelaskan konsep dari struktur Algoritma Percabangan/Pemilihan dan macam bentuk struktur yang ada di dalamnya.
2. Buat algoritma program yang menampilkan mana bilangan terbesar dan mana bilangan terkecil dari 2 buah bilangan yang diinputkan oleh user. Buatlah dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).
3. Buat algoritma program yang dapat mendeteksi bilangan negatif atau bilangan positif dari bilangan yang diinputkan oleh user. Buatlah dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).
4. Buat contoh Masalah yang memiliki algoritma penyelesaian dalam bentuk struktur Algoritma Percabangan/Pemilihan, dan buat algoritmanya dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).

## Algoritma Perulangan (Looping Algorithm)

**Algoritma Perulangan (Looping Algorithm)** adalah sebuah struktur dasar algoritma yang menjalankan beberapa langkah tertentu secara **berulang-ulang sampai** terpenuhinya **suatu kondisi**.

Pada kehidupan sehari-hari banyak yang kita lakukan secara berulang-ulang, contohnya seperti algoritma menjemur pakaian:

1. Siapkan tiang Jemuran
2. Ambil satu pakaian yang sudah dicuci
3. Peras pakaian hingga sedikit mengering
4. Letakan pakaian pada tiang jemuran
5. Ulangi langkah 2 sampai 4 hingga pakaian habis.

Dari algoritma menjemur pakaian di atas, kita dapat mendapat gambaran tentang struktur algoritma looping. Jenis struktur dasar ini akan selesai jika sebuah kondisi tertentu terpenuhi, seperti menjemur pakaian maka proses jemur akan selesai jika pakaian yang akan dijemur sudah habis. Struktur dasar algoritma ini mempunyai beberapa bentuk seperti Struktur FOR, Struktur While dan Struktur Do....While.

## Macam Bentuk Struktur Algoritma Perulangan

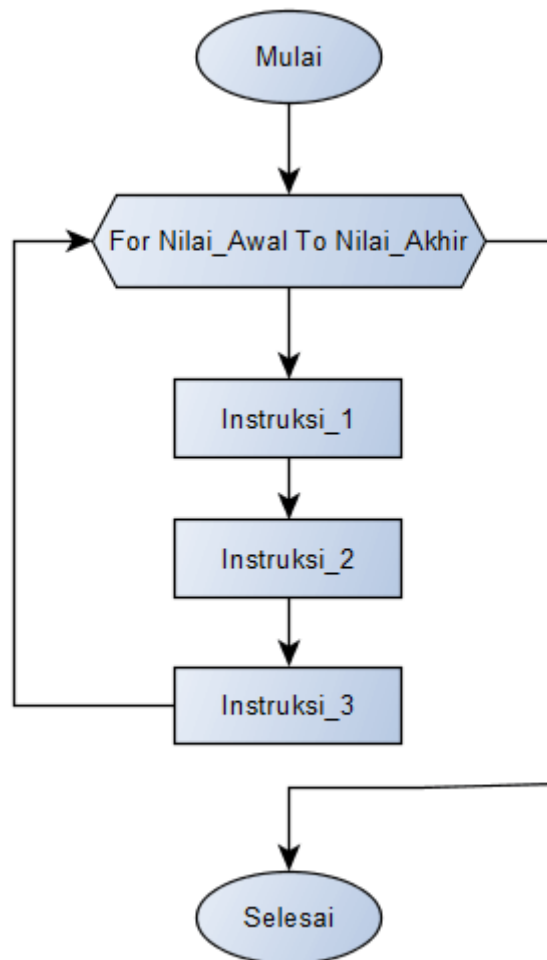
### A. Struktur FOR

Struktur dasar algoritma perulangan menggunakan instruksi FOR digunakan untuk mengulang satu baris instruksi atau beberapa baris instruksi sampai jumlah perulangan

---

yang disyaratkan terpenuhi. Ciri-ciri utama struktur perulangan menggunakan FOR adalah terdapat nilai awal dan nilai akhir yang menunjukkan syarat yang harus terpenuhi. Berikut Flowchart menggunakan FOR :

### Flowchart Struktur FOR

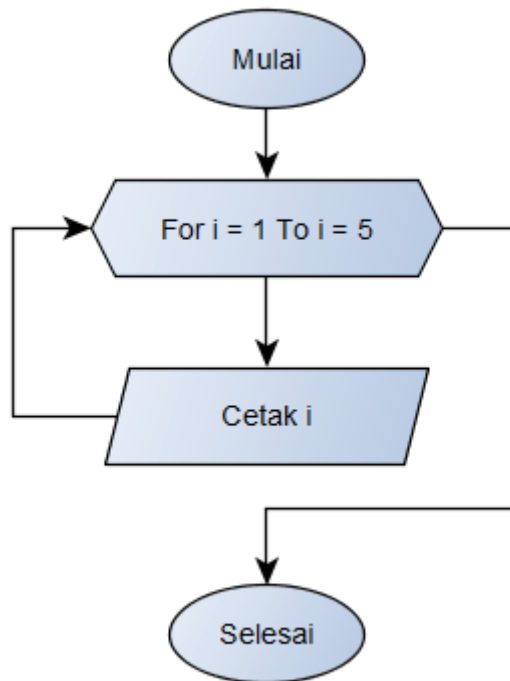


Gambar di atas menjelaskan bahwa Instruksi\_1, Instruksi\_2 dan Instruksi\_3 akan dieksekusi atau dikerjakan ketika syarat terpenuhi. Perulangan menggunakan struktur FOR dimulai dari Nilai\_Awal dan akan berhenti pada Nilai\_Akhir. Untuk lebih jelasnya kita lihat contoh algoritma mencetak angka menggunakan FOR, program akan dicetak angka 1 sampai 5.

### Deskripsi :

1. Mulai
2. Kerjakan Langkah 3 mulai  $i = 1$  sampai  $i = 5$
3. Cetak  $i$
4. Selesai

---

**Flowchart Cetak Angka dengan FOR**

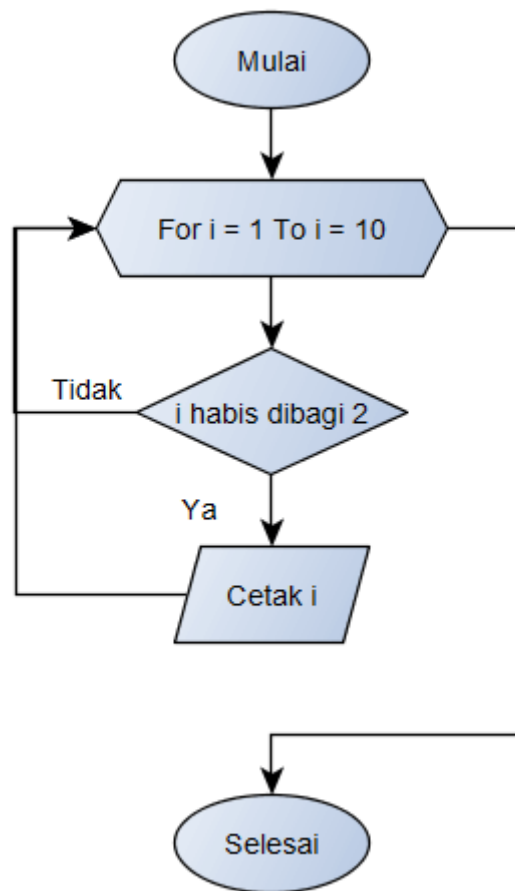
Gambar flowchart di atas, menjelaskan bahwa nilai  $i$  awal berisi 1, kemudian dicetak nilai  $i$  awal yaitu 1. Dalam struktur perulangan menggunakan FOR, nilai variabel  $i$  akan bertambah secara otomatis sehingga nilai variabel  $i$ . Setelah ditambahkan secara otomatis, sekarang  $i = 2$  lalu cetak nilai variabel  $i$ . Proses tersebut akan dijalankan sampai kondisi yang sudah ditentukan terpenuhi yaitu  $i = 5$  maka proses looping akan berhenti.

Jika belum paham kita pelajari dengan seksama contoh ke 2, algoritma Cetak Bilangan Genap menggunakan struktur FOR, program akan mencetak bilangan genap mulai dari 0 dengan batas akhir 10 dengan menggunakan struktur looping FOR.

**Deskripsi :**

1. Mulai
2. Kerjakan langkah 3 sampai langkah 4 mulai  $i = 1$  sampai  $i = 10$
3. Jika  $i$  dibagi 2 sama dengan 0 maka kerjakan langkah 4
4. Cetak  $i$
5. Selesai

### Flowchart Cetak Bilangan Genap dengan FOR



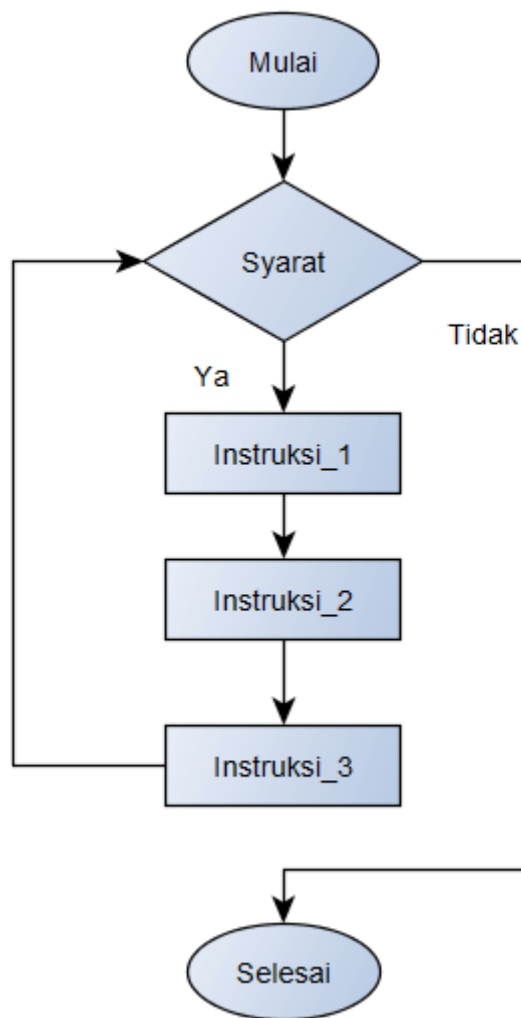
Gambar flowchart di atas, menjelaskan bahwa program akan mengeksekusi syarat yang dijabarkan dengan FOR, diketahui nilai  $i$  pertama kali berisi 1 kemudian diuji apakah nilai  $i$  tersebut habis dibagi dengan 2. Jika benar maka nilai  $i$  akan dicetak kemudian dilakukan penambahan otomatis sehingga nilai  $i$  menjadi 2. Namun jika salah, nilai  $i$  akan langsung ke langkah penambahan otomatis. Proses tersebut akan dieksekusi terus sampai nilai  $i$  lebih besar dari 10.

### B. Struktur WHILE

Struktur looping dengan menggunakan WHILE berfungsi hampir mirip dengan FOR yaitu mengulang satu baris instruksi atau beberapa baris instruksi selama syarat yang ditentukan masih terpenuhi. Ciri-ciri utama dari struktur WHILE adalah syarat yang ditentukan akan diuji lebih dahulu sebelum instruksi-instruksi dieksekusi dalam perulangan.

**Penting** diingat bahwa pada Struktur WHILE akan menguji syarat yang ditentukan terlebih dahulu, sehingga jika syarat tidak terpenuhi ada kemungkinan instruksi-instruksi pada perulangan tidak dikerjakan.

### Flowchart Struktur While

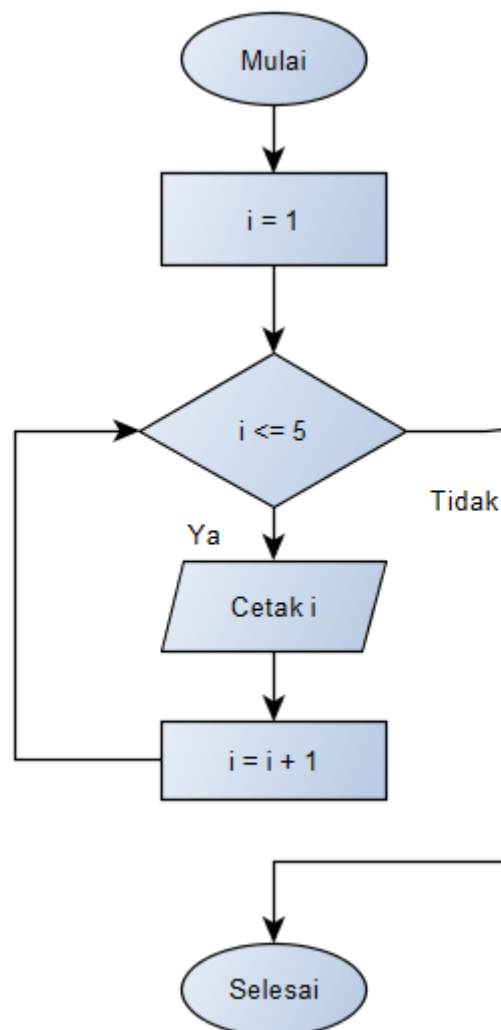


Gambar di atas menjelaskan bahwa syarat akan diuji oleh pemroses terlebih dahulu sebelum mengeksekusi instruksi-instruksi yang akan diulang. Jika syarat yang diuji benar, maka instruksi akan dieksekusi. Setelah instruksi dikerjakan maka syarat akan diuji lagi, proses perulangan ini akan berhenti jika syarat yang telah diuji bernilai salah. Untuk lebih jelasnya, kita simak contoh algoritma cetak angka menggunakan WHILE, program akan mencetak angka 1 sampai 5 dengan struktur WHILE.

**Deskripsi :**

1. Mulai
2.  $i = 1$
3. Selama  $i \leq 5$  kerjakan langkah 4 sampai langkah 5
4. Cetak  $i$
5.  $i = i + 1$
6. Selesai

### Flowchart Cetak Angka dengan WHILE



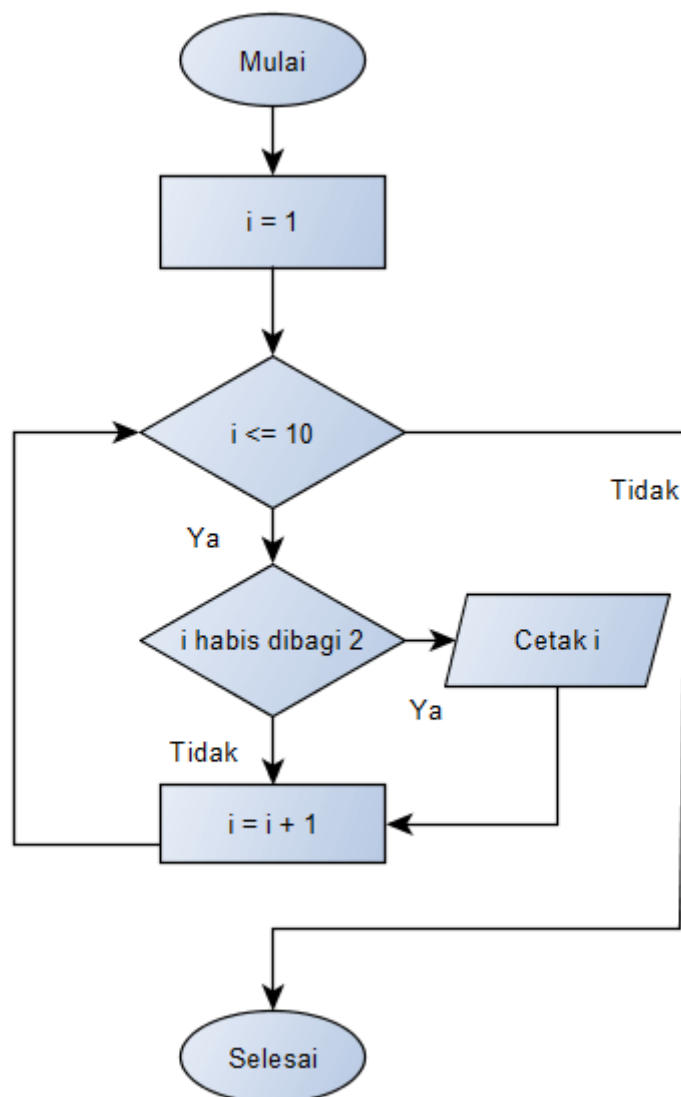
Gambar flowchart di atas, menjelaskan bahwa nilai  $i$  pertama kali bernilai 1. Kemudian akan diuji apakah nilai  $i$  lebih kecil atau sama dengan ( $\leq$ ) 5, jika benar maka nilai  $i$  dicetak. Kemudian nilai  $i$  ditambah 1, lalu nilai  $i$  akan diuji kembali apakah masih memenuhi syarat yang sudah di deklarasikan diawal, jika benar maka nilai  $i$  akan dicetak. Langkah perulangan akan berjalan terus sampai nilai  $i$  lebih besar dari 5.

**Penting** diingat bahwa pada struktur WHILE penambahan tidak dilakukan secara otomatis seperti pada struktur FOR, sehingga kita perlu menambahkan instruksi lagi.

Untuk lebih jelasnya kita liat lagi contoh berikutnya algoritma cetak bilangan genap menggunakan WHILE, program akan mencetak bilangan genap sampai 10 dengan menggunakan struktur perulangan WHILE.

**Deskripsi :**

1. Mulai
2.  $i = 1$
3. Selama  $i \leq 10$  kerjakan langkah 4 sampai langkah 6
4. Jika  $i$  habis dibagi 2 kerjakan langkah 5
5. Cetak  $i$
6.  $i = i + 1$
7. Selesai

**Flowchart Cetak Angka Genap dengan WHILE**

Pada gambar flowchart di atas, dapat dijelaskan pertama adalah nilai  $i$  diberikan nilai 1. Lalu nilai  $i$  diuji apakah nilai  $i$  lebih kecil atau sama dengan ( $\leq$ ) 10, jika benar maka akan dilakukan operasi aritmatika yaitu nilai  $i$  dibagi 2, jika habis atau hasilnya 0 maka akan dicetak nilai  $i$  dan jika sebaliknya maka nilai  $i$  tidak tercetak dan langsung

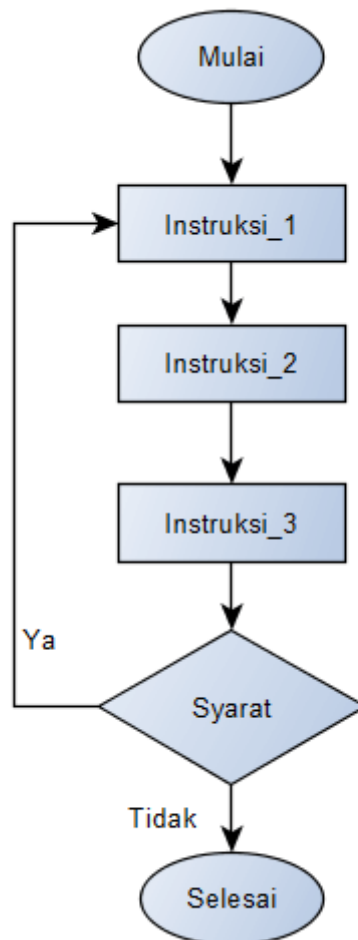
keproses berikutnya. Proses berikutnya adalah nilai  $i$  akan ditambah 1, sebelumnya nilai  $i$  adalah 1 maka setelah proses ini menjadi 2. Kemudian nilai  $i$  diuji kembali sampai nilai  $i$  lebih besar dari 10 barulah proses perulangan menggunakan WHILE berakhir.

### C. Struktur DO...WHILE

Struktur looping dengan DO...WHILE digunakan untuk mengulangi satu baris instruksi atau beberapa baris instruksi sampai syarat yang ditetapkan tidak terpenuhi. Ciri-ciri utama dari struktur DO...WHILE ialah syarat akan diuji setelah instruksi dikerjakan seluruhnya atau bisa kita katakan pengujian pada syarat dilakukan dibelakang.

**Penting** diingat bahwa Struktur menggunakan DO...WHILE berbeda dengan menggunakan WHILE, jika DO...WHILE melakukan pengujian di akhir dan WHILE melakukan pengujian di awal.

#### Flowchart struktur DO...WHILE



Pada gambar flowchart di atas, kita bisa lihat bahwa instruksi\_1, instruksi\_2 dan instruksi\_3 akan dikerjakan dahulu baru syarat akan diuji. Jika syarat yang diuji bernilai benar maka instruksi\_1 sampai instruksi\_3 akan dikerjakan kembali. Setelah dikerjakan kembali maka syarat akan diuji lagi, perulangan akan selesai jika syarat yang diuji bernilai salah. Untuk lebih jelasnya kita lihat contoh algoritma cetak angka



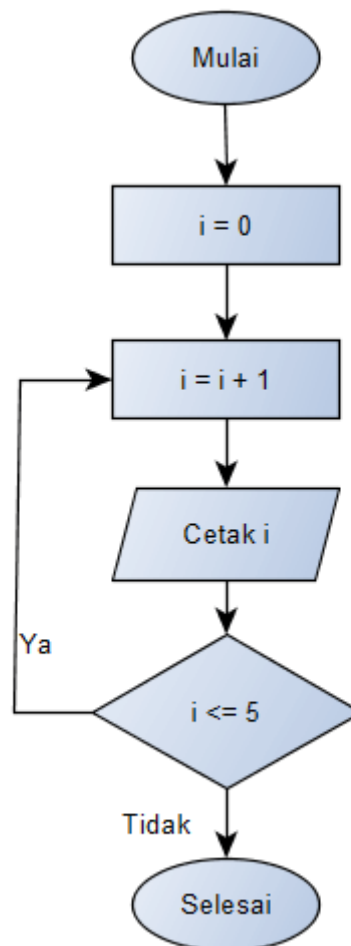
---

DO...WHILE, program akan mencetak angka 1 sampai 5 dengan menggunakan DO...WHILE.

**Deskripsi :**

1. Mulai
2.  $i = 0$
3.  $i = i + 1$
4. Cetak  $i$
5. Jika  $i \leq 5$  kerjakan langkah 3 sampai langkah 4
6. Selesai

**Flowchart Cetak Angka dengan DO...WHILE**



Pada gambar di atas, pertama kali variabel  $i$  diberi nilai awal adalah 0. Kemudian nilai  $i$  ditambah 1 sehingga nilai  $i$  menjadi 1, Lalu nilai  $i$  dicetak. Langkah selanjutnya adalah nilai  $i$  diuji apakah lebih kecil atau sama dengan ( $\leq$ ) 5, jika hasil uji bernilai benar maka akan kembali ke instruksi ke 1 yaitu nilai  $i$  ditambah 1 sehingga nilai  $i$  sekarang menjadi 2. Kemudian nilai  $i$  dicetak dan diuji kembali, perulangan akan berhenti sampai nilai  $i$  lebih besar dari 5.

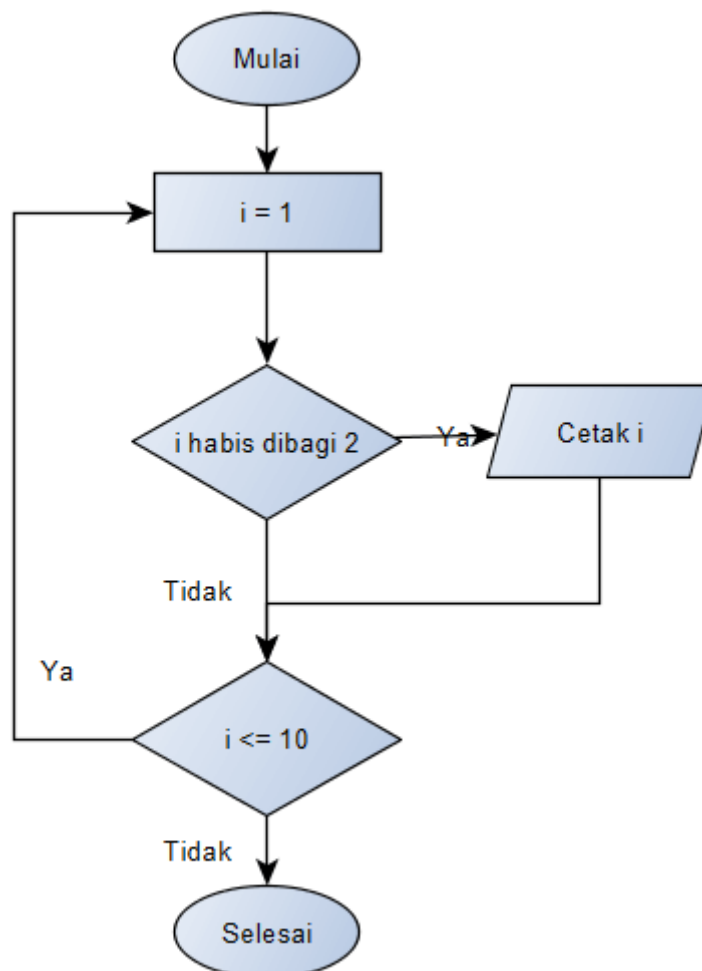
---

Kita lihat contoh berikutnya algoritma cetak bilangan genap, program akan mencetak bilangan genap dari 1 sampai 10 menggunakan struktur perulangan DO...WHILE.

**Deskripsi :**

1. Mulai
2.  $i = 1$
3. Selama  $i \leq 10$  kerjakan langkah 4 sampai langkah 6
4. Jika  $i$  habis dibagi 2 kerjakan langkah 5
5. Cetak  $i$
6.  $i = i + 1$
7. Selesai

**Flowchart Cetak Angka Genap dengan DO...WHILE**



Pada gambar flowchart, instruksi pertama adalah nilai  $i$  adalah 1. Instruksi selanjutnya pengujian nilai  $i$  habis dibagi 2, jika benar maka nilai  $i$  dicetak kemudian nilai  $i$  ditambah 1 sehingga nilai  $i$  menjadi 2. Jika salah maka instruksi nilai  $i$  ditambah 1 sehingga menjadi 2. Instruksi berikutnya adalah pengujian apakah nilai  $i$  lebih kecil atau sama

---

dengan 10, jika benar maka akan kembali pada instruksi ke 2 yaitu diuji apakah nilai i habis dibagi 2, begitulah seterusnya sampai nilai i lebih besar dari 10 barulah perulangan selesai.

### ✓ Latihan Soal

1. Jelaskan konsep dari struktur Algoritma Perulangan dan macam bentuk struktur yang ada di dalamnya.
2. Buat algoritma program untuk menampilkan nama yang diinputkan oleh user sebanyak 10 kali menggunakan struktur FOR. Buat algoritmanya dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).
3. Buat algoritma program untuk menampilkan angka dari n hingga 1 menggunakan struktur WHILE, dimana n adalah angka yang diinputkan oleh user dan n harus lebih dari atau sama dengan 1. Buat algoritmanya dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).
4. Buat algoritma program untuk menampilkan bilangan ganjil antara 1 hingga n menggunakan struktur DO...WHILE, dimana n adalah angka yang diinputkan oleh user dan n harus lebih dari atau sama dengan 1. Buat algoritmanya dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).
5. Buat contoh Masalah yang memiliki algoritma penyelesaian dalam bentuk struktur Algoritma Perulangan, dan buat algoritmanya dalam bentuk bahasa natural dan flowchart (boleh ditambah dengan pseudocodenya).

### Referensi Sumber

<https://masirwin.com/struktur-dasar-algoritma/>

----- Be Truth To Your Self And Others -----