BINAR

Programming Algorithm part 1

Silver - Chapter 2 - Topic 1

BINAR

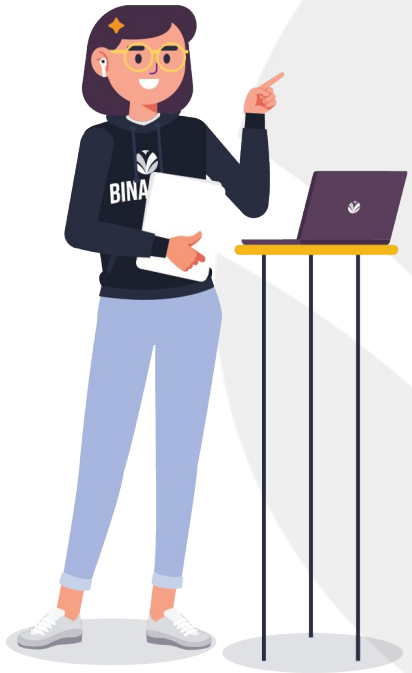
Selamat datang di Chapter 2 Topic 1 online course Back End Java dari Binar Academy!

Chapter baru, hari baru! 😀 ✨

Pada topik sebelumnya kamu selesai mempelajari tentang syntax dasar pada Java.

Pada chapter ini kita bakal ngobrolin mulai dari basic algoritma, pengenalan alat bantu algoritma, identifikasi algoritma, konsep OOP, konsep interface dan abstract class, Java collection sampai ke Java standard class.

Khusus pada topik pertama, kita mengelaborasi tentang **Programming Algorithm** dulu. Let's go!



Dari sesi ini, kita bakal bahas hal-hal berikut:

- Konsep Algorithm
- Cara Penggunaan Flowchart
- Konsep Pseudocode
- Identifikasi jenis algorithm melalui latihan

Buat pemanasan di topic pertama ini, kita bakal kenalan tipis-tipis sama materi yang namanya **Algorithm.**

Penasaran? yuk, kita geser slide-nya 👉

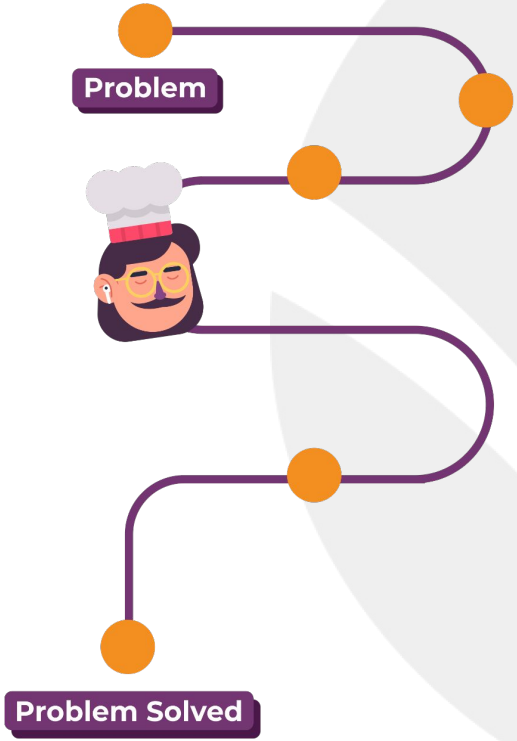
BINAR

Algoritma itu apa sih?

Algoritma adalah **alur kerja untuk memecahkan suatu masalah.**

Gampangnya nih, algoritma adalah kumpulan dari langkah-langkah yang disatukan dalam alur kerja.

Tujuannya apa kok dibuat alur kerja? Supaya itu tadi, bisa menghasilkan proses penyelesaian masalah.

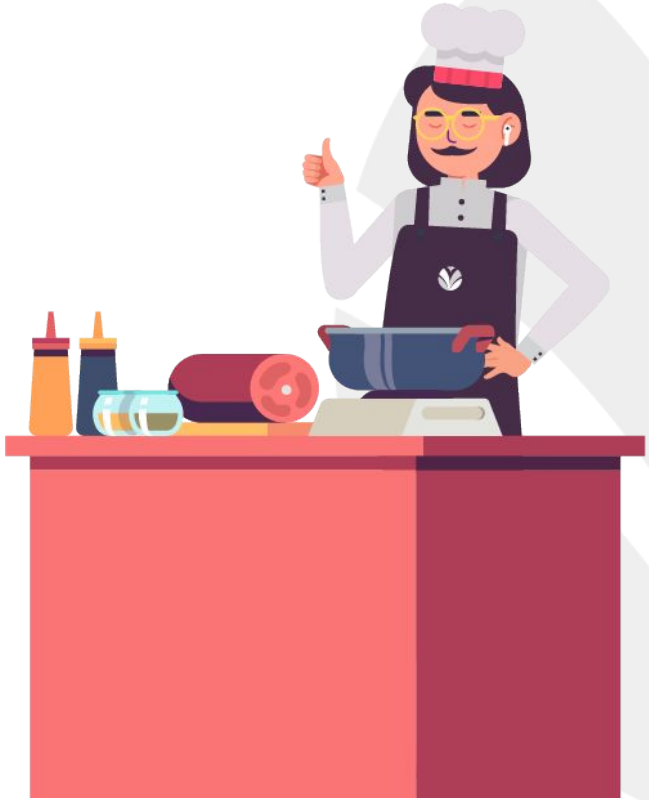


BINAR

Biar lebih paham, kita pakai analogi tutorial masak ya!

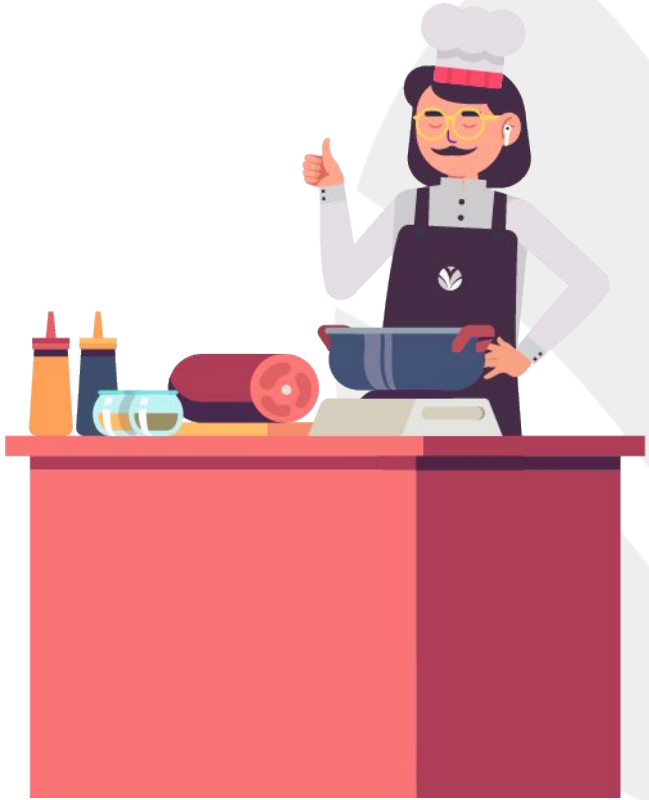
Tanpa kamu sadari, sebenarnya banyak kegiatan sehari-hari kita yang termasuk penerapan algoritma.

Salah satunya adalah tutorial memasak yang biasa kamu lihat di sosial media.



“Kok bisa?”

Iya, tutorial memasak itu mirip kaya algoritma. Yaitu mencakup langkah-langkah untuk menyelesaikan masalah berupa perut lapar.



Udah kebayang? Masih perlu contoh lagi?

Kalau gitu kita pakai contoh yang lain. Misalnya nih, kamu baru pindah rumah dan mau jajan ke Indoapril dengan cara jalan kaki. Supaya bisa sampai ke tujuan, kamu harus punya **petunjuk**, kan?

BINAR

“Kalau gitu petunjuknya apa dong?”

Okey, sekarang saatnya kita kupas satu per satu~

- Untuk menyelesaikan perjalanan, kamu perlu beberapa pertimbangan yang disesuaikan dengan kondisi yang ada.

  Misalnya “Hujan atau nggak, ya?”

- Terus, tentukan langkah-langkah yang bakal dilakukan kalau kondisi tersebut terpenuhi.

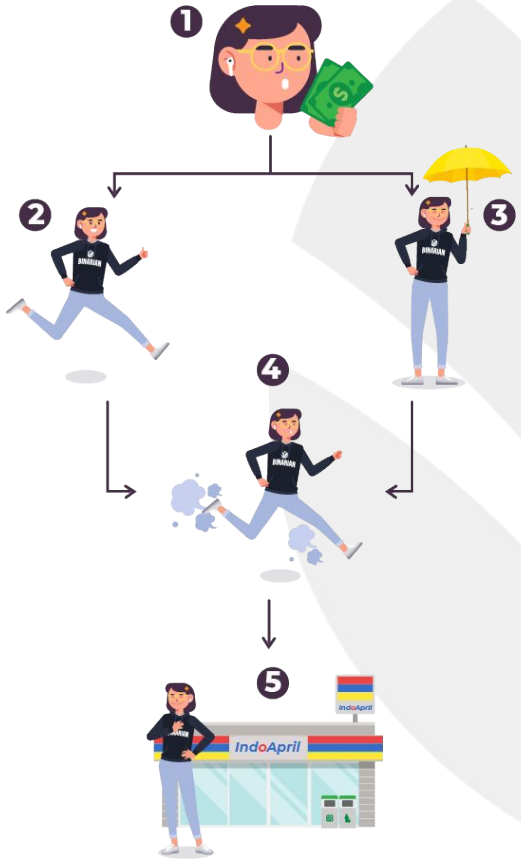
  Misalnya, kalau hujan, berarti harus bawa dan pakai payung.

- Jangan lupa, paparkan proses apa aja yang terlibat.
- Yang terakhir, prosesnya ada awal dan juga ada akhir. Berawal dari rumah dan berakhir di Indoapril.

Sesuai dengan petunjuk tadi, maka rencanamu ke Indoapril bisa digambarkan menjadi algoritma kayak gini, nih:

1. Mulai dari persiapan berangkat. Misalnya, bawa uang cash yang cukup buat jajan.
2. Kalau nggak hujan, maka perjalanan bisa dimulai.
3. Tapi kalau hujan, kamu bisa cari payung dulu sampai ketemu dan perjalanan baru bisa dilakukan.
4. Cus mulai jalan ke Indoapril.
5. Sampai deh di Indoapril.

Selamat berjajan ria~



BINAR

Pada algoritma, ada 5 kriteria yang harus diperhatikan, bestie~

Suatu alur dapat dikatakan menjadi algoritma kalau memenuhi kriteria berikut:

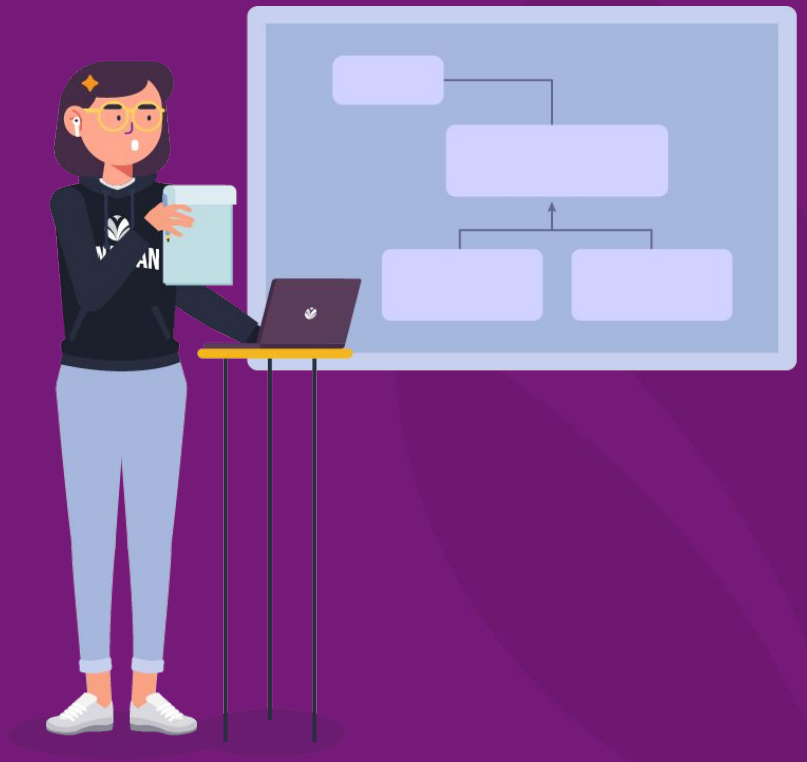
1. **Input**, yaitu nilai mentah yang bakal diproses.
2. **Output**, yaitu sesuatu yang dihasilkan setelah diproses.
3. **Definiteness** , yaitu langkah-langkah yang harus jelas dan nggak ambigu.
4. **Finiteness**, yaitu proses yang nggak punya akhir.
5. **Effectiveness**, yaitu setiap langkah yang mungkin buat dilakukan.

BINAR

Dalam membuat algoritma kayak slide sebelumnya, ada dua alat bantu yang bisa dimanfaatkan

Alat bantu ini bisa membantu algoritma supaya lebih terstruktur dan mudah dipahami. Alat bantunya ada dua , yaitu:

1. **Flowchart**
2. **Pseudocode**



Nyambung ke pembahasan sebelumnya, sekarang saatnya kita bahas alat bantu yang pertama, yaitu **Flowchart.**

Let's Go!

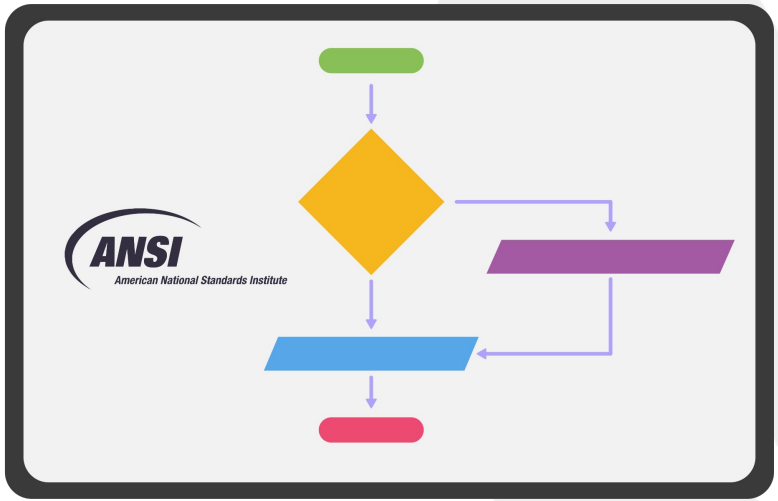
Flowchart? Apaan tuh?

Jadi, **Flowchart adalah diagram yang menjadi visualisasi alur kerja dari sebuah proses**.

Flowchart digambarkan berdasarkan simbol-simbol yang udah distandarisasi oleh **ANSI (American National Standard Institute)**.

Lewat visualisasi ini, alur pemecahan masalah bisa lebih gampang dipahami karena lebih sistematis dan terstruktur.

Gambar disamping adalah contoh simbol-simbol yang digunakan dalam menggambarkan flowchart.



“Terus, fungsi flowchart tuh buat apa, ya?”

Seorang programmer yang merancang suatu program biasanya memanfaatkan flowchart untuk melakukan dokumentasi alur kerja program yang berkaitan.

Kenapa?

Karena flowchart ini menjadi **alat bantu untuk menjelaskan alur kerja program** ke audiens non-programmer supaya program mudah dipahami dengan high level language.

BINAR

Makna simbol di balik flowchart~

Tadi udah disebutin tuh, kalau visualisasi flowhchart dibuat dengan simbol-simbol yang terstandarisasi.

Ternyata eh ternyata, masing-masing simbol ini punya **fungsinya masing-masing**.

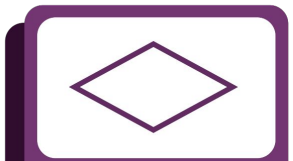
Fungsinya kayak apa, yuk kita ke halaman berikutnya!

Programming Algorithm (Part 1)

Awal dan akhir suatu program

Suatu proses

Operasi dari suatu input dan output



Proses decision making

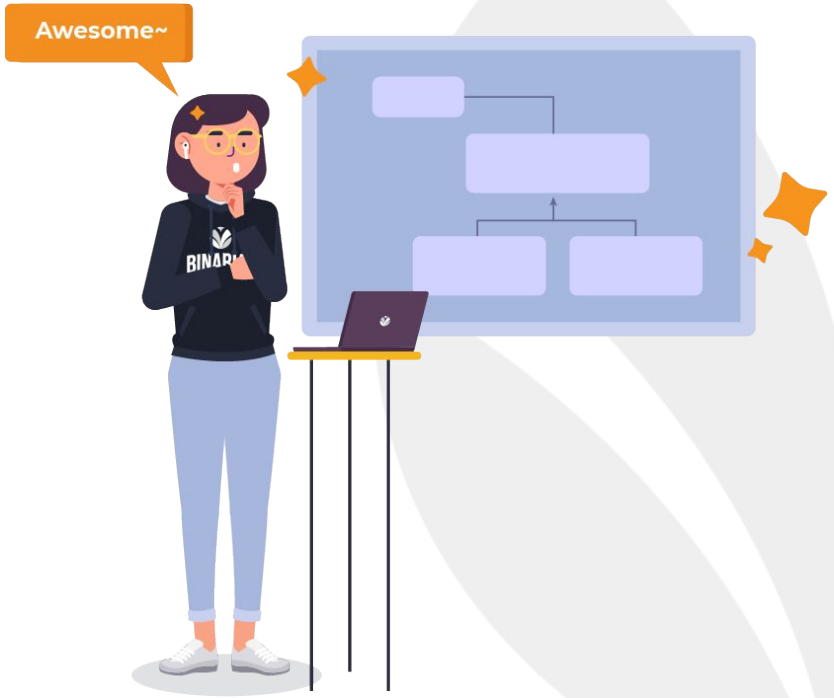
Connector atau titik temu dari dua program

BINAR

Flowchart ini punya kelebihan dan kekurangan.

Kelebihannya:

1. Flowchart mampu menyajikan control flow algoritma secara visual. Jadi, kita lebih gampang buat ngerti alur penyelesaian suatu masalah atau kondisi.
2. Lebih gampang mendeteksi kesalahan atau ketidakakuratan suatu flowchart yang kompleks dan detail.



Kekurangannya:

1. Pengerjaannya memakan waktu lama bingits. Prosesnya nih, kita harus memposisikan, ngasih label, dan ngehubungin simbol dalam flowchart.
2. Pakai tools khusus untuk membuat flowchart, itu malah berpotensi menghambat pemahaman kita tentang algoritma.

BINAR

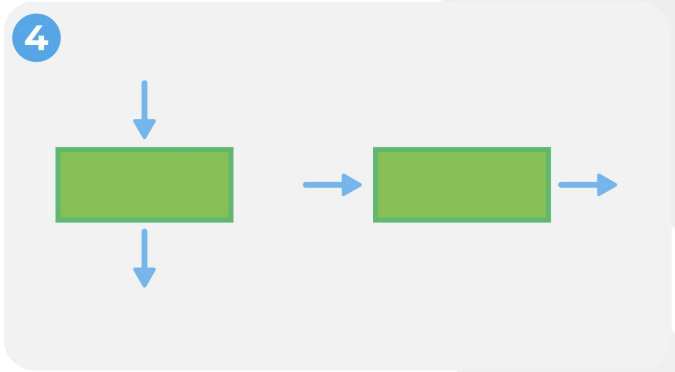
Selain itu, ada juga lho pedoman yang harus kita ikuti pas bikin flowchart~

Setelah slide ini akan ada 7 pedoman yang harus kamu tanam dalam lubuk hatimu.

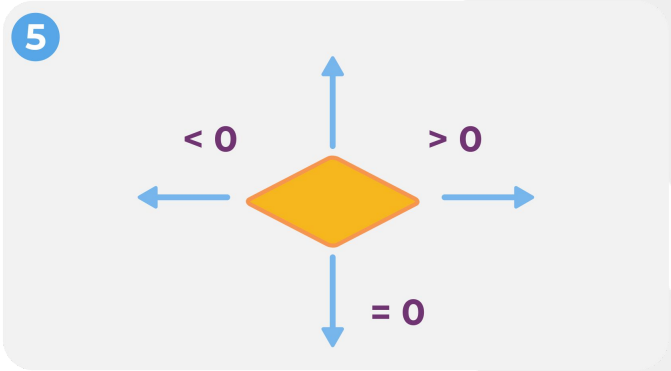
Tujuannya, supaya flowchart yang dibuat oleh programmer lebih mudah dipahami. Let's go!

1. Semua langkah digambarkan sama logical order yang tepat.
2. Flowchart digambarkan dengan rapi, jelas dan nggak ambigu.
3. Untuk menggambarkan suatu proses dari awal sampai akhir disusun dari atas ke bawah atau dari kiri ke kanan.

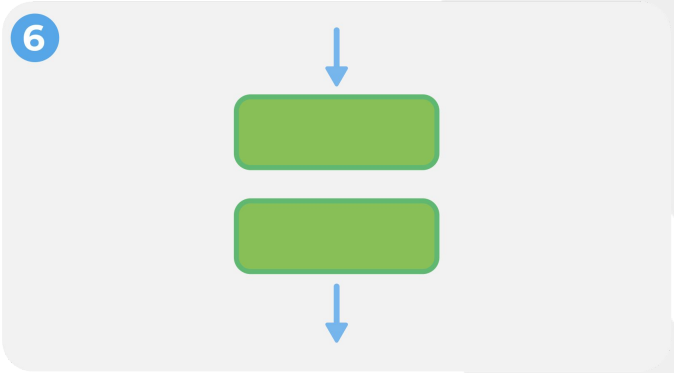
4. Hanya satu alur yang masuk dan keluar dari simbol berikut alias cuma punya satu pintu.



5. Untuk pengambilan keputusan di case yang jumlahnya banyak, bisa pake simbol decision making kayak gini.

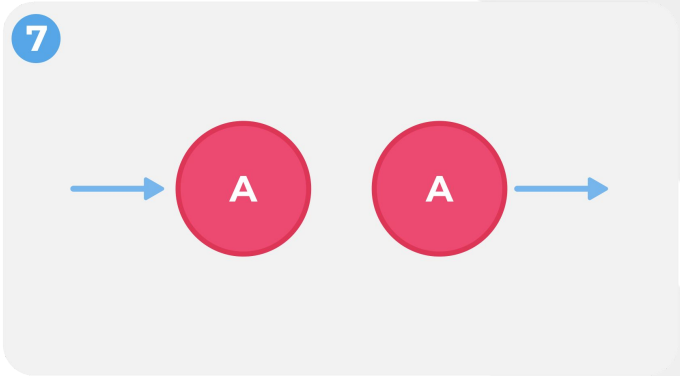


6. Awal dan akhir digambarkan sama satu panah alur aja.



7. Kalau udah nggak muat, bisa pake connector yang jadi titik buat lanjutin prosesnya.

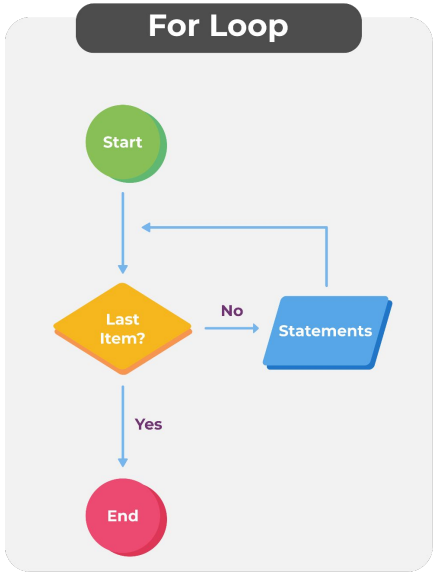
   Contohnya panah yang menuju connector A, berasal dari flowchart bagian satu. Lalu panah yang berasal dari connector A akan tersambung dengan bagian flowchart yang lain.

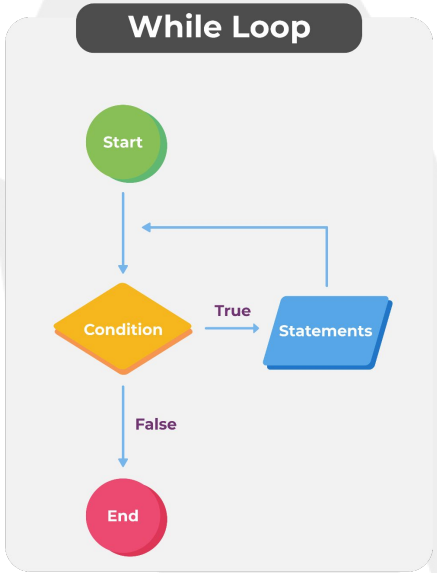


Disamping ini adalah contoh flowchart untuk menggambarkan For Loop dan While Loop

Dari gambar kita bisa mengetahui bahwa:

1. Pada perulangan loop kita udah tahu jumlah pasti dari perulangan.
2. Sedangkan di while, perulangan bakal terus berlanjut selama masih memenuhi kondisi yang ada.





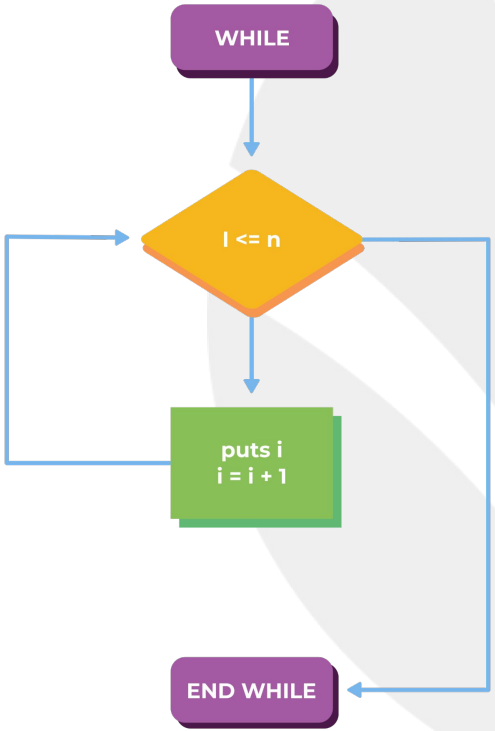
BINAR

Kalau ini adalah contoh flowchart dari program yang ngitung 1 sampai dengan n~

“Menghitung sampai n itu gimana?”

Jadi gini, gambar disamping menjelaskan bahwa program bakal terus melakukan perulangan selama nilai i<=n.

Kalau i>n, maka loop bakal berakhir alias the end~



BINAR

Tadi masih inget nggak kenapa kita membahas Flowchart? Hayoo 😬

Betul, karena flowchart adalah alat bantu algoritma. Selain Flowchart ada lagi alat bantu kedua, yaitu **Pseudocode.**

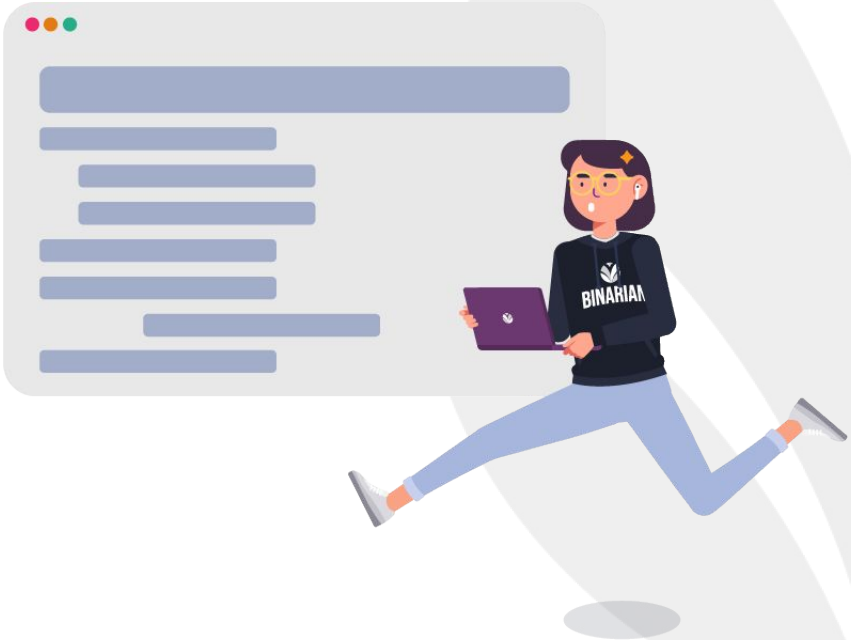
Iya, yang bakal kita bahas abis ini~

BINAR

Bukan cuma flowchart, Pseudocode juga termasuk alat bantu di algoritma, gengs!

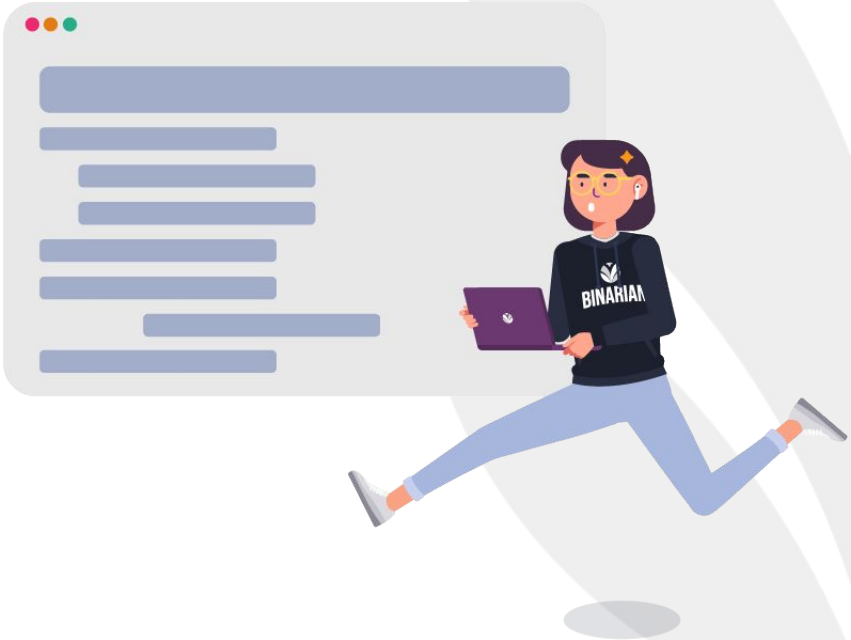
Pseudocode adalah algoritma yang ditulis mirip sebuah code.

Meskipun pseudocode ditulis mirip sebuah code, tapi pseudocode **nggak ditulis secara baku** sesuai syntax yang ada pada code.



Dari penulisan ini, pas kita menulis pseudocode, kita nggak perlu khawatir kalau ada syntax yang salah.

Iya, karena pseudocode bakal **lebih mengutamakan penulisan dari logic flow.**



Jadi, Pseudocode itu apa sih?

Pseudocode adalah draft version dari sebuah code yang bakal ditulis.

Lewat pseudocode, kita bakal kebantu banget untuk memikirkan logic flow sebelum melakukan coding.

Selain itu, pseudocode juga menjadi alat bantu dalam **menjelaskan code supaya mudah dipahami,** terutama antar developer pas lagi men-design.

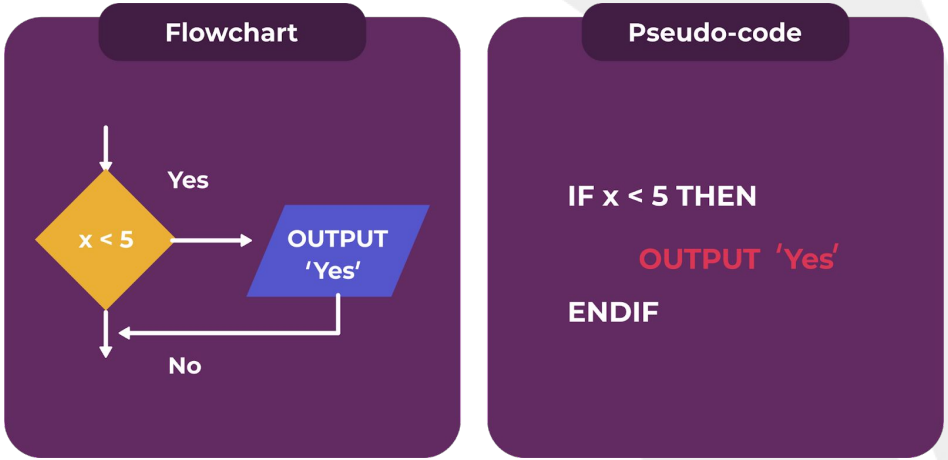
BINAR

Terus bedanya flowchart dan pseudocode apa?

Nice question! **Flowchart** adalah alur berpikir yang direpresentasikan lewat diagram alur dengan berbagai simbol.

Sedangkan **pseudocode**, merupakan alur berpikir yang direpresentasikan lewat teks atau kata-kata.

Kalo kamu perhatikan gambar di samping, keduanya menjelaskan hal yang sama, tapi cara merepresentasikannya berbeda.



BINAR

Ternyata ada notasi yang biasa dipake nih, sob~

Meskipun nggak ada panduan yang jelas dan terstandarisasi untuk membuat pseudocode, ternyata ada beberapa notasi yang biasa dipakai.

Apa aja, ya?

Yuk kita intip di slide setelah ini!

Notasi yang dimaksud yaitu:

- **Input** buat nunjukin masukkan.
- **Output**, buat nunjukin keluaran.
- **While**, buat nunjukin while loop.
- **For**, buat nunjukin for loop.
- **Repeat – Until**, buat perulangan yang punya kondisi akhir.
- **If – Then – Else**, buat ngambil keputusan dari beberapa kondisi.

BINAR

Biar kamu jadi lebih paham sama gambaran pseudocode, yuk simak contoh di samping ini.

HoursWorked merupakan input.

Kalau HoursWorked lebih besar dari NormalMax, maka program bakal ngeluarin output **overtime message**.

Apabila terjadi sebaliknya, maka yang ditampilkan adalah **regular time message.**

```
INPUT HoursWorked
  IF HoursWorked > NormalMax THEN
  Display overtime message
  ELSE
  Display regular time message
  ENDIF
```

Selesai deh teori pada topik 1 part 1 😃

Supaya pemahaman kita makin ajib, setelah slide ini bakal ada beberapa latihan yang mengasah skill kamu dalam membuat algoritma.

Latihan Programming Algorithm (Part 1)

BINAR

Latihannya tentang apa aja? Kamu akan melakukan:

- **sorting algorithm,**
- **fizz buzz algorithm,**
- **palindrome algorithm dan,**
- **sorting with recursion.**

Karena ada pada konteks belajar, anggep aja latihan ini ibarat pemanasan kamu. Gimana udah siaap? Go!

Latihan Programming Algorithm (Part 1)

Latihan pertama, silakan buatlah program untuk mengurutkan array berikut mulai dari yang terkecil ke yang terbesar:

- [100, 20, 15, 30, 5 ,75, 40]

Latihan Programming Algorithm (Part 1)

Selanjutnya, buat program untuk menampilkan game FizzBuzz, dengan memasukan angka tertentu sesuai contoh berikut:

- Kalau dimasukin angka 17, maka bakal menghasilkan.

  1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17.

- Pas kelipatan 3 bakal menghasilkan output Fizz.
- Pas kelipatan 5, bakal menghasilkan output Buzz.
- Pas kelipatan 3 dan 5 bakal menghasilkan output FizzBuzz.

Latihan Programming Algorithm (Part 1)

Lagi, nih. Buatlah program untuk mengidentifikasi sebuah input apakah termasuk palindrome algorithm atau bukan, misalnya:

- Input: BINAR

  Output: String is not palindrome

- Input: KATAK

  Output: String is palindrome

Latihan Programming Algorithm (Part 1)

Ada spesial buat kamu. Buat program untuk melakukan sorting dari kecil ke besar pake perulangan recursive untuk array:

- [64, 34, 25, 12, 22, 11, 90]

Gimana latihan programming algorithmnya? Apa cukup menantang?

Dari seluruh soal latihan, menurutmu soal mana yang paling sulit? Boleh dong sini cerita sama Sabrina

Nah, selesai sudah pembahasan kita di Chapter 2 Topic 1 ini.

Selanjutnya, kita bakal bahas tentang **Programming Algorithm part 2.**

Penasaran kayak gimana? Yuk langsung ke topik selanjutnya~