**Pengenalan Aksi Sekuensial**

Selama Anda mempelajari beberapa materi pertama kelas ini, Anda telah membuat banyak program sederhana. Namun, program yang Anda buat masih terdiri dari satu atau dua baris kode saja. Ke depannya, Anda akan membuat kode program yang tidak hanya terdiri dari satu baris, tetapi bisa banyak baris dan lebih kompleks.

Kali ini, Anda akan mempelajari cara komputer menjalankan kode program yang kompleks dan memiliki banyak baris. Ini adalah modal dasar Anda untuk paham hal tersebut.

Dalam bahasa pemrograman Python, program yang Anda buat akan dijalankan secara sekuensial. Apa maksudnya?

Kode yang Anda bangun akan berjalan sesuai dengan urutan perintahnya. Aksi sekuensial sendiri memiliki makna sebagai sederetan instruksi yang akan dijalankan oleh komputer berdasarkan urutan penulisannya.

Dengan kata lain, program yang Anda bangun haruslah memiliki awal dan akhir. Jadi, ketika program tersebut dijalankan, bisa diketahui waktu berakhirnya. Simak kode di bawah ini untuk melihat implementasinya.

print("Selamat datang dalam program Python!\n")

print("Silakan masukkan data diri Anda.")

nama = input("Masukkan nama Anda: ")

tahun\_lahir = input("Masukkan tahun lahir Anda: ")

umur = 2023 - int(tahun\_lahir)

print(f"Selamat datang {nama} dalam program Python, per 2023 umur Anda adalah {umur} tahun.\n")

print("Terima kasih telah menggunakan program Python!")

Output:

Selamat datang dalam program Python!

Silakan masukkan data diri Anda:

Masukkan nama Anda: Evans

Masukkan tahun lahir Anda: 2005

Selamat datang Evans dalam program Python, per 2023 umur Anda adalah 18 tahun.

Terima kasih telah menggunakan program Python!

Untuk memaksimalkan contoh implementasi, silakan salin kode di atas dan jalankan menggunakan notebook atau IDE Anda. Mari bedah kode tersebut.

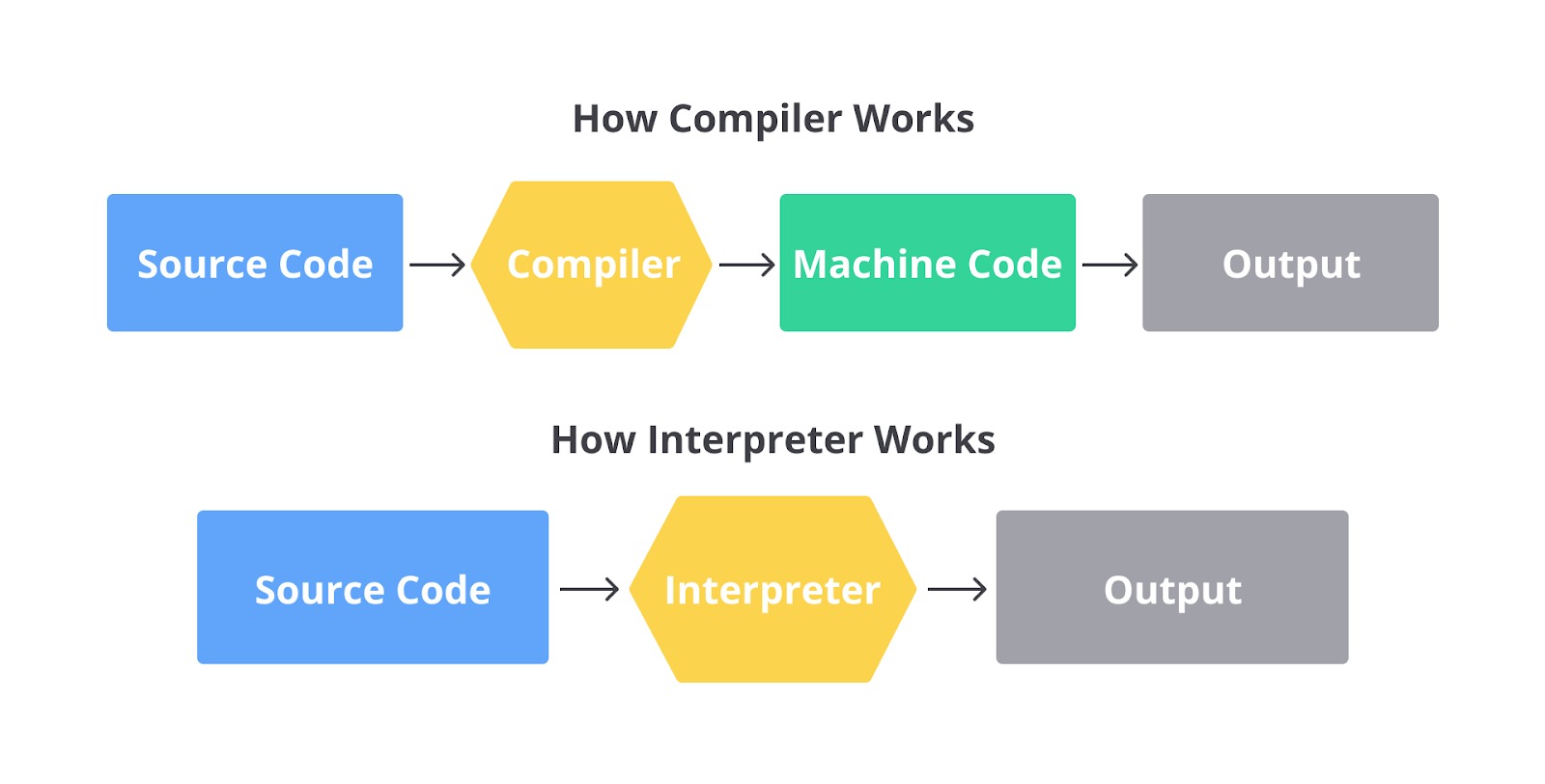
1. Komputer akan menjalankan kode pertama dengan menampilkan teks "Selamat datang dalam Program Python".
2. Setelah berhasil, kode kedua akan dijalankan dengan menampilkan teks "Silakan masukkan data diri Anda."
3. Lalu, kode ketiga akan dijalankan. Program akan meminta Anda memasukkan nama sembari memunculkan teks "Masukkan nama Anda: ". Input ini akan disimpan pada variabel bernama "nama".
4. Kemudian, kode keempat akan dijalankan. Program akan meminta Anda memasukkan tahun lahir sembari memunculkan teks "Masukkan tahun lahir Anda: ". Input ini akan disimpan pada variabel bernama tahun\_lahir.
5. Setelah itu, variabel tahun\_lahir akan dikalkulasikan untuk mengetahui umur Anda per tahun 2023. Hasil kalkulasi tersebut disimpan pada variabel umur.
6. Program akan memunculkan teks "Selamat datang {nama} dalam program Python, per 2023 umur Anda adalah {umur} tahun." Dengan {nama} dan {umur} merupakan nilai dari variabel dengan nama yang sama.
7. Program ditutup dengan memunculkan teks "Terima kasih telah menggunakan program Python!".

Keseluruhan kode tersebut menggambarkan bahwa program akan dijalankan berdasarkan urutan perintahnya. Perlu diperhatikan bahwa terdapat program yang akan berubah hasilnya jika urutan baris instruksinya diubah. Ada juga program yang hasilnya TIDAK akan berubah jika urutan baris instruksinya diubah.

**Python Interpreter**

Python termasuk bahasa pemrograman yang mudah dimengerti oleh manusia karena sintaksnya yang mudah dipahami. Tahukah Anda bahwa proses komputer menjalankan kode yang Anda bangun tidak sesederhana Anda memahaminya?

Kode dari program Python yang Anda bangun akan ditransformasi menjadi kode yang mudah dimengerti oleh mesin menggunakan program *compiler* atau *interpreter*. *Compiler* merupakan program yang akan menerjemahkan bahasa pemrograman menjadi bahasa mesin sebelum dijalankan dan menghasilkan output. Ini artinya program yang Anda bangun secara keseluruhan akan diubah terlebih dahulu semuanya menjadi bahasa mesin.



Hal berbeda terjadi pada *interpreter*, yangakan menerjemahkan bahasa Python satu per satu dan menghasilkan output secara langsung. Hal ini memungkinkan Anda untuk melihat hasil program segera setelah satu baris kode dieksekusi hingga selesai. Implementasi interpreter ada pada mode interaktif Python. Anda dapat menjalankan satu atau dua lebih baris kode secara langsung dan melihat hasilnya.

Python memiliki aturan yang membantu Anda untuk menulis kode dengan baik dan benar. Guido van Rossum selaku pembuat bahasa Python merasa bahwa kode lebih sering dibaca dibandingkan ditulis. Dengan membuat kode yang baik dan benar akan memudahkan Anda untuk memahami kode bahkan membantu interpreter atau compiler berjalan dengan baik.

**Block Code**

Sebuah program Python dapat berupa pernyataan atau statement, bisa juga terdiri atas blok kode. Sebuah blok merujuk pada potongan kode program Python yang dijalankan sebagai satu unit. Kode blok dapat berupa modul, fungsi, kelas, control flow, dan sebagainya. Tenang! Anda akan mempelajari istilah-istilah tadi lebih dalam pada materi-materi selanjutnya.

Kode di bawah merupakan contoh implementasi dari blok kode. Kode ini merupakan penerapan materi perulangan. Anda akan mempelajari lebih detail pada materi "Control Flow".

Mengapa memahami kode blok penting? Alasannya adalah Anda harus membuat kode yang mudah dimengerti oleh Anda dan orang lain sebagai seorang programmer. Selain itu, kode blok yang baik dan benar akan memudahkan interpreteratau compiler untuk berjalan dengan baik dan tidak menghasilkan error.

Python sangat memperhatikan indentasi karena hal tersebut tidak hanya membantu merapikan kode yang Anda bangun, tetapi juga menjelaskan satu kode blok secara utuh. Dengan indentasi, program akan mengetahui letak awal dan akhir sebuah blok kode. Ke depannya, Anda akan melihat sebuah fungsi, modul, dan kelas yang ada dalam Python juga dengan memperhatikan indentasi untuk menyatakan kode blok secara utuh.

**Case-sensitive**

Python termasuk bahasa pemrograman *case-sensitive*. Ini artinya Python memperlakukan huruf besar dan kecil sebagai karakter yang berbeda dalam penamaan variabel, nama fungsi, atau penulisan kode secara umum.

**One-liner**

Selain membangun kode berdasarkan bloknya, Anda juga dapat membuat sebuah kode hanya dalam satu baris saja atau berupa *single statement*. Konsep ini dikenal sebagai *one-liner*.

*One-liner* merupakan gaya penulisan pada Python yang memungkinkan Anda untuk membuat sebuah kode hanya dalam satu baris. *One-liner* adalah salah satu keunggulan dalam Python yang susah untuk diimplementasikan bagi beberapa bahasa pemrograman lainnya.

Tujuan dari *one-liner* ini adalah membuat satu baris kode yang singkat dan jelas. Perlu diingat bahwa tidak semua kode blok dapat dijadikan *one-liner*, seperti deklarasi fungsi, modul, dan kelas.

x = 1

y = 2

temp = x

x = y

y = temp

print("Setelah pertukaran: ")

print("x = ", x)

print("y =", y)

Output:

Setelah pertukaran:

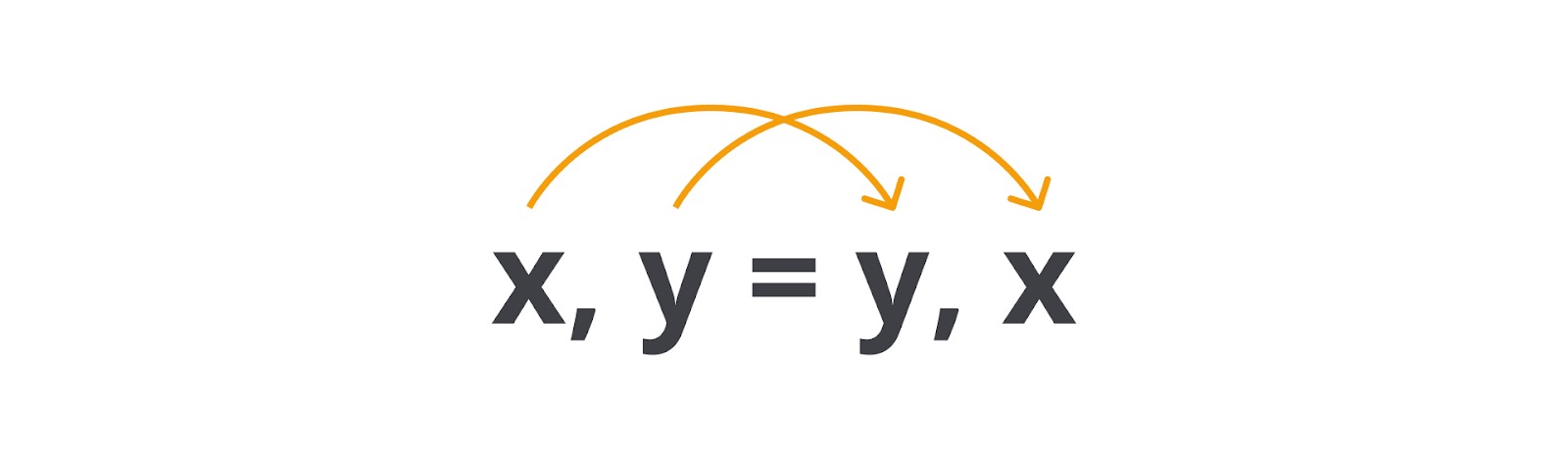
x = 2

y = 1

Mari bedah kode tersebut.

1. Anda menginisialisasi variabel x dengan nilai 1 dan variabel y dengan nilai 2.
2. Anda menginisialisasi variabel temp dengan nilainya adalah variabel x. Hal ini menyebabkan variabel temp memiliki nilai 1.
3. Anda menginisialisasi variabel x dengan nilai baru, yakni variabel y. Hal ini menyebabkan nilai dari variabel x menjadi 2.
4. Anda menginisialisasi variabel y dengan nilai baru, yakni variabel temp. Hal ini menyebabkan nilai dari variabel y menjadi 1.
5. Proses penukaran variabel telah selesai. Selanjutnya, Anda menampilkan nilai pada variabel tersebut dengan sintaks "print()".

Mungkin Anda bertanya, mengapa variabel x dan y bisa berubah? Ingat konsep aksi sekuensial, program akan menjalankan kode tersebut baris per baris. Jadi, nilai dari variabel x dan y setelah inisialisasi pertama akan berubah karena pada sintaks berikutnya Anda menetapkan nilai baru pada variabel x dan y. Anda menggunakan variabel bantuan, yakni variabel "temp" untuk menyimpan nilai awal dari variabel x.



Pada kode di atas, Anda seolah-olah menginisialisasikan ulang variabel x dengan nilai variabel y di sebelah kanan. Anda juga menginisialisasikan ulang variabel y dengan nilai variabel x yang ada di sebelah kanan. Sederhana, bukan? Dengan menginisialisasikan ulang variabel masing-masing, nilai tersebut pada akhirnya bisa saling bertukar.