C:\Users\aisyah\IdeaProjects\EDMS

**Data Types & Variable**

Data types atau tipe data adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Untuk mengembangkan sebuah program, ada beberapa tipe data yang akan kita pelajari. Di antaranya adalah **Character**, **String**, **Array**, **Numbers**, dan **Booleans**. Semuanya akan kita bahas sejelas dan sesederhana mungkin di dalam submodul ini.

Namun, ada satu hal yang kita perlu tahu terlebih dahulu, yaitu **Variabel**. Umumnya variabel digunakan untuk menyimpan informasi atau nilai yang akan dikelola di dalam sebuah program. Sebuah variabel akan membutuhkan kata kunci var atau val, **identifier**, **type**, serta **initialization**. Kira-kira strukturnya seperti berikut.

var identifier: Type = initialization

Mari kita ulas setiap bagian pada struktur variabel di atas.

* **var**atau **val**  
  var atau val digunakan untuk mengontrol nilai dari sebuah variabel. Dengan kata kunci var, kita bisa mengubah nilai yang sudah diinisialisasikan.Berikut contohnya.

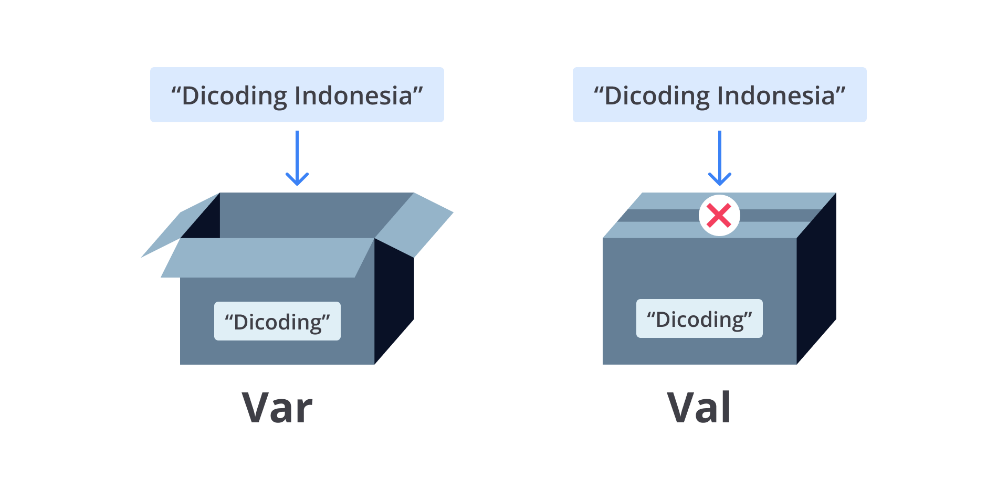
var company: String = "Dicoding"

company = "Dicoding Academy"

Variabel company yang awalnya memiliki nilai **“Dicoding”**sekarang sudah diubah menjadi **“Dicoding Academy”**. Ketika menggunakan kata kunci val, kita tidak bisa mengubah nilai yang sebelumnya sudah diinisialisasi. Jika kita memaksa untuk mengubahnya, akan terjadi eror seperti berikut.

val company: String = "Dicoding"

company = "Dicoding Academy" // Val cannot be reassigned



* **Identifier**  
  *Identifier* merupakan nama dari sebuah variabel. Pada contoh kode di atas yang merupakan *identifier* adalah company. Perlu diketahui bahwa di dalam sebuah program, kita tidak bisa membuat lebih dari 1 (satu) variabel dengan nama sama.
* **Type**  
  Pada bagian inilah kita menentukan tipe data dari variabel tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Tipe data dari contoh variabel di atas adalah String. Karena Kotlin mendukung *type inference,* kita diperbolehkan untuk tidak menuliskan tipe data secara eksplisit.

val company = "Dicoding"

* **Initialization**

Dan yang terakhir adalah *initialization* atau nilai awal dari sebuah variabel. Pada contoh di atas, yang berperan sebagai *initialization* adalah **“Dicoding”**dan **“Dicoding Academy”**.

Pada kode di atas, kedua nilai dari variabel firstWord dan lastWord akan digabungkan menjadi satu nilai string. Berbeda ketika kita menggunakan operator + pada variabel yang bertipe Int seperti berikut.

fun main() {

val valueA: Int = 10

val valueB = 20

print(valueA + valueB)

}

/\*

output: 30

\*/

Selain operator tambah (**+**), ada juga operator lain yang perlu Anda ketahui untuk memudahkan dalam menuliskan kode Kotlin. Berikut beberapa di antaranya.

1. **+**, **-**, **\***, **/**, dan **%**adalah operator matematika. Anda bisa mengelola variabel bertipe angka dengan berbagai operator matematika yang ada.
2. **=**adalah operator penugasan. Biasanya digunakan untuk menentukan nilai dari sebuah variabel.
3. **+=,-=**, **\*=**, **/=**, dan **%=**adalah operator penugasan yang menggabungkan konsep matematika.
4. **&&**, **||**, dan**!** adalah operator logika. Operator AND (&&) dan OR (||) digunakan untuk membandingkan dua nilai sesuai dengan kondisi yang Anda inginkan. Berbeda dengan operator NOT (!), biasanya ia digunakan untuk melakukan negasi pada value yang ada.
5. **++** dan **--** adalah operator *increment* dan *decrement*yang untuk menambah atau mengurangi satu nilai.

Tentunya masih banyak simbol spesial dan operator yang bisa Anda gunakan dalam membangun kode dengan Kotlin. Anda bisa melihat selengkapnya di [Keyword di Kotlin](https://kotlinlang.org/docs/keyword-reference.html#hard-keywords).

Anda juga dapat mempelajari lebih lanjut referensi berikut untuk memperkaya wawasan Anda.

* [Basic types | Kotlin Documentation](https://kotlinlang.org/docs/basic-types.html)
* [The val Property != Immutable in Kotlin](https://proandroiddev.com/the-val-property-immutable-in-kotlin-2e4cf49207d0)

Ketika kita mengembangkan sebuah program kita pasti membutuhkan variabel dengan tipe data yang mampu menyimpan nilai berbentuk teks. Terdapat dua (2) tipe data yang bisa kita gunakan, yaitu **Char** dan **String**.

**Char**

Characters direpresentasikan menggunakan tipe Char. Untuk mendefinisikan sebuah variabel dengan tipe data Char kita bisa menggunakan tanda kutip tunggal **(' ')** seperti berikut:

val character = 'A'

Tipe data Char hanya dapat kita gunakan untuk menyimpan karakter tunggal. Sebaliknya jika kita memasukkan lebih dari 1 (satu) karakter, akan terjadi eror:

val character: Char = 'ABC' // Incorrect character literal

Yang menarik, kita bisa melakukan operasi *increment* (++) dan *decrement* (--) pada sebuah variabel dengan tipe data Char seperti berikut:

fun main() {

var vocal = 'A'

println("Vocal " + vocal++)

println("Vocal " + vocal++)

println("Vocal " + vocal++)

println("Vocal " + vocal--)

println("Vocal " + vocal--)

println("Vocal " + vocal--)

println("Vocal " + vocal--)

}

/\*

output:

Vocal A

Vocal B

Vocal C

Vocal D

Vocal C

Vocal B

Vocal A

\*/

**String**

String merupakan tipe data yang mirip dengan Char. Ia dapat digunakan untuk menyimpan nilai berupa teks. Perbedaannya, String bisa menampung beberapa karakter di dalamnya.

String direpresentasikan menggunakan tipe String. Nilai yang berada di dalam sebuah variabel dengan tipe data String merupakan kumpulan dari beberapa karakter. Kita bisa mendefinisikan variabel tersebut dengan tanda petik ganda (**" "**) seperti berikut.

val textString = "Kotlin"

Caranya, manfaatkan **indexing** seperti berikut.

fun main() {

val text = "Dicoding"

val firstChar = text[3]

print("The 4th character of the $text is $firstChar")

}

/\*

output : The 4th character of the Dicoding is o

\*/

**Apa itu Indexing?**  
*Indexing* merupakan sebuah cara yang memudahkan kita untuk mengakses **elemen** yang berada di dalam sebuah Collection dengan memanfaatkan *index* atau posisi dari elemen tersebut. Posisi dari sebuah elemen pada umumnya dimulai dari angka **0**. Untuk materi tentang Collection akan sama-sama kita pelajari pada submodul berikutnya. Semangat!

Nilai **0** yang berada pada *indexing* di atas adalah posisi karakter yang akan diakses. Selain itu, kita juga dapat melakukan iterasi terhadap objek String dengan menggunakan **for-loop** seperti berikut.

fun main() {

val text = "Kotlin"

for (char in text){

print("$char ")

}

}

/\*

output : K o t l i n

\*/

**Escaped String**

Kotlin memiliki dua jenis tipe **Literal String**, yaitu **Escaped String**dan**Raw String.** Mari kita mulai pembahasannya dari Escaped String terlebih dahulu.

**Escaped String** memungkinkan kita untuk mengurangi ambiguitas nilai yang berada di dalam sebuah String. Misalnya, ketika kita mendefinisikan sebuah String berikut.

val statement = "Kotlin is Awesome!"

Kemudian, kita ingin menambahkan tanda petik ganda di dalam sebuah String seperti berikut.

val statement = "Kotlin is "Awesome!""

Selain **\”** di atas, terdapat beberapa karakter lain yang dapat digunakan untuk melakukan *escaped* di dalam sebuah String, antara lain:

* **\t**: menambah tab ke dalam teks.
* **\n**: membuat baris baru di dalam teks.
* **\’**: menambah karakter *single quote* kedalam teks.
* **\”**: menambah karakter *double quote* kedalam teks.
* **\\**: menambah karakter *backslash* kedalam teks.

Selain itu, kita juga bisa menambahkan sebuah **Unicode** ke dalam sebuah String seperti berikut.

**Raw String**

Setelah membahas Escaped String, kita akan beralih ke **Raw String** yang memungkinkan untuk menuliskan *multiline* dan *arbitrary text*. Ketika ingin membuat beberapa baris String, biasanya kita melakukan *escaped* terhadap String dengan memanfaatkan karakter *escape* **\n** seperti berikut.

val line = "Line 1\n" +

"Line 2\n" +

"Line 3\n" +

"Line 4\n"

Dengan Raw String, kita dapat membuatnya dengan cara yang lebih mudah yaitu seperti berikut.

fun main() {

val line = """

Line 1

Line 2

Line 3

Line 4

""".trimIndent()

print(line)

}

/\*

output:

Line 1

Line 2

Line 3

Line 4

\*/

Pada kode di atas, kita mendefinisikan sebuah Raw String menggunakan *triple quote*(**""" """**). Raw String memungkinkan kita untuk membuat beberapa baris String tanpa penggabungan (*concatenation*) dan penggunaan karakter *escaped*. Lebih lanjut, kita menggunakan metode trimIndent() untuk menghilangkan indentasi yang kita gunakan pada kode.

**String Template**

fun main() {

val name = "Kotlin"

print("My name is " + name)

/\*

output : My name is Kotlin

\*/

}