

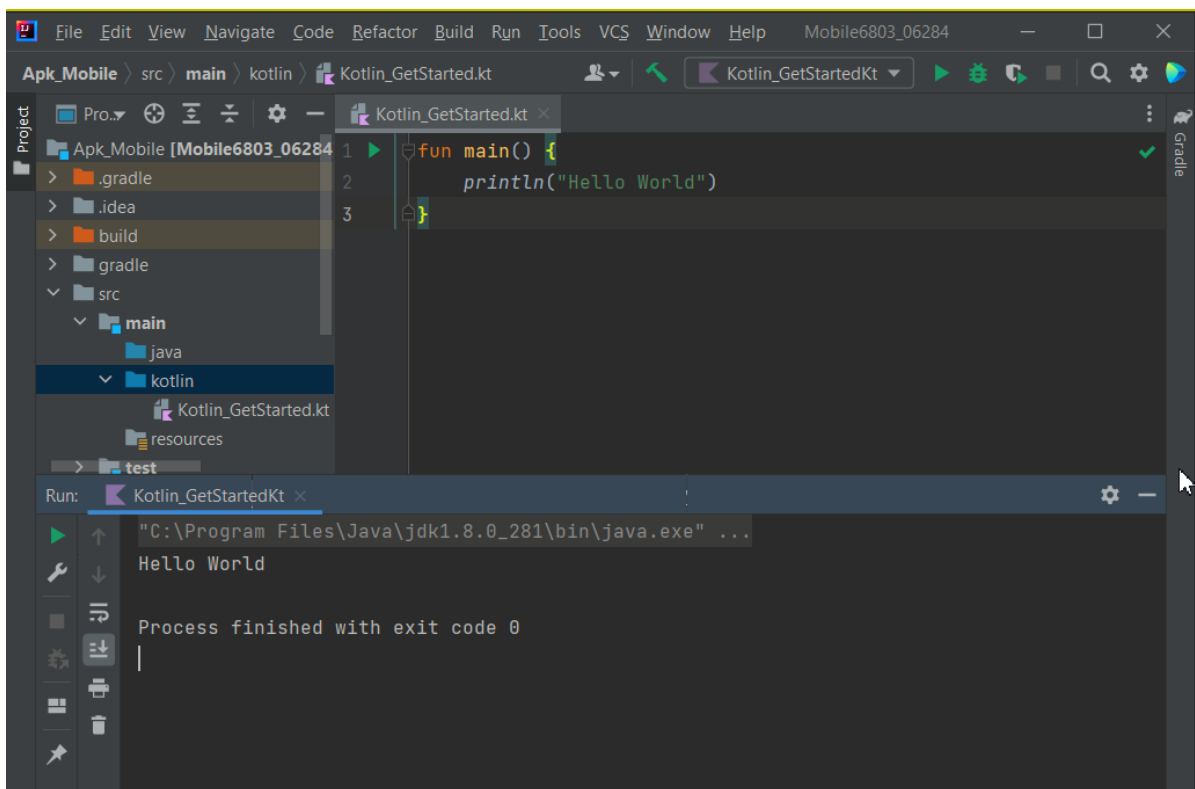
**Name: Dewi Putri Ayuningsih**

**Nim : A12.2019.06284**

## **APLIKASI MOBILE (A12.6803)**

- **Kotlin Get Started and Syntax**

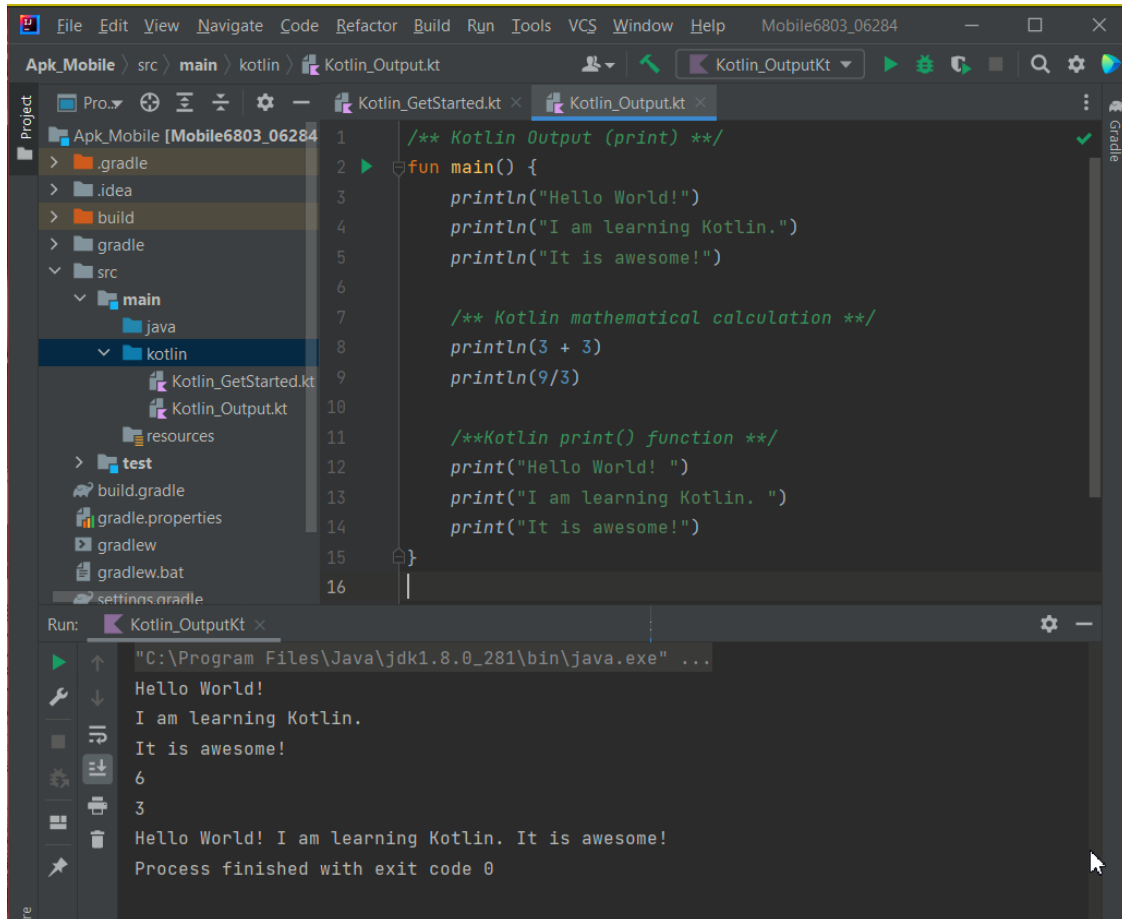
Merupakan Latihan untuk menunjukkan sintaks dasar pada sebuah bahasa pemrograman. Pada bab ini juga merupakan penjelasan dari kegunaan fungsi yang akan sering kita temui sebelum masuk pada bab fundamentalnya. Seperti ( `fun main(){}` ) yang akan digunakan untuk **mengeksekusi** kode. Kode apa pun yang ada dalam tanda kurung keriting akan **dieksekusi**.



- **Kotlin Output (Print Text) and Kotlin Comment**

Komentar yang dapat digunakan untuk menjelaskan kode Kotlin, dan membuatnya lebih mudah dibaca. juga digunakan untuk mencegah eksekusi saat menguji kode alternatif.

Ada macam-macam comment seperti (*/\*...\*/*, *//*, */\*\*...\*/*)



The screenshot shows an IDE window with a project named 'Apk\_Mobile [Mobile6803\_06284]'. The left sidebar displays the project structure, including folders like 'gradle', 'build', 'src', 'main', 'java', 'kotlin', 'resources', and 'test'. The main editor area shows the file 'Kotlin\_Output.kt' with the following code:

```
1  /** Kotlin Output (print) */
2  fun main() {
3      println("Hello World!")
4      println("I am learning Kotlin.")
5      println("It is awesome!")
6
7      /** Kotlin mathematical calculation */
8      println(3 + 3)
9      println(9/3)
10
11     /**Kotlin print() function */
12     print("Hello World! ")
13     print("I am learning Kotlin. ")
14     print("It is awesome!")
15 }
16
```

The bottom panel shows the output of the program:

```
Run: Kotlin_OutputKt x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
Hello World!
I am learning Kotlin.
It is awesome!
6
3
Hello World! I am learning Kotlin. It is awesome!
Process finished with exit code 0
```

- **Kotlin Variabel and Kotlin String**

Disini bab Variabel dan String saya jadi kan 1 file karena bab variable sendiri merupakan kontainer untuk menyimpan nilai data. dengan kata kunci **val** and **var**.  
dan String sendiri berisi kumpulan karakter yang dikelilingi oleh tanda kutip ganda “Hello” yang digunakan untuk menyimpan teks.

The screenshot shows an IDE window with the following components:

- Project Explorer:** Shows the project structure with folders like `gradle`, `src`, and `main`. The `main` folder contains a `kotlin` sub-folder with several files, including `Kotlin_variables&String.kt` which is currently selected.
- Code Editor:** Displays the Kotlin code for `Kotlin_variables&String.kt`. The code includes imports, comments, and a `main` function that declares and uses variables and strings.

```
1 import java.util.*
2
3 /** Kotlin Variable Var & Val */
4 fun main(){
5     var name = "Hello " // String (text)
6     val nim = 112201906284 // String (Number)
7     println(" \'Variables\' \n" + name + "Dewi Putri Ayuningsih") //print the value of
8     println(nim) //print the value of nim
9
10    /** Kotlin STRING */
11    val txt = "Teh Pucuk Harum menyegarkan"
12    val txt1 = "Aqua Mineral"
13    val txt2 = "Aqua Mineral"
14    val txt4 = "Aqua"
15    val txt3 = "Teh Pucuk \'Harum\' Menyegarkan! "
```
- Run Console:** Shows the output of the program execution.

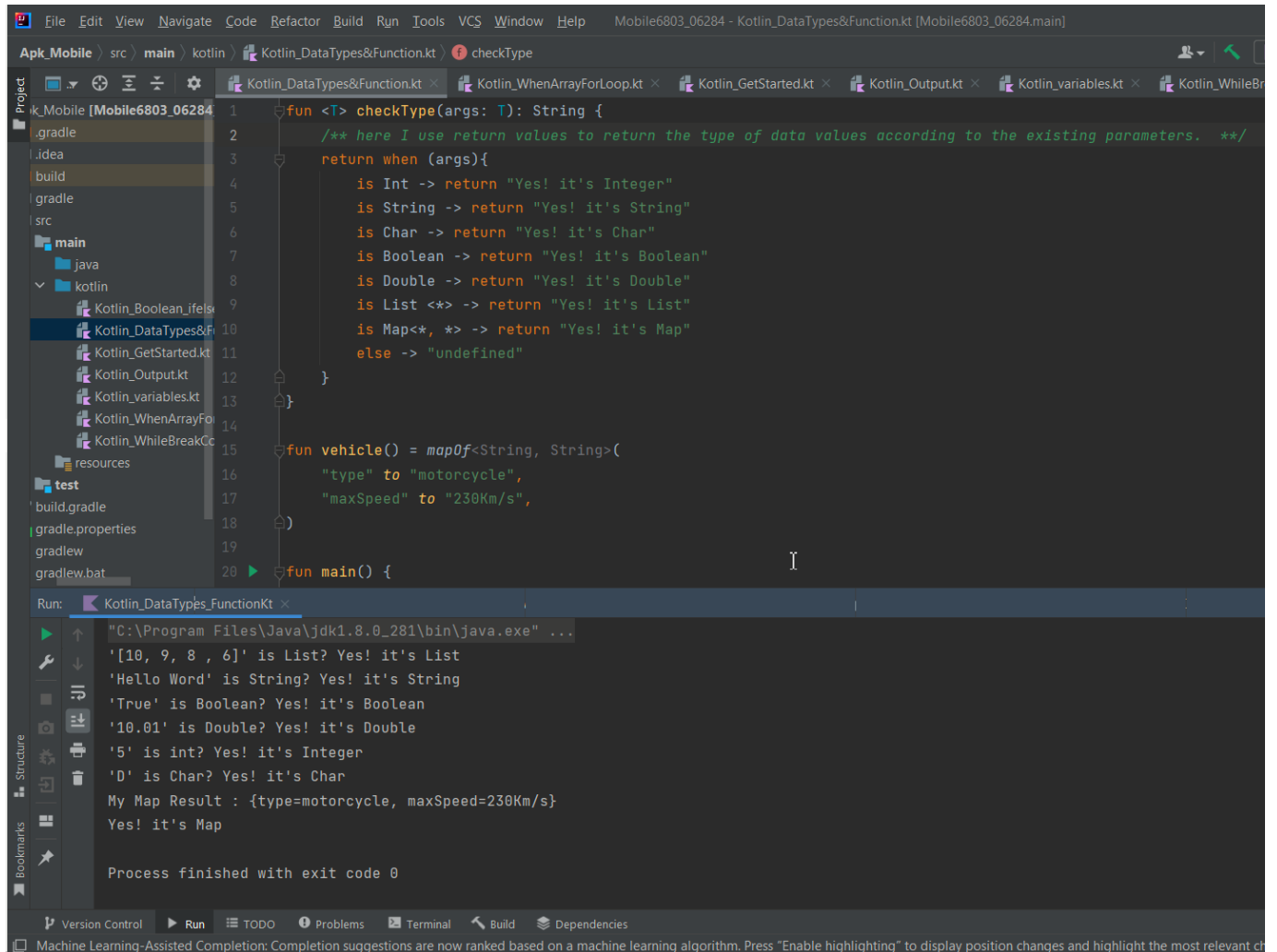
```
Run: Kotlin_variables_StringKt x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
'Variables'
Hello Dewi Putri Ayuningsih
112201906284

'String'
The Length of the txt string is : 27
AQUA MINERAL
aqua mineral
-19
0
11
Hello Aqua Mineral
Hello Aqua
My Name is Hello

Process finished with exit code 0
```
- Bottom Bar:** Contains tabs for `Version Control`, `Run`, `TODO`, `Problems`, `Terminal`, `Build`, and `Dependencies`.

- **Kotlin Data Types and Function**

Data types atau tipe data adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Yang mana jenis dari suatu variable ditentukan dari nilai tipe datanya (Numbers, Characters, Booleans, Strings, Arrays). Disini setiap type data saya jadi kan dalam 1 file dengan menggunakan return values untuk mengembalikan jenis nilai data sesuai dengan parameter yang ada.



```
1 fun <T> checkType(args: T): String {
2     /** here I use return values to return the type of data values according to the existing parameters. **/
3     return when (args){
4         is Int -> return "Yes! it's Integer"
5         is String -> return "Yes! it's String"
6         is Char -> return "Yes! it's Char"
7         is Boolean -> return "Yes! it's Boolean"
8         is Double -> return "Yes! it's Double"
9         is List <*> -> return "Yes! it's List"
10        is Map<*, *> -> return "Yes! it's Map"
11        else -> "undefined"
12    }
13 }
14
15 fun vehicle() = mapOf<String, String>(
16     "type" to "motorcycle",
17     "maxSpeed" to "230Km/s",
18 )
19
20 fun main() {
```

Run: Kotlin\_DataTypes\_FunctionKt x

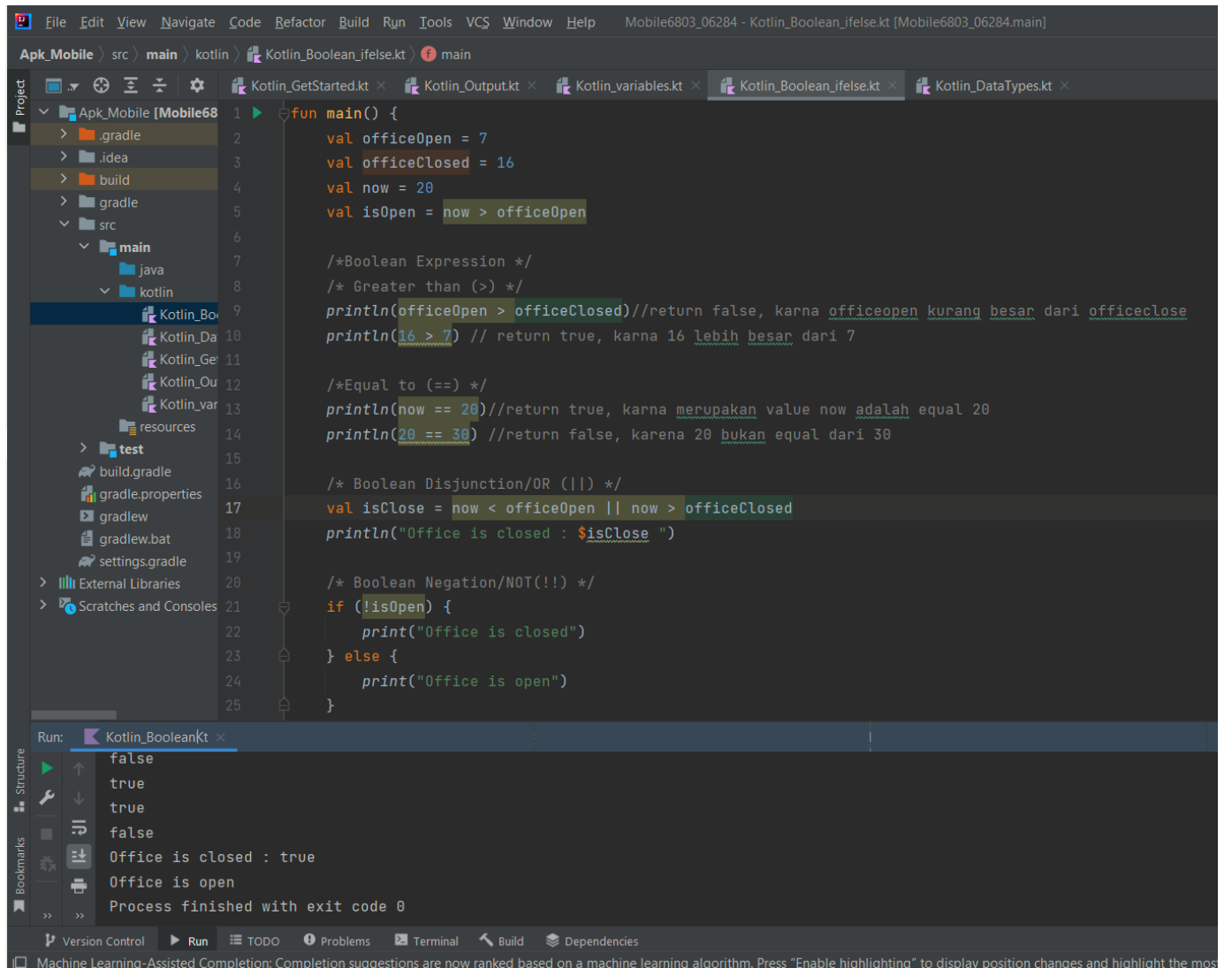
```
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
'[10, 9, 8 , 6]' is List? Yes! it's List
'Hello Word' is String? Yes! it's String
'True' is Boolean? Yes! it's Boolean
'10.01' is Double? Yes! it's Double
'5' is int? Yes! it's Integer
'D' is Char? Yes! it's Char
My Map Result : {type=motorcycle, maxSpeed=230Km/s}
Yes! it's Map

Process finished with exit code 0
```

Machine Learning-Assisted Completion: Completion suggestions are now ranked based on a machine learning algorithm. Press "Enable highlighting" to display position changes and highlight the most relevant ch

- **Kotlin Booleans and Kotlin If...Else**

Boolean adalah sebuah tipe data yang hanya memiliki dua nilai, yaitu **true** dan **false**. Dan kita membuat suatu kondisi dimana akan menjalankan sebuah statement atau expression dalam menghasilkan sebuah hasil dari beberapa kondisi yang diberikan.



```
1 fun main() {
2     val officeOpen = 7
3     val officeClosed = 16
4     val now = 20
5     val isOpen = now > officeOpen
6
7     /*Boolean Expression */
8     /* Greater than (>) */
9     println(officeOpen > officeClosed) //return false, karna officeopen kurang besar dari officeclose
10    println(16 > 7) // return true, karna 16 lebih besar dari 7
11
12    /*Equal to (==) */
13    println(now == 20) //return true, karna merupakan value now adalah equal 20
14    println(20 == 30) //return false, karena 20 bukan equal dari 30
15
16    /* Boolean Disjunction/OR (||) */
17    val isClose = now < officeOpen || now > officeClosed
18    println("Office is closed : $isClose ")
19
20    /* Boolean Negation/NOT(!) */
21    if (!isOpen) {
22        print("Office is closed")
23    } else {
24        print("Office is open")
25    }
26 }
```

Run: Kotlin\_BooleanKt x

```
false
true
true
false
Office is closed : true
Office is open
Process finished with exit code 0
```

Machine Learning-Assisted Completion: Completion suggestions are now ranked based on a machine learning algorithm. Press "Enable highlighting" to display position changes and highlight the most

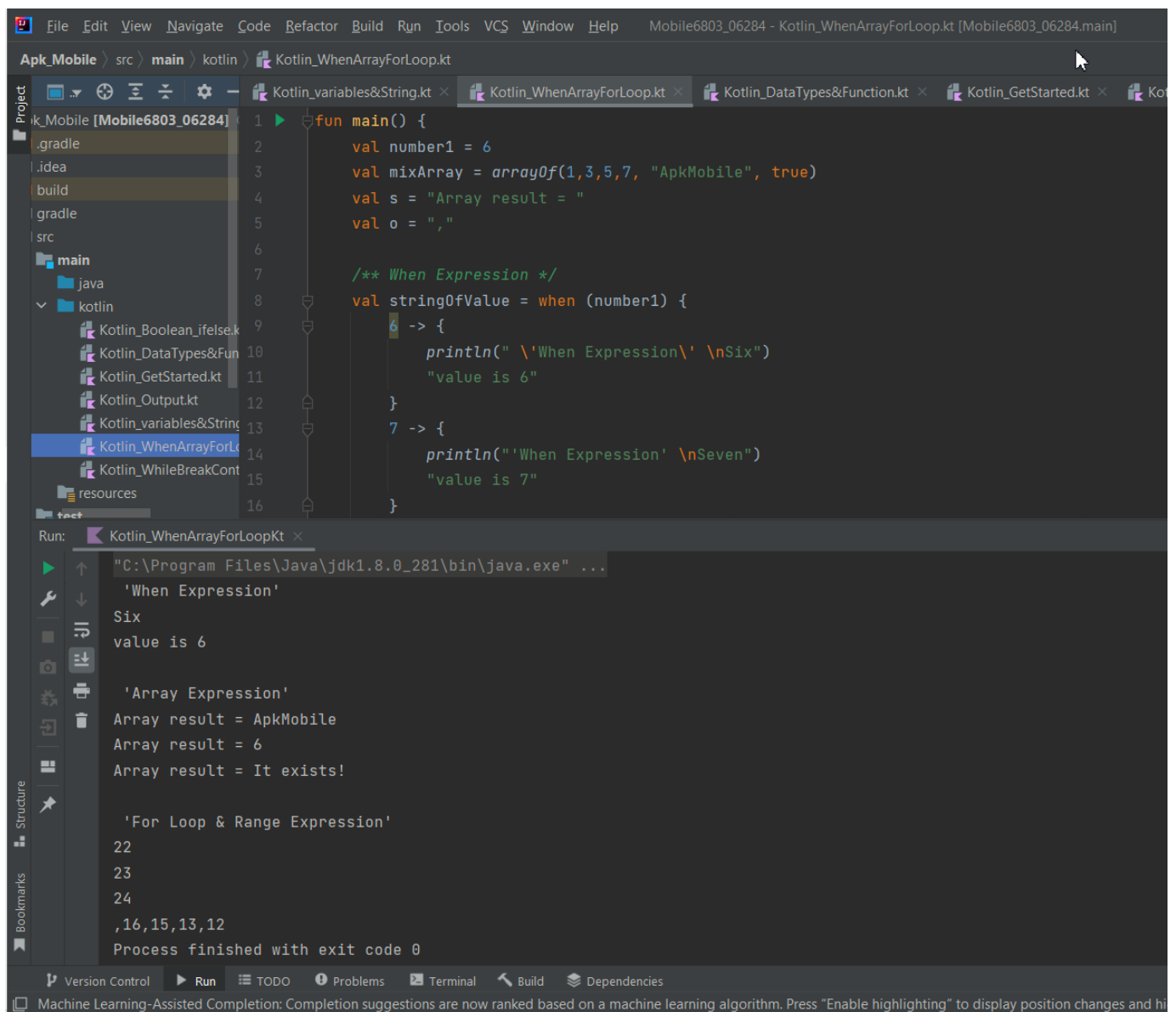
- **Kotlin When, Kotlin Array, Kotlin For Loop, and Kotlin Range**

**Kotlin When** memiliki kata kunci “when” yang kita juga bisa menambahkan *branch* “else” didalamnya.

**Kotlin Array** merupakan sebuah tipe data yang memungkinkan kita untuk menyimpan objek didalam sebuah variable. Yang memiliki fungsi **get** and **set** serta property **size**.

**Kotlin For Loop** sama seperti while and do while, tetapi for merupakan konsep perulangan pada blok yg sama selama hasil evaluasi kondisi yang diberikan terpenuhi/bernilai true. For dapat digunakan pada Ranges, Collections, Arrays and apapun yang menyediakan iterator.

**Kotlin Range** merupakan salah satu tipe unik pada kotlin untuk membuat rentang nilai, dengan operator (“..”, “rangeTo()”, “downTo()”) dan kata kunci “in” pada range digunakan untuk memeriksa apakah suatu nilai masukan pada cakupan nilai range.



The screenshot displays an IDE window with a Kotlin project named 'Apk\_Mobile'. The main editor shows the file 'Kotlin\_WhenArrayForLoop.kt' with the following code:

```
1 fun main() {
2     val number1 = 6
3     val mixArray = arrayOf(1,3,5,7, "ApkMobile", true)
4     val s = "Array result = "
5     val o = ","
6
7     /** When Expression */
8     val stringOfValue = when (number1) {
9         6 -> {
10             println("'When Expression' \nSix")
11             "value is 6"
12         }
13         7 -> {
14             println("'When Expression' \nSeven")
15             "value is 7"
16         }
17     }
18 }
```

The Run console at the bottom shows the execution output:

```
Run: Kotlin_WhenArrayForLoopKt x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
'When Expression'
Six
value is 6

'Array Expression'
Array result = ApkMobile
Array result = 6
Array result = It exists!

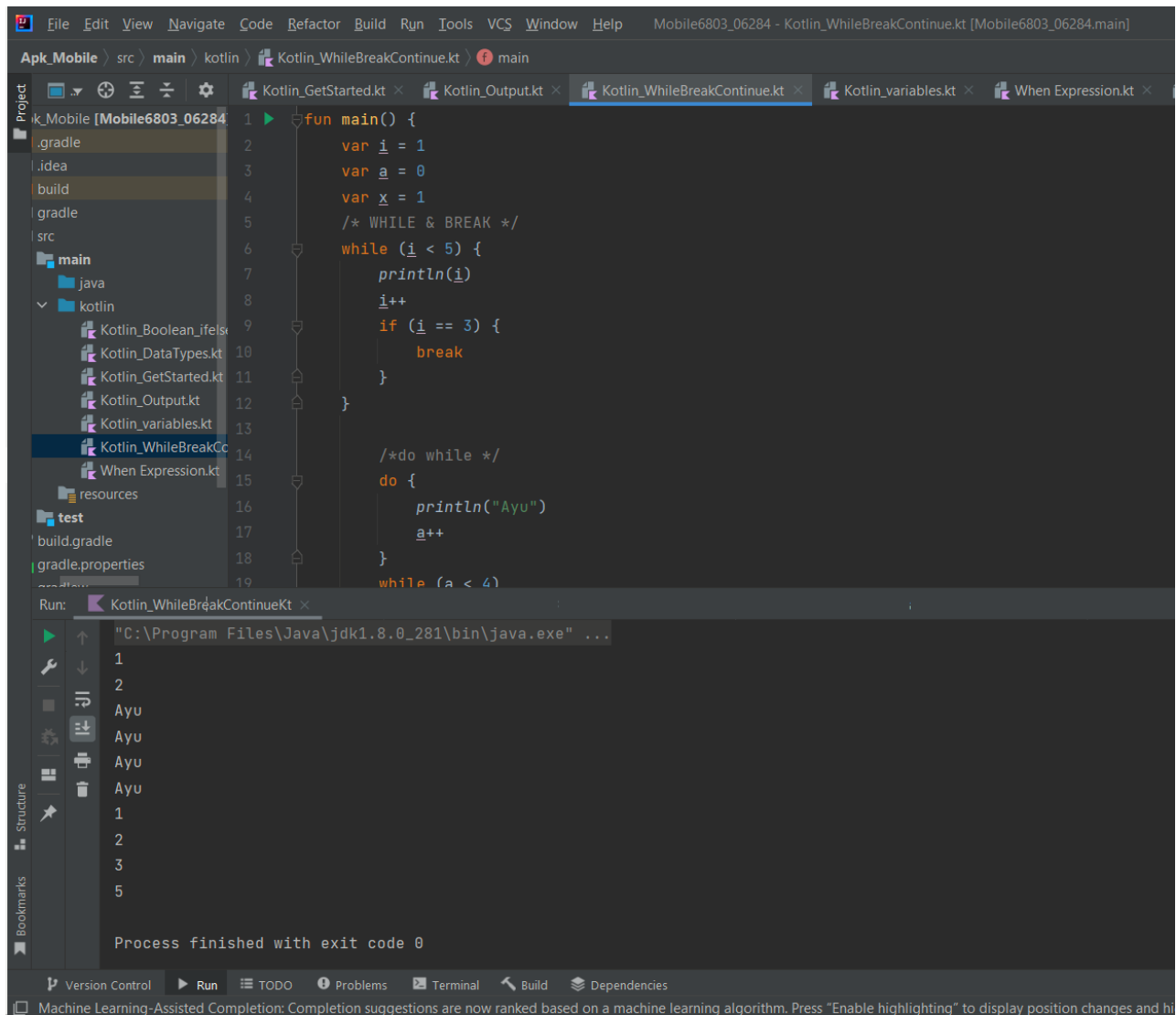
'For Loop & Range Expression'
22
23
24
,16,15,13,12
Process finished with exit code 0
```

The bottom status bar indicates 'Machine Learning-Assisted Completion: Completion suggestions are now ranked based on a machine learning algorithm. Press "Enable highlighting" to display position changes and hi'.

- **Kotlin While and Do While Loop, Kotlin Break and Continue**

**Kotlin While and Do While Loop** adalah proses perulangan blok yang sama tanpa henti sampai kondisi yang diberikan tidak terpenuhi atau bernilai **false**.

**Kotlin Break and Continue** -> **Continue** digunakan untuk melewati proses iterasi dan lanjut dengan proses iterasi berikutnya. Sementara itu, **Break** digunakan untuk menghentikan proses iterasi.



The screenshot shows an IDE with the following Kotlin code in `Kotlin_WhileBreakContinue.kt`:

```
1 fun main() {  
2     var i = 1  
3     var a = 0  
4     var x = 1  
5     /* WHILE & BREAK */  
6     while (i < 5) {  
7         println(i)  
8         i++  
9         if (i == 3) {  
10            break  
11        }  
12    }  
13  
14    /*do while */  
15    do {  
16        println("Ayu")  
17        a++  
18    }  
19    while (a < 4)
```

The run console output shows the execution results:

```
Run: Kotlin_WhileBreakContinueKt x  
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...  
1  
2  
Ayu  
Ayu  
Ayu  
Ayu  
1  
2  
3  
5  
  
Process finished with exit code 0
```

At the bottom, there is a message: "Machine Learning-Assisted Completion: Completion suggestions are now ranked based on a machine learning algorithm. Press 'Enable highlighting' to display position changes and hi