

Pertemuan 2

Perkenalan Kotlin

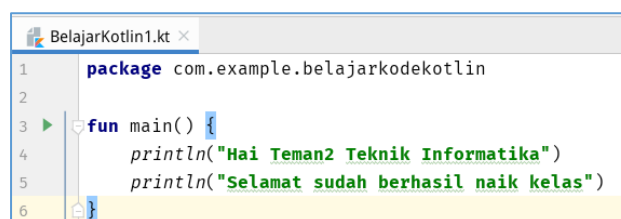
Bahasa pemrograman Kotlin merupakan bahasa pemrograman yang dikeluarkan oleh JetBrains yang mulai dikembangkan pada tahun 2011. Kotlin merupakan nama dari sebuah pulau yang dekat dengan kota St.Petersburg, yang akhirnya dijadikan nama resmi dari bahasa pemrograman ini. Kotlin merupakan bahasa pemrograman yang mengaplikasikan metode „*static typed*“ yang berjalan pada *Java Virtual Machine* (JVM). Kotlin termasuk bahasa pemrograman *open source* yang telah mendukung OOP dan fitur fungsional lainnya. Dimulai pada tahun 2017, Google mengumumkan bahwa Kotlin merupakan Bahasa alternative selain java yang digunakan untuk pengembangan aplikasi Android. Semenjak 2019 Kotlin resmi digunakan sebagai bahasa resmi untuk pengembangan aplikasi android secara luas.

Kotlin dan java dapat digunakan secara bersama-sama dalam sebuah proyek tanpa ada masalah. Jika telah terbiasa mengembangkan aplikasi dengan menggunakan Java pun tidak akan begitu kesulitan untuk menterjemahkan ke dalam bahasa Kotlin. Sebab, bahasa Kotlin jauh lebih mudah dan ringkas jika dibandingkan dengan Java. Meski begitu, bahasa pemrograman Kotlin ini dipengaruhi oleh bahasa pemrograman lain seperti Java, Scala, Groovy, dll. Secara internal, Kotlin masih bergantung pada librari Java untuk memberikan hasil yang diharapkan.

Buat sebuah proyek baru, beri nama BelajarKotlinDasar. Kemudian pilih Empty Activity sebagai layout dasarnya. Setelah proyek selesai terbentuk, buat sebuah file kotlin baru dan beri nama **BelajarKotlin1.kt**. Selain itu kita juga bisa menggunakan kotlin online pada web berikut ini <https://play.kotlinlang.org/>.

1. Tipe Data

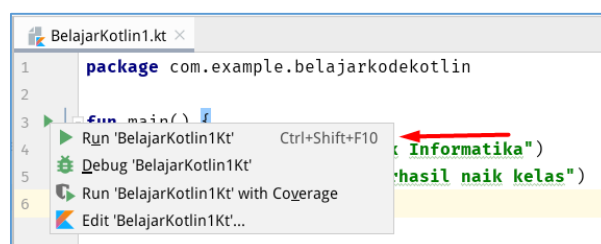
Kemudian silahkan buat sebuah **fungsi main()** didalamnya seperti pada gambar kode dibawah ini. Pada percobaan pertama kita akan mencoba untuk mendeklarasikan tipe data pada Kotlin.



```
1 package com.example.belajarkodekotlin
2
3 fun main() {
4     println("Hai Teman2 Teknik Informatika")
5     println("Selamat sudah berhasil naik kelas")
6 }
```

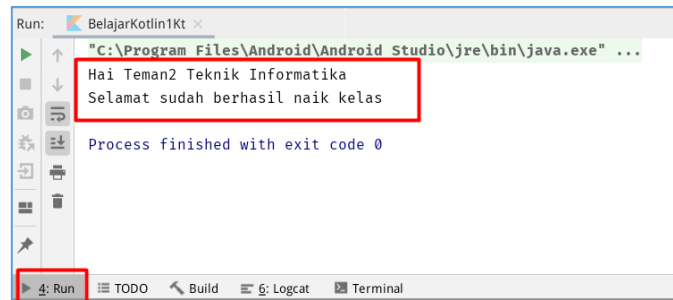
Gambar 1. Program 1

Setelah itu silahkan jalankan kode programnya dengan klik tanda panah hijau seperti gambar dibawah ini.



Gambar 2. Jalankan Aplikasi

Jika berhasil maka aplikasi akan memberikan hasil seperti pada gambar 3 dibawah.



Gambar 3. Hasil Aplikasi

Kemudian silahkan cobakan kode program kotlin dibawah ini.



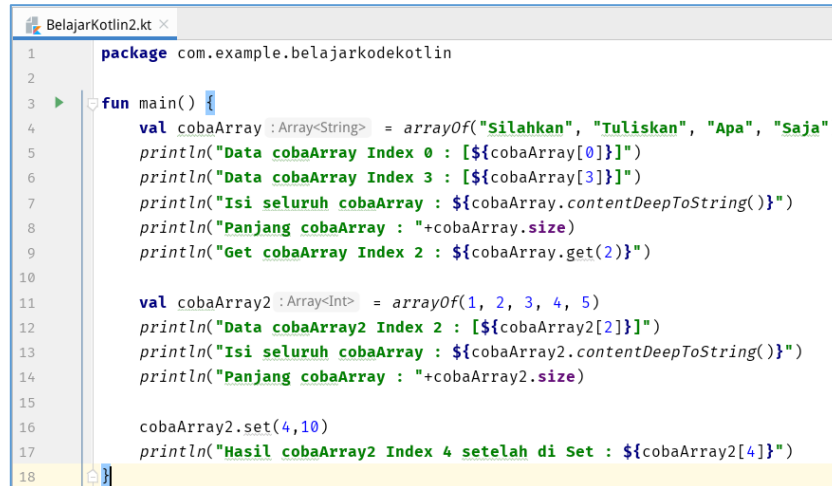
Gambar 4. Program 2 Tipe Data

Tugas Analisa :

1. Jelaskan pada laporan anda, mengapa kita tidak perlu menuliskan tipe data pada variable nilaiInt, nilaiDouble dan nilaiFloat?
2. Jelaskan mengapa kita perlu menuliskan tipe data pada nilaiLong, nilaiShort dan nilaiByte?
3. Jelaskan apa perbedaan penggunaan Val dan Var.
4. Tampilkan hasil dari kode program di atas pada laporan.

2. Array

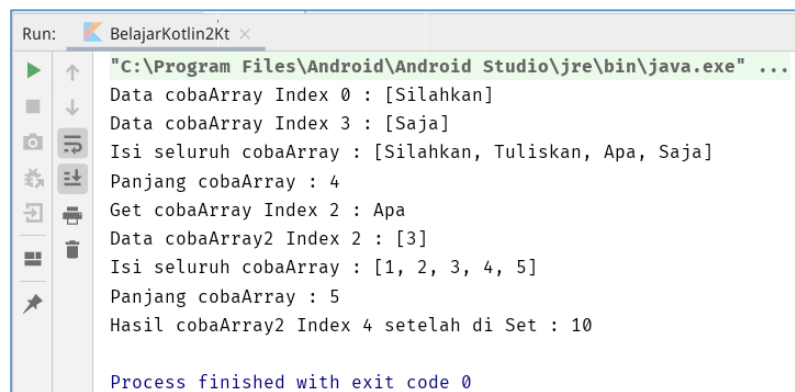
Array di Kotlin mampu menyimpan banyak nilai dari tipe data yang berbeda atau homogen. Akan tetapi jika anda ingin membatasi array untuk sebuah tipe data tertentu juga tidak akan ada masalah. Selanjutnya anda boleh membuat sebuah file kotlin baru bernama **BelajarKotlin2.kt**. Kemudian jangan lupa untuk membuat **fungsi main()** didalamnya agar kode program dapat dijalankan. Silahkan cobakan penulisan Array dengan menggunakan bahasa pemrograman kotlin dibawah ini.



```
1 package com.example.belajarkodekotlin
2
3 fun main() {
4     val cobaArray : Array<String> = arrayOf("Silahkan", "Tuliskan", "Apa", "Saja")
5     println("Data cobaArray Index 0 : [{cobaArray[0]}]")
6     println("Data cobaArray Index 3 : [{cobaArray[3]}]")
7     println("Isi seluruh cobaArray : ${cobaArray.contentDeepToString()}")
8     println("Panjang cobaArray : "+cobaArray.size)
9     println("Get cobaArray Index 2 : ${cobaArray.get(2)}")
10
11     val cobaArray2 : Array<Int> = arrayOf(1, 2, 3, 4, 5)
12     println("Data cobaArray2 Index 2 : [{cobaArray2[2]}]")
13     println("Isi seluruh cobaArray : ${cobaArray2.contentDeepToString()}")
14     println("Panjang cobaArray : "+cobaArray2.size)
15
16     cobaArray2.set(4,10)
17     println("Hasil cobaArray2 Index 4 setelah di Set : ${cobaArray2[4]}")
18 }
```

Gambar 5. Program 3 Array

Jika telah selesai, maka silahkan coba untuk menjalankan kode program diatas. Jika berhasil maka akan menampilkan output seperti pada contoh dibawah.



```
Run: BelajarKotlin2Kt
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Data cobaArray Index 0 : [Silahkan]
Data cobaArray Index 3 : [Saja]
Isi seluruh cobaArray : [Silahkan, Tuliskan, Apa, Saja]
Panjang cobaArray : 4
Get cobaArray Index 2 : Apa
Data cobaArray2 Index 2 : [3]
Isi seluruh cobaArray : [1, 2, 3, 4, 5]
Panjang cobaArray : 5
Hasil cobaArray2 Index 4 setelah di Set : 10
Process finished with exit code 0
```

Gambar 6. Hasil Program 3

Tugas Analisa :

1. Apa tipe data dari cobaArray dan cobaArray2? Jelaskan.
2. Coba tampilkan isi array dengan menggunakan cobaArray.toString(). Apa yang terjadi? Coba berikan penjelasan.
3. Silahkan coba modifikasi cobaArray dengan menambahkan karakter 'A' kedalam array tersebut. Apa yang terjadi? Jelaskan.
4. Silahkan coba gabungkan isi dari cobaArray dan cobaArray2 kedalam sebuah variable baru bernama cobaArray3. Apa yang terjadi? Jelaskan.

3. Collection

Kotlin memiliki dua buah tipe *collection*, *immutable* (tidak dapat diubah) dan *mutable* (dapat diubah). Pada data immutable, maka data seperti list, map dan value menjadi tidak dapat diubah setelah di set di awal dan mutable adalah sebaliknya. Harap untuk memperhatikan jenis koleksi yang anda gunakan karena Kotlin tidak akan menunjukkan informasi dan perbedaan tentang hal tersebut.

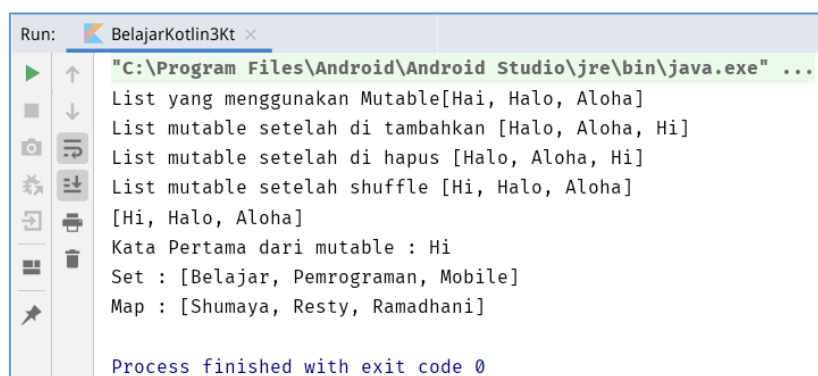
Anda dapat membuat file kotlin baru bernama **BelajarKotlin3.kt**. Kemudian coba ketikkan kode program di bawah ini.



```
1 package com.example.belajarkodekotlin
2
3 fun main() {
4     val kataMutable: MutableList<String> = mutableListOf("Hai", "Halo", "Aloha")
5     println("List yang menggunakan Mutable" + kataMutable)
6
7     kataMutable.add("Hi")
8     kataMutable.removeAt(index: 0)
9
10    println("List mutable setelah di tambahkan " + kataMutable)
11    println("List mutable setelah di hapus " + kataMutable)
12
13    kataMutable.shuffle()
14    println("List mutable setelah shuffle " + kataMutable)
15
16    val kataImmutable: List<String> = kataMutable
17    println(kataImmutable)
18
19    println("Kata Pertama dari mutable : " + kataImmutable.first())
20
21    val cobaSet : Set<String> = setOf("Belajar", "Pemrograman", "Mobile")
22    println("Set : " + cobaSet)
23
24    val cobaMap : Map<Int, String> = mapOf(1 to "Shumaya", 2 to "Resty", 3 to "Ramadhani")
25    println("Map : " + cobaMap.values)
26 }
```

Gambar 7. Program 4 Collection

Setelah anda ketikkan kode program diatas, silahkan dijalankan. Kemudian hasilnya akan seperti di bawah ini.



```
Run: BelajarKotlin3Kt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
List yang menggunakan Mutable[Hai, Halo, Aloha]
List mutable setelah di tambahkan [Halo, Aloha, Hi]
List mutable setelah di hapus [Halo, Aloha, Hi]
List mutable setelah shuffle [Hi, Halo, Aloha]
[Hi, Halo, Aloha]
Kata Pertama dari mutable : Hi
Set : [Belajar, Pemrograman, Mobile]
Map : [Shumaya, Resty, Ramadhani]

Process finished with exit code 0
```

Gambar 8. Hasil Program 4

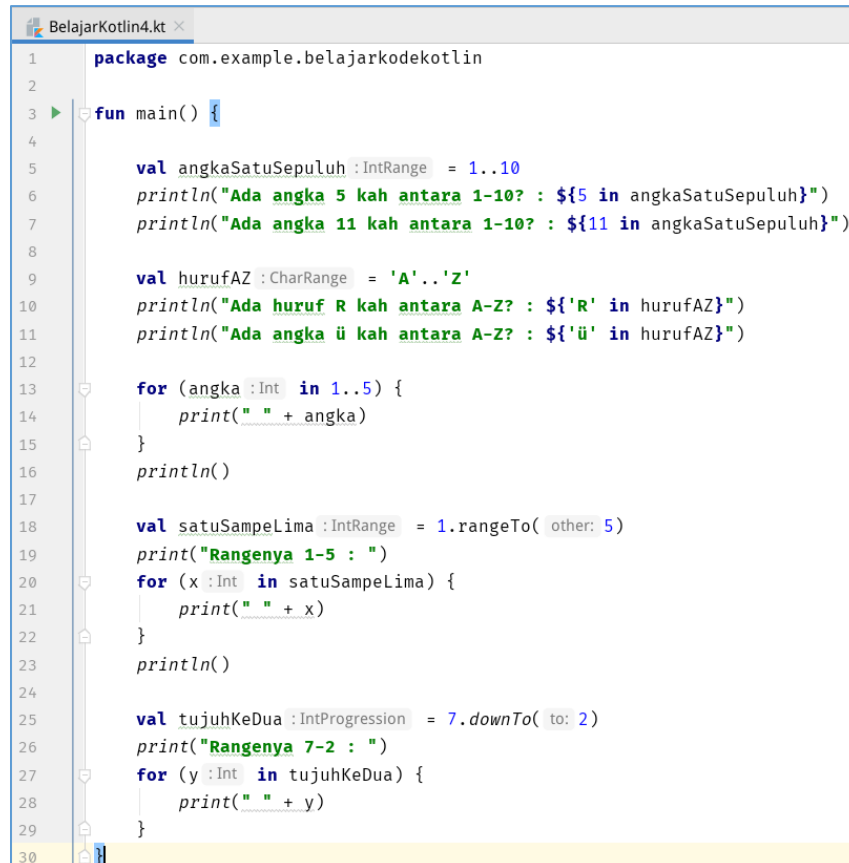
Tugas Analisa :

1. Coba hapus `:MutableList<String>` pada baris kode no. 4. Apakah terjadi error? Mengapa?
2. Apa fungsi `.shuffle()` ?
3. Coba tambahkan kata "Belajar" pada variable `cobaSet` sehingga menjadi ("Belajar", "Pemrograman", "Mobile", "Belajar"). Silahkan jalankan, jelaskan apa yang terjadi.
4. Apa perbedaan List, Set dan Map? Jelaskan berdasarkan contoh kode program diatas.

4. Ranges

Dengan menggunakan Range yang disediakan di Kotlin, kita dapat dengan mudah membuat daftar urutan dengan menentukan nilai awal dan akhir. Misalnya rentang 1..5 akan membuat serangkaian nilai 1, 2, 3, 4, 5. Selain itu, kita juga dapat membuat rentang karakter seperti 'A' .. 'D' yang akan membuat serangkaian nilai A, B, C, D. Kita juga dapat membuat rentang dalam urutan terbalik dan beberapa hal lain dengan rentang.

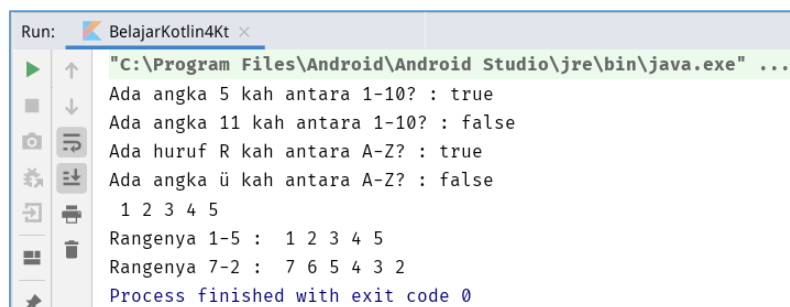
Silahkan buat file kotlin baru bernama **BelajarKotlin4.kt**. Kemudian buat kode program Kotlin di bawah ini.



```
1 package com.example.belajarkodekotlin
2
3 fun main() {
4
5     val angkaSatuSepuluh : IntRange = 1..10
6     println("Ada angka 5 kah antara 1-10? : ${5 in angkaSatuSepuluh}")
7     println("Ada angka 11 kah antara 1-10? : ${11 in angkaSatuSepuluh}")
8
9     val hurufAZ : CharRange = 'A'..'Z'
10    println("Ada huruf R kah antara A-Z? : ${'R' in hurufAZ}")
11    println("Ada angka ü kah antara A-Z? : ${'ü' in hurufAZ}")
12
13    for (angka : Int in 1..5) {
14        print(" " + angka)
15    }
16    println()
17
18    val satuSampelima : IntRange = 1.rangeTo( other: 5)
19    print("Ranganya 1-5 : ")
20    for (x : Int in satuSampelima) {
21        print(" " + x)
22    }
23    println()
24
25    val tujuhKeDua : IntProgression = 7.downTo( to: 2)
26    print("Ranganya 7-2 : ")
27    for (y : Int in tujuhKeDua) {
28        print(" " + y)
29    }
30 }
```

Gambar 9. Program 5 Range

Jika sudah, maka silahkan jalankan kode program tersebut. Maka hasilnya akan seperti tampilan dibawah ini.



```
Run: BelajarKotlin4Kt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Ada angka 5 kah antara 1-10? : true
Ada angka 11 kah antara 1-10? : false
Ada huruf R kah antara A-Z? : true
Ada angka ü kah antara A-Z? : false
 1 2 3 4 5
Ranganya 1-5 :  1 2 3 4 5
Ranganya 7-2 :  7 6 5 4 3 2
Process finished with exit code 0
```

Gambar 10. Hasil Program 5


Perkenalan Kotlin Lanjutan

1. Condition

Dalam setiap bahasa pemrograman pasti membutuhkan pernyataan kondisi untuk mengontrol alur program berdasarkan kondisi yang terjadi. Kita membutuhkan sebuah pernyataan kontrol untuk memeriksa apakah suatu kondisi terpenuhi atau tidak, serta apa konsekuensi yang terjadi atas keduanya.

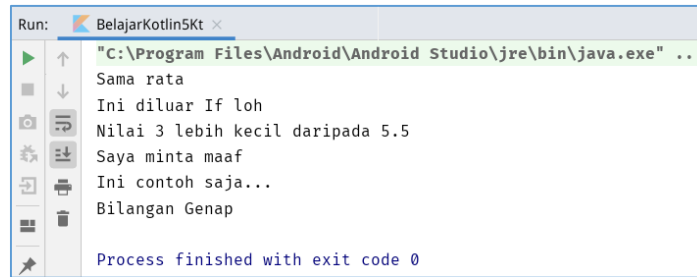
a. If-Else

Pada bahasa pemrograman kotlin, kondisi masih menggunakan ekspresi if-else seperti yang sering kita jumpai pada bahasa pemrograman Java. Silahkan buat sebuah file kotlin baru bernama BelajarKotlin5.kt. Kemudian silahkan coba jalankan kode program dibawah ini.



```
1 package com.example.belajarkodekotlin
2
3 fun main() {
4     val angka = 50
5     if (angka % 2 == 0)
6         println("Sama rata")
7     println("Ini diluar If loh")
8     /*-----*/
9
10    val nilai1 = 3
11    val nilai2 = 5.5
12
13    if (nilai1 < nilai2) {
14        println("Nilai 3 lebih kecil daripada 5.5")
15    } else {
16        println("Coba cek lagi...")
17    }
18    /*-----*/
19
20    val pernyataan = "Marah"
21    if (pernyataan == "Senang") {
22        println("Saya juga ikut senang")
23    } else if (pernyataan == "Sedih") {
24        println("Jangan sedih lagi ya")
25    } else {
26        println("Saya minta maaf")
27    }
28    println("Ini contoh saja...")
29
30
31    val nilai = 10
32    if (nilai < 0)
33        println("Bilangan Negatif")
34    else {
35        if (nilai % 2 == 0)
36            println("Bilangan Genap")
37        else
38            println("Bilangan Ganjil")
39    }
40 }
```

Gambar 11. Program 6 If-Else



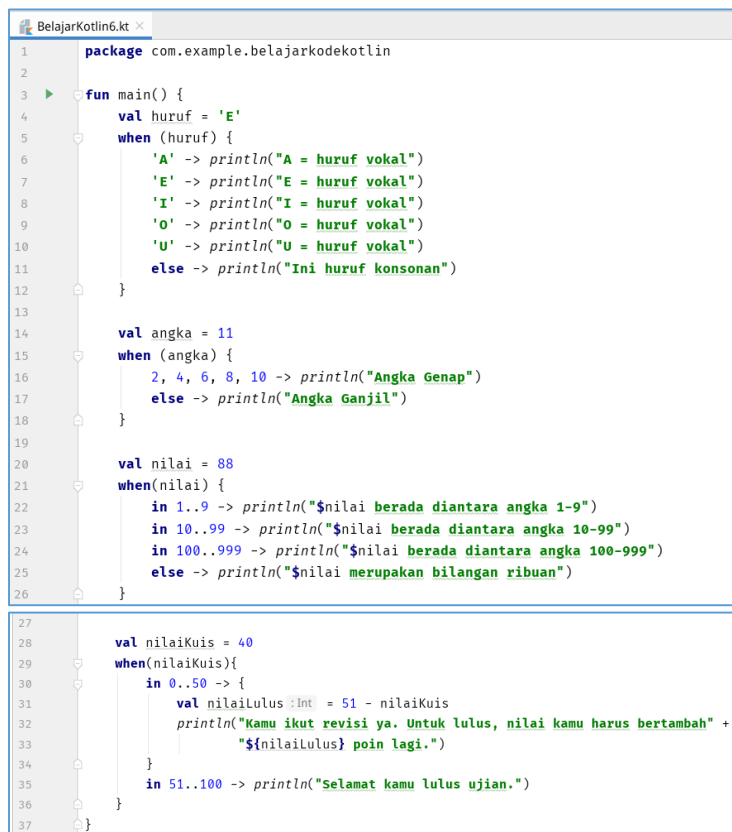
```
Run: BelajarKotlin5Kt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Sama rata
Ini diluar If loh
Nilai 3 lebih kecil daripada 5.5
Saya minta maaf
Ini contoh saja...
Bilangan Genap
Process finished with exit code 0
```

Gambar 12. Hasil Program 6

b. When

When digunakan pada Bahasa pemrograman Kotlin dan sangat mirip penggunaannya seperti switch-case pada Java. Penggunaan fungsi when dapat dituliskan dengan lebih sederhana seperti contoh kode program di bawah ini. Selain itu, when juga dapat dikombinasikan dengan kondisi bersarang maupun penggunaan Range.

Silahkan buat sebuah file kotlin baru bernama BelajarKotlin6.kt. Kemudian cobakan kode program di bawah ini agar anda lebih memahami cara penulisan when pada Bahasa Kotlin.



```
1 package com.example.belajarkodekotlin
2
3 fun main() {
4     val huruf = 'E'
5     when (huruf) {
6         'A' -> println("A = huruf vokal")
7         'E' -> println("E = huruf vokal")
8         'I' -> println("I = huruf vokal")
9         'O' -> println("O = huruf vokal")
10        'U' -> println("U = huruf vokal")
11        else -> println("Ini huruf konsonan")
12    }
13
14    val angka = 11
15    when (angka) {
16        2, 4, 6, 8, 10 -> println("Angka Genap")
17        else -> println("Angka Ganjil")
18    }
19
20    val nilai = 88
21    when (nilai) {
22        in 1..9 -> println("$nilai berada diantara angka 1-9")
23        in 10..99 -> println("$nilai berada diantara angka 10-99")
24        in 100..999 -> println("$nilai berada diantara angka 100-999")
25        else -> println("$nilai merupakan bilangan ribuan")
26    }
27
28    val nilaiKuis = 40
29    when (nilaiKuis) {
30        in 0..50 -> {
31            val nilaiLulus :Int = 51 - nilaiKuis
32            println("Kamu ikut revisi ya. Untuk lulus, nilai kamu harus bertambah" +
33                "${nilaiLulus} poin lagi.")
34        }
35        in 51..100 -> println("Selamat kamu lulus ujian.")
36    }
37 }
```

Gambar 13. Program 7 When

Jika sudah selesai menuliskan kode program diatas, silahkan jalankan. Jika berhasil, maka hasilnya akan seperti pada contoh di bawah ini.



Gambar 14. Hasil Program 7

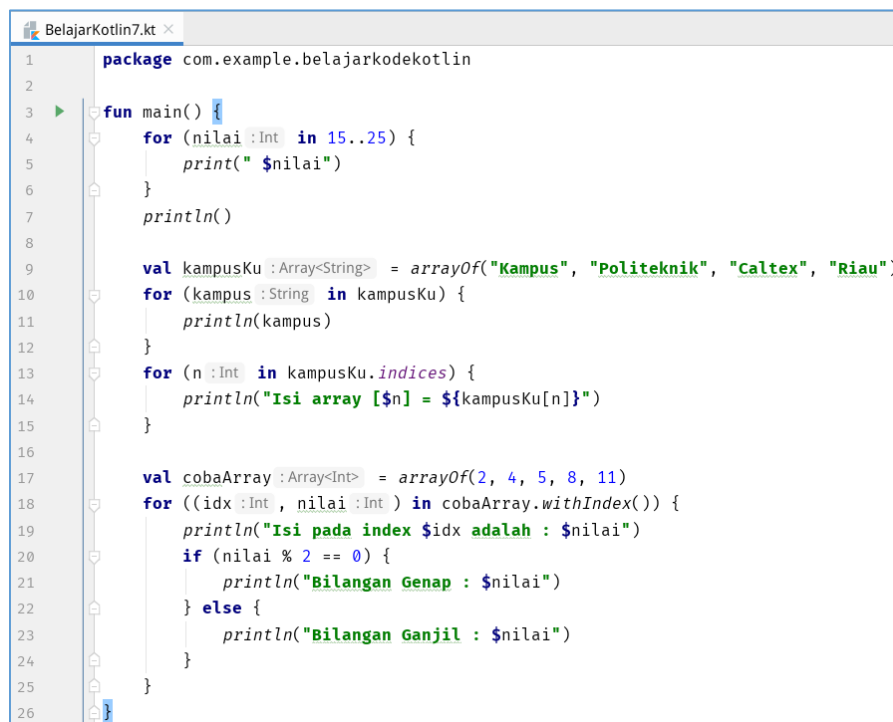
Tugas Analisa :

1. Apa fungsi tanda panah -> pada penulisan kode program when di atas? Apa arti dari tanda panah tersebut?
2. Apakah dalam Bahasa pemrograman Kotlin bisa menggunakan kode switch-case? Silahkan coba, kemudian tampilkan hasil percobaan anda pada poin ini.

2. Looping

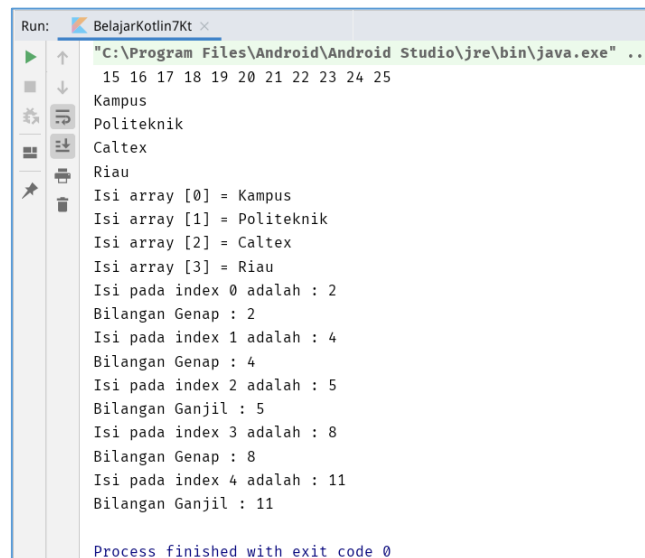
a. For Loop

Perulangan for di Kotlin digunakan untuk melakukan proses iterasi atau menggilir elemen array, rentang, koleksi, dll. Dalam panduan ini, kita akan mempelajari cara menggunakan for loop di Kotlin dengan kombinasi fungsi lainnya. Silahkan buat file kotlin baru bernama **BelajarKotlin7.kt**, kemudian ketikkan kode program kotlin berikut.



Gambar 15. Program 8 For Loop

Jika sudah selesai menuliskan kode program diatas, silahkan jalankan. Jika berhasil, maka hasilnya akan seperti pada contoh di bawah ini.




```
Run: BelajarKotlin7Kt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
15 16 17 18 19 20 21 22 23 24 25
Kampus
Politeknik
Caltex
Riau
Isi array [0] = Kampus
Isi array [1] = Politeknik
Isi array [2] = Caltex
Isi array [3] = Riau
Isi pada index 0 adalah : 2
Bilangan Genap : 2
Isi pada index 1 adalah : 4
Bilangan Genap : 4
Isi pada index 2 adalah : 5
Bilangan Ganjil : 5
Isi pada index 3 adalah : 8
Bilangan Genap : 8
Isi pada index 4 adalah : 11
Bilangan Ganjil : 11
Process finished with exit code 0
```

Gambar 16. Hasil Program 8

b. While Loop dan Do-While Loop

While dan Do-While bekerja dengan cara yang sama seperti bahasa pemrograman lainnya. Satu-satunya perbedaan antara dua loop ini adalah, dalam kasus loop Do-while, kondisinya akan diuji di akhir loop, sementara dengan menggunakan while saja, maka kondisi akan di cek di awal. While loop digunakan untuk mengulang satu blok kode berulang kali selama kondisi yang diberikan mengembalikan nilai true.

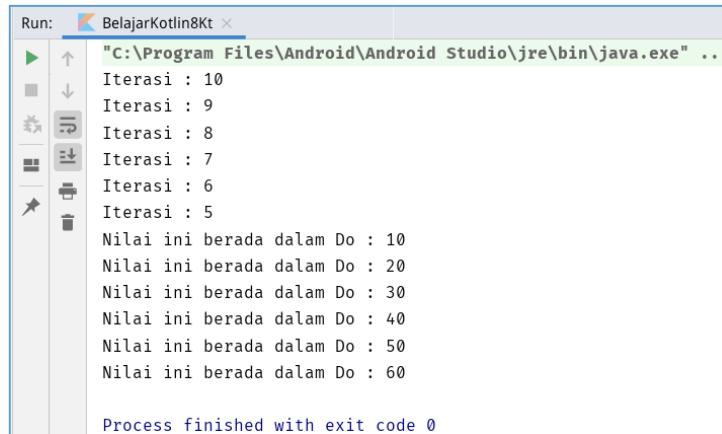
Untuk mencoba kode programnya, silahkan buat sebuah file kotlin bernama **BelajarKotlin8.kt**. Kemudian coba buat kode program dibawah ini.



```
BelajarKotlin8.kt x
1 package com.example.belajarkodekotlin
2
3 fun main() {
4     var angka = 10
5
6     while(angka>=5){
7         println("Iterasi : $angka")
8         angka--
9     }
10
11     var nilai = 0
12     do {
13         nilai = nilai + 10
14         println("Nilai ini berada dalam Do : "+nilai)
15     } while(nilai <= 50)
16 }
```

Gambar 17. Program 9 While dan Do-While

Jika sudah selesai menuliskan kode program diatas, silahkan jalankan. Jika berhasil, maka hasilnya akan seperti pada contoh di bawah ini.



```
Run: BelajarKotlin8Kt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Iterasi : 10
Iterasi : 9
Iterasi : 8
Iterasi : 7
Iterasi : 6
Iterasi : 5
Nilai ini berada dalam Do : 10
Nilai ini berada dalam Do : 20
Nilai ini berada dalam Do : 30
Nilai ini berada dalam Do : 40
Nilai ini berada dalam Do : 50
Nilai ini berada dalam Do : 60
Process finished with exit code 0
```

Gambar 18. Hasil Program 9

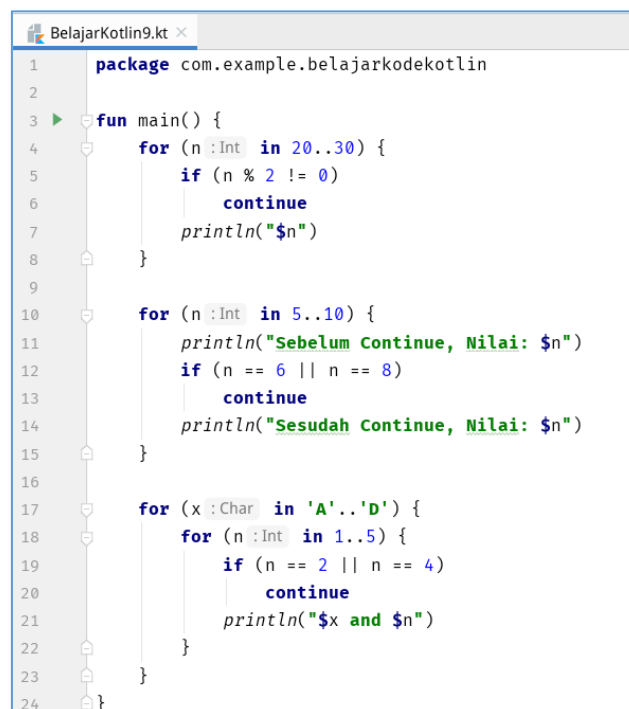
Tugas Analisa :

1. Apa itu infinity loop?
2. Coba berikan satu contoh infinity loop. Tuliskan kode program dan hasilnya. Kemudian jangan lupa untuk menjelaskan mengapa terjadi infinity loop.

3. Break, Continue

Break dan Continue adalah bagian terpenting dari masalah logis. Kata kunci "break" menghentikan aliran pengontrol jika beberapa kondisi gagal dan "lanjutkan" melakukan yang sebaliknya. Konstruksi Continue akan melompati iterasi pengulangan saat terkena kondisi yang ditentukan dan langsung lompat menuju kontrol ke akhir pengulangan untuk iterasi berikutnya. Continue biasanya digunakan dengan ekspresi if else untuk melewati iterasi pada kondisi tertentu.

Untuk mencobanya silahkan buat file BelajarKotlin9.kt, kemudian ketikkan kode program di bawah ini.



```
BelajarKotlin9.kt
1 package com.example.belajarkodekotlin
2
3 fun main() {
4     for (n : Int in 20..30) {
5         if (n % 2 != 0)
6             continue
7         println("$n")
8     }
9
10    for (n : Int in 5..10) {
11        println("Sebelum Continue, Nilai: $n")
12        if (n == 6 || n == 8)
13            continue
14        println("Sesudah Continue, Nilai: $n")
15    }
16
17    for (x : Char in 'A'..'D') {
18        for (n : Int in 1..5) {
19            if (n == 2 || n == 4)
20                continue
21            println("$x and $n")
22        }
23    }
24 }
```

Gambar 19. Program 10 Continue

Jika sudah selesai menuliskan kode program diatas, silahkan jalankan. Jika berhasil, maka hasilnya akan seperti pada contoh di bawah ini.

```

Run: BelajarKotlin9Kt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
20
22
24
26
28
30
Sebelum Continue, Nilai: 5
Sesudah Continue, Nilai: 5
Sebelum Continue, Nilai: 6
Sebelum Continue, Nilai: 7
Sesudah Continue, Nilai: 7
Sebelum Continue, Nilai: 8
Sebelum Continue, Nilai: 9
Sesudah Continue, Nilai: 9
Sebelum Continue, Nilai: 10
Sesudah Continue, Nilai: 10
A and 1
A and 3
A and 5
B and 1
B and 3
B and 5
C and 1
C and 3
C and 5
D and 1
D and 3
D and 5

```

Gambar 20. Hasil Program 10

Pernyataan `break` digunakan untuk menghentikan loop tanpa perlu menyelesaikan proses iterasi pada kondisi loop. Setelah pernyataan `break` ditemukan di dalam sebuah loop, loop segera berhenti tanpa mengeksekusi sisa pernyataan setelah pernyataan `break`. Buat sebuah file kotlin baru lagi. Kemudian buat kode program berikut.

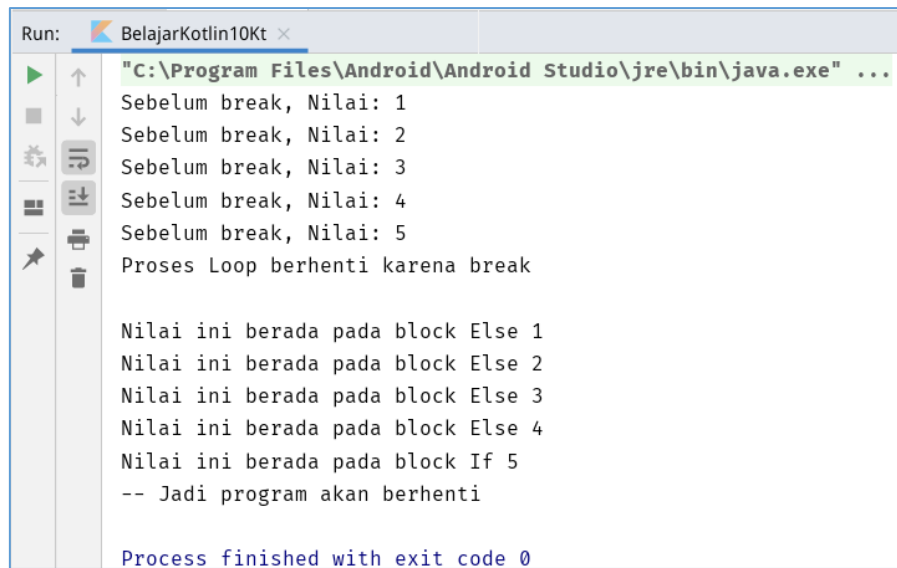
```

1 package com.example.belajarkodekotlin
2
3 fun main() {
4     for (n : Int in 1..10) {
5         println("Sebelum break, Nilai: $n")
6         if (n == 5) {
7             println("Proses Loop berhenti karena break")
8             break
9         }
10    }
11
12    for (ch : Char in 'A'..'C') {
13        for (n : Int in 1..4) {
14            println("$ch and $n")
15            if (n == 2) {
16                break
17            }
18        }
19    }
20
21    println()
22    contohLoop@ for (nilai : Int in 1..10) {
23        if (nilai == 5) {
24            println("Nilai ini berada pada block If " + nilai + "\n-- Jadi program akan berhenti")
25            break@contohLoop
26        } else {
27            println("Nilai ini berada pada block Else " + nilai)
28            continue@contohLoop
29        }
30    }
31 }

```

Gambar 21. Program 11 Break

Jika sudah selesai, maka silahkan coba jalankan kode program tersebut.



```
Run: BelajarKotlin10Kt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Sebelum break, Nilai: 1
Sebelum break, Nilai: 2
Sebelum break, Nilai: 3
Sebelum break, Nilai: 4
Sebelum break, Nilai: 5
Proses Loop berhenti karena break

Nilai ini berada pada block Else 1
Nilai ini berada pada block Else 2
Nilai ini berada pada block Else 3
Nilai ini berada pada block Else 4
Nilai ini berada pada block If 5
-- Jadi program akan berhenti

Process finished with exit code 0
```

Gambar 22. Hasil Program 11

Tugas Analisa :

1. Kapan sebaiknya kita mengimplementasikan Break dan Continue?
2. Apa maksud label contohLoop@ pada baris kode 21 pada Gambar 22? Jelaskan.