

Setoran:

File Laporan Pengamatan: PAM_LaporanPraktikum2_{NIM}.pdf

Source Code: PAM_SourceCode2_{NIM}.zip

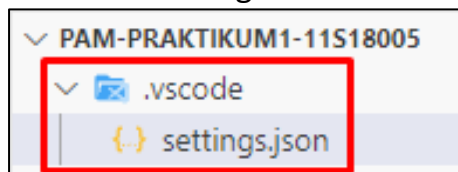
Persiapan Praktikum:

1. Workspace

Buat workspace baru dengan format penamaan "pam-praktikum2-{nim}".

2. Menyembunyikan File di VSCode

Untuk menghilangkan file hasil compile Kotlin, silahkan membuat folder baru dengan nama ".vscode" di dalam folder tersebut buat file baru dengan nama "settings.json", seperti berikut:



Modifikasi isi file ".vscode/settings.json", seperti berikut:

```
settings.json
1  {
2    "files.exclude": {
3      "*.class": true,
4      "*.jar": true,
5      "**/META-INF": true,
6    },
7  }
```

Latihan Praktikum:

Praktikan, amati, pahami, dan jawab semua pertanyaan yang ada serta laporkan hasilnya pada laporan praktikum.

A. Dasar Kotlin 2

1. Operator

Modifikasi isi file “Operator.kt”, seperti berikut:

```
Operator.kt
1 fun main(){
2     val nilai1 = 20
3     val nilai2 = 4
4
5     // Operator aritmatika
6     // -- penambahan
7     val hasil1 = nilai1 + nilai2
8     // -- pengurangan
9     val hasil2 = nilai1 - nilai2
10    // -- perkalian
11    val hasil3 = nilai1 * nilai2
12    // -- pembagian
13    val hasil4 = nilai1 / nilai2
14    // -- sisa bagi
15    val hasil5 = nilai1 % nilai2
16
17    println(hasil1)
18    println(hasil2)
19    println(hasil3)
20    println(hasil4)
21    println(hasil5)
22
23    // Operator penugasan
24    var x = 0;
25    // -- inisialisasi nilai
26    x = 5;
27    // -- penugasan dengan operasi penambahan
28    x += 1;
29    // -- penugasan dengan operasi pengurangan
30    x -= 1;
31    // -- penugasan dengan operasi perkalian
32    x *= 2;
33    // -- penugasan dengan operasi pembagian
34    x /= 10;
35
36    println(x)
37
38    // Operator perbandingan
39    val a = 5
```

Operator.kt

```
40 val b = 5
41 // -- sama dengan
42 println(a == b)
43 // -- tidak sama dengan
44 println(a != b)
45 // -- lebih besar dari
46 println(a > b)
47 // -- lebih besar dari atau sama dengan
48 println(a >= b)
49 // -- lebih kecil dari
50 println(a < b)
51 // -- lebih kecil dari atau sama dengan
52 println(a <= b)
53
54 // Operator logika
55 val kondisi = arrayOf(true, true, false, false)
56 // -- logika and
57 println(kondisi[0] && kondisi[1])
58 println(kondisi[0] && kondisi[2])
59 println(kondisi[2] && kondisi[1])
60 println(kondisi[2] && kondisi[3])
61 // -- logika or
62 println(kondisi[0] || kondisi[1])
63 println(kondisi[0] || kondisi[2])
64 println(kondisi[2] || kondisi[1])
65 println(kondisi[2] || kondisi[3])
66 // -- logika not
67 println(!kondisi[0])
68 println(!kondisi[1])
69 println(!kondisi[2])
70 println(!kondisi[3])
71
72 // Operator elvis
73 val think: String? = null
74 val hasil = think ?: "Sedang berpikir!"
75 println(hasil)
76
77 // Operator increment
78 var z = 10
79 // -- increment
80 var y = z
81 println(y++)
82 y = z
83 println(++y)
84 // -- decrement
85 y = z
86 println(y--)
87 y = z
88 println(--y)
```

Operator.kt

```
89 }
```

Rangkum pemahaman yang kamu peroleh dari mempraktikkan kode program di atas. Silahkan melakukan compile & run program secara mandiri.

2. Decision Making

Modifikasi isi file “DecisionMaking.kt”, seperti berikut:

DecisionMaking.kt

```
1 fun main(){
2     // single if statement
3     val x = 10
4     if (x > 5) {
5         println("x lebih besar dari 5")
6     }
7
8     // if-else statement
9     val y = 8
10    if (y % 2 == 0) {
11        println("y adalah bilangan genap")
12    } else {
13        println("y adalah bilangan ganjil")
14    }
15
16    // if-else if-else
17    val nilai = 75
18    if (nilai >= 90) {
19        println("A")
20    } else if (nilai >= 80) {
21        println("B")
22    } else if (nilai >= 70) {
23        println("C")
24    } else {
25        println("D")
26    }
27
28    // ekspresi when (mirip switch pada bahasa pemrograman lainnya)
29    val hari = 3
30    when (hari) {
31        1 -> println("Senin")
32        2 -> println("Selasa")
33        3 -> println("Rabu")
34        4 -> println("Kamis")
35        5 -> println("Jumat")
36        6 -> println("Sabtu")
37        7 -> println("Minggu")
38        else -> println("Hari tidak valid")
39    }
40
41    // ekspresi when dengan kondisi
```

DecisionMaking.kt

```
42 val suhu = 25
43 when {
44     suhu < 0 -> println("Sangat dingin")
45     suhu in 0..10 -> println("Dingin")
46     suhu in 11..20 -> println("Sejuk")
47     suhu in 21..30 -> println("Hangat")
48     else -> println("Panas")
49 }
50
51 // operator ternary
52 val angka = 7
53 val hasil = if (angka % 2 == 0) "Genap" else "Ganjil"
54 println(hasil)
55 }
```

Rangkum pemahaman yang kamu peroleh dari mempraktikkan kode program di atas. Silahkan melakukan compile & run program secara mandiri.

3. Looping

Modifikasi isi file “Looping.kt”, seperti berikut:

Looping.kt

```
1 fun main(){
2     // Perulangan for dengan rentang angka
3     for (i in 1..5) {
4         println("Iterasi ke-$i")
5     }
6     println()
7
8     // Perulangan for dengan koleksi
9     val fruits = listOf("Apel", "Jeruk", "Pisang")
10    for (fruit in fruits) {
11        println(fruit)
12    }
13    println()
14
15    // Perulangan while
16    var counter = 0
17    while (counter < 5) {
18        println("Iterasi ke-$counter")
19        counter++
20    }
21    println()
22
23    // Perulangan do-while
24    var x = 0
25    do {
26        println("Iterasi ke-$x")
27        x++
28    } while (x < 5)
```

Looping.kt

```
29     println()
30
31     // Penggunaan break untuk keluar dari loop
32     for (i in 1..10) {
33         if (i == 5) {
34             break
35         }
36         println("Iterasi ke-$i")
37     }
38     println()
39
40     // Penggunaan continue untuk melanjutkan ke iterasi berikutnya
41     for (i in 1..5) {
42         if (i == 3) {
43             continue
44         }
45         println("Iterasi ke-$i")
46     }
47     println()
48
49     // penggunaan forEach pada List
50     val numbers = listOf(1, 2, 3, 4, 5)
51     println("forEach pada List:")
52     numbers.forEach { println(it) }
53     println()
54
55     // penggunaan map pada List untuk transformasi
56     val squaredNumbers = numbers.map { it * it }
57     println("map pada List untuk transformasi:")
58     squaredNumbers.forEach { println(it) }
59     println()
60
61     // penggunaan filter pada List
62     val evenNumbers = numbers.filter { it % 2 == 0 }
63     println("filter pada List:")
64     evenNumbers.forEach { println(it) }
65     println()
66
67     // penggunaan forEachIndexed pada List
68     println("forEachIndexed pada List:")
69     numbers.forEachIndexed { index, value -> println("Index: $index,
Value: $value") }
70     println()
71
72     // penggunaan forEach pada Map
73     val personMap = mapOf("nama" to "Abdullah", "umur" to 24, "kota" to
"Sitoluaman")
74     println("forEach pada Map:")
75     personMap.forEach { key, value -> println("$key: $value") }
```

Looping.kt

```
76     println()
77
78     // penggunaan mapValues pada Map
79     val ageMap = mapOf("Abdullah" to 24, "Kevin" to 22, "Eduward" to 23)
80     val ageStringMap = ageMap.mapValues { (_, age) -> "Umur: $age tahun"
81 }
82     println("mapValues pada Map:")
83     ageStringMap.forEach { println(it.value) }
84     println()
85
86     // penggunaan flatMap pada List
87     val words = listOf("Belajar", "Bahasa", "Kotlin")
88     val flattenedChars = words.flatMap { it.toList() }
89     println("flatMap pada List:")
90     flattenedChars.forEach { println(it) }
91     println()
92
93     // penggunaan return dalam lambda expression
94     println("penggunaan return dalam lambda expression:")
95     val numbers2 = listOf(1, 2, 3, 4, 5)
96     numbers2.forEach {
97         if (it == 3) {
98             return@forEach
99         }
100         println("Angka: $it")
101     }
102     println()
103
104     // penggunaan repeat function
105     println("penggunaan repeat function:")
106     repeat(3) {
107         println("Hai!")
108     }
109     println()
110
111     // memanggil fungsi hitungFaktorial()
112     println(hitungFaktorial(5))
113 }
114
115 // Rekursif
116 fun hitungFaktorial(n: Int): Int {
117     return if (n == 0 || n == 1) {
118         1
119     } else {
120         n * hitungFaktorial(n - 1)
121     }
122 }
```

Rangkum pemahaman yang kamu peroleh dari mempraktikkan kode program di atas. Silahkan melakukan compile & run program secara mandiri.

4. Function

Modifikasi isi file “Function.kt”, seperti berikut:

```
Function.kt
1  fun main(){
2      // Memanggil fungsi tanpa parameter dan tanpa nilai kembalian
3      sapa()
4      sapaKembali()
5
6      // Memanggil fungsi dengan parameter
7      sapaNama("Abdullah")
8
9      // Memanggil fungsi dengan nilai kembalian
10     val nilaiPI = getPI()
11     println("Nilai PI: $nilaiPI")
12
13     // Memanggil fungsi dengan parameter dan nilai kembalian
14     val hasilKali = kali(2, 4)
15     println("Hasil perkalian 2 dengan 4: $hasilKali")
16     val hasilTambah = tambah(2, 4)
17     println("Hasil penjumlahan 2 dengan 4: $hasilTambah")
18 }
19
20 // Fungsi tanpa parameter dan tanpa nilai kembalian
21 fun sapa() {
22     println("Halo!")
23 }
24
25 // Fungsi dengan parameter
26 fun sapaNama(nama: String) {
27     println("Halo, $nama!")
28 }
29
30 // Fungsi dengan nilai kembalian
31 fun getPI(): Double {
32     return 3.14159265359
33 }
34
35 // Fungsi dengan parameter dan nilai kembalian
36 fun kali(a: Int, b: Int): Int {
37     return a * b
38 }
39
40 // Fungsi tanpa parameter dan tanpa nilai kembalian dengan ekspresi
41 fun sapaKembali() = println("Hay!")
42
43 // Fungsi dengan nilai kembalian menggunakan ekspresi
```


Function.kt

```
44 fun tambah(a: Int, b: Int) = a + b
```

Rangkum pemahaman yang kamu peroleh dari mempraktikkan kode program di atas. Silahkan melakukan compile & run program secara mandiri.

Tantangan Praktikum:

Tantangan praktikum merupakan serangkaian studi kasus terkait beberapa permasalahan yang perlu untuk kamu selesaikan dengan memanfaatkan pemahaman selama mempraktikkan latihan praktikum.

1. Format Jam Valid

isJamValid(j, m, d) adalah fungsi untuk melakukan pemeriksaan jika 3 bilangan integer j, m, d menyusun format jam yang valid. Definisi jam yang valid adalah jika elemen jam (j) bernilai antara 0 sampai 23, elemen menit (m) bernilai antara 0 sampai 59, dan elemen detik (d) bernilai antara 0 dan 59.

Modifikasi isi file “Tantangan1.kt”, seperti berikut:

Tantangan1.kt

```
fun isJamValid(j: Int, m: Int, d: Int) = // TODO: tuliskan kode kamu disini

fun main(){
    val j: Int? = readLine()?.toIntOrNull()
    val m: Int? = readLine()?.toIntOrNull()
    val d: Int? = readLine()?.toIntOrNull()

    if (j != null && m != null && d != null) {
        println(isJamValid(j, m, d))
    }
}
```

Hasil yang diharapkan sebagai berikut:

Input	Output
2 3 2	true
-1 100 2	false

2. Konversi Detik ke String Jam

detikToFormatJam(x) adalah fungsi untuk mengubah bilangan integer (x) sebagai detik, menjadi string jam dengan format "j:m:d".

Modifikasi isi file "Tantangan2.kt", seperti berikut:

Tantangan2.kt

```
fun detikToFormatJam(x: Int): String {  
    // TODO: tuliskan kode kamu disini  
}  
  
fun main(){  
    val x: Int? = readLine()?.toIntOrNull()  
  
    if (x != null) {  
        println(detikToFormatJam(x))  
    }  
}
```

Hasil yang diharapkan sebagai berikut:

Input	Output
1	0:0:1
61	0:1:1
10101	2:48:21

3. Konversi String Jam ke Detik

formatJamToDetik(format) adalah fungsi untuk mengubah string (format) sebagai format jam, menjadi bilangan bulat sebagai detik.

Modifikasi isi file "Tantangan3.kt", seperti berikut:

Tantangan3.kt

```
fun formatJamToDetik(format: String): Int {  
    // TODO: tuliskan kode kamu disini  
}  
  
fun main(){  
    val format: String? = readLine()  
  
    if (format != null) {  
        println(formatJamToDetik(format))  
    }  
}
```

Hasil yang diharapkan sebagai berikut:

Input	Output
0:0:1	1
0:1:1	61
2:48:21	10101

4. Dia ABC

diaABC(s, n) adalah fungsi untuk menampilkan pesan tersembunyi jika bilangan habis dibagi 3 maka tampilkan "I", jika bilangan habis di bagi 5 maka tampilkan "U", jika bilangan habis di bagi 7 maka tampilkan "E", apabila terdapat bilangan yang habis di bagi 3, 5 dan 7 kemudian habis dibagi 2 maka tampilkan ABC jika tidak tampilkan CBA. Parameter s merupakan nilai awal dan n merupakan nilai akhir yang harus dilakukan looping dimulai dari angka s sampai n.

Modifikasi isi file "Tantangan4.kt", seperti berikut:

Tantangan4.kt

```
fun diaABC(s: Int, n: Int) {  
    // TODO: tuliskan kode kamu disini  
}  
  
fun main(){  
    val s: Int? = readLine()?.toIntOrNull()  
    val n: Int? = readLine()?.toIntOrNull()  
  
    if (s != null && n != null) {  
        diaABC(s, n)  
    }  
}
```

Hasil yang diharapkan sebagai berikut:

Input	Output
100	U
110	101 I 103 104 CBA 106 107 I 109 U