TUGAS INDIVIDU KOTLIN

RASYID RAMADHANI
MOBILE MORNING

APRIL 2023

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

KOTLIN FUNDAMENTAL

1.	Data Types

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // String
    val firstWord = "Infinite"
    val secondWord = "Learning"
    println("$firstWord $secondWord") //--> Infinite Learning

    // Char
    println("\nChar Section\t\t: ")
    var grade = 'A'
    println("Grade " + grade++)      // B
    println("Grade " + grade++)      // C
    println("Grade " + grade++)      // D
    println("Grade " + grade--)      // C
    println("Grade " + grade--)      // B
    println("Grade " + grade--)      // A

    // String with Index
    println("\nString with Index\t: ")
    val indexString = "Kotlin Language"
    val firstChar = indexString [0]
    println("Karakter Pertama dari $indexString adalah $firstChar")

    // String with Excaped String
    println("\nString with Excaped String\t: ")
    val escapedString = "Kampus Merdeka \nby \"Infininte Learning\""
    println(escapedString)

    // String with Raw String
    println("\nString with Raw String\t: ")
    val rawString = """
    Belajar Bahasa Kotlin
    --------------------
    Di Infinite Learning
    Bersama Kampus Merdeka
    """
    println(rawString)

    // Boolean
    println("\nBoolean\t: ")
    var x = true
    var y = false
    println("x && y = " + (x && y)) // --> false
    println("x || y = " + (x || y)) // --> true
    println("!y = " + (!y)) // --> true

    // Boolean Expression
    println("\nBoolean Expression\t: ")
    val a: Int = 50
    val b: Int = 25
    println("a > b = " + (a > b))
    println("a < b = " + (a < b))
    println("a >= b = " + (a >= b))
    println("a <= b = " + (a <= b))
```

1

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

```kotlin
58      println("a >= b = " + (a >= b))
59      println("a <= b = " + (a <= b))
60      println("a == b = " + (a == b))
61      println("a != b = " + (a != b))
62
63  // Boolean Function
64      println("\nBoolean Function\t: ")
65      val e: Boolean = true
66      val f: Boolean = false
67      val g: Boolean = true
68      println("e.and (f) = " + e.and(f))
69      println("e.or(f) = " + e.or(f))
70      println("e.and(g) = " + e.and(g))
71
72  // Numbers
73      println("\nNumbers\t\t: ")
74      val h: Int = 10000
75      val i: Double = 100.00
76      val j: Float = 100.00f
77      val k: Long = 1000000004L
78      val l: Short = 10
79      val m: Byte = Byte.MAX_VALUE
80
81      println("Int Value is " + h)
82      println("Double Value is " + i)
83      println("Float Value is " + j)
84      println("Long Value is " + k)
85      println("Short Value is " + l)
86      println("Byte Value is " + m)
88  // Array
89      println("\nArray\t\t: ")
90      val groups = arrayOf<String>("Group 1", "Group 2", "Group 3", "Group 4")
91
92  // Get Value
93      groups[0]
94      groups.get(0)
95
96  // Set Value
97      groups [0]= "Group 1"
98      groups.set(1, "Group Dua")
99
100 // Array with Get & Set
101     println("\nArray with Get & Set\t\t: ")
102     val group = arrayOf<String>("Group 1", "Group 2", "Group 3", "Group 4")
103     println(group [0] )
104     println(group.get(1) )
105     println("_____")
106
107     group [0] = "Group Satu"
108     group.set(1, "Group Dua")
109
110     println(group[0] )
111     println(group.get(1))
112     println(group.get(2))
114 // Array with Primitive type Arrays
115     println("\nPrimitive Type Arrays\t\t: ")
116     val groupie = intArrayOf(1, 2, 3, 4)
117     val intArray = intArrayOf(2, 9, 11, 15)     // [2, 9, 11, 15]
118     intArray[2] = 30                            // [2, 9, 30, 15]
119     print(intArray[2])
120 }
```

Output

Nama   : Rasyid Ramadhani

Sesi Kelas : Morning

Program   : Android Mobile Dev

```
Infinite Learning

Char Section       :
Grade A
Grade B
Grade C
Grade D
Grade C
Grade B

String with Index   :
Karakter Pertama dari Kotlin Language adalah K

String with Excaped String  :
Kampus Merdeka
by "Infininte Learning"

String with Raw String  :

    Belajar Bahasa Kotlin

    ---------------------
    Di Infinite Learning
    Bersama Kampus Merdeka

Boolean :
x && y = false
x || y = true
!y = true

Boolean Expression  :
a > b = true
a < b = false
a >= b = true
a <= b = false
a == b = false
a != b = true

Boolean Function    :
e.and (f) = false
e.or(f) = true
e.and(g) = true

Numbers     :
Int Value is 10000
Double Value is 100.0
Float Value is 100.0
Long Value is 1000000004
Short Value is 10
Byte Value is 127
Array       :

Array with Get & Set        :
Group 1
Group 2
---------------
Group Satu
Group Dua
Group 3

Primitive Type Arrays       :
30
Process finished with exit code 0
```

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

2.	Functions

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val hasil = setUser( name: "Rasyid Ramadhani", age: 21)
    println(hasil) // --> Nama kamu adalah Rasyid Ramadhani, dan umur kamu 21 tahun
}
fun setUser(name: String, age: Int): String {
    return "Nama kamu adalah $name, dan umur kamu $age tahun"
}
```

Output

```
Nama kamu adalah Rasyid Ramadhani, dan umur kamu 21 tahun

Process finished with exit code 0
```

3.	If Expressions

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // If Statement
    println("\nIf Statement\t\t: ")
    val condition = true

    if (condition) {
        println("code block to be executed if condition is true")
    } else {
        println("code block to be executed if condition is false")
    }

    // If...else Expression
    println("\nIf...else Expression\t\t: ")
    val timeOpen = 8
    val timeClose = 12
    val timeNow = 7

    val classStatus = if (timeNow >= timeClose) {
        "Class already closed"
    } else if(timeNow >= timeOpen) {
        "Class is open"
    } else {
        "Class is not open yet"
    }
    println(classStatus)
}
```

Output

```
If Statement        :
code block to be executed if condition is true

If...else Expression        :
Class is not open yet

Process finished with exit code 0
```

4

Nama   : Rasyid Ramadhani

Sesi Kelas : Morning

Program   : Android Mobile Dev

4.	Nullable Types

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // val text: String = null // Null can not be a value of a non-null type String [ERROR]
    val text: String? = null                                        // [SUCCESS]

    val texts: String? = null
    if(texts != null) {
        println(texts.length)
    }
}
```

Output

```
Process finished with exit code 0
```

5.	Safe Calls and Elvis Operator

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val safeText: String? = null
    safeText?.length

    var dateOut: String? = null
    var status = "Booked"
    status = "Checkout"
    if (status.equals("Checkout")) {
        dateOut = "28/10/2022 12:05:00"
    }
    println(dateOut)
}
```

Output

```
28/10/2022 12:05:00

Process finished with exit code 0
```

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

6.	String Template

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val company = "Infinite Learning"
//  print("We are studying in " + $company)
    print("We are studying in $company")

    val score = 80
    print("Results: ${ if(score >= 80) "You win!" else "Please try again!" }")
}
```

Output

```
We are studying in Infinite LearningResults: You win!
Process finished with exit code 0
```

7.	Variable

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
//  Penggunaan var dalam inisialisasi
    var company: String = "Infinite Learning"
    company = "Nongsa Digital"
    println(company)

//  Penggunaan val dalam inisialisasi
    val companyies: String = "Nongkrong Digital"
//  companyies = "Nongsa Digital"   --> Akan error, karena inisialisasi dengan val tidak bisa mengubah isi data variabel
    println(companyies)
}
```

Output

```
Nongsa Digital
Nongkrong Digital

Process finished with exit code 0
```

6

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

KOTLIN CONTROL FLOW

1.	If Else Expressions

Code

```kotlin
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
//    val timeClose = 8
//    val timeNow = 8
//    val isClosed = if(timeNow >= timeClose) "Class already closed" else "Class is open"
//    println(isClosed)

//  another
    val timeOpen = 8
    val timeClose = 12
    val timeNow = 7
    val classStatus = if (timeNow >= timeClose) {
        "Class already closed"
    } else if(timeNow >= timeOpen) {
        "Class is open"
    } else {
        "Class is not open yet"
    }
    println(classStatus)
}
```

Output

```
Class is not open yet

Process finished with exit code 0
```

2.	When Expressions

Code

```kotlin
/*
    Created by Rasyid Ramadhani [20104099] - 03/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val day = 1
    val result = when (day) {
        1 -> "Monday"
        2 -> "Tuesday"
        3 -> "Wednesday"
        4 -> "Thursday"
        5 -> "Friday"
        6 -> "Saturday" 7 -> "Sunday"
        else -> "Invalid day."
    }
    println(result)

//  Multiply Condition
    val days = 2
    when (days) {
        1, 2, 3, 4, 5 -> println("Weekday")
        else -> println("Weekend")
    }
```

Nama   : Rasyid Ramadhani

Sesi Kelas : Morning

Program   : Android Mobile Dev

```
27    //  Example
28        val dayz = 2
29        when (dayz) {
30            1 -> { println("First day of the week")
31                println("Monday")
32            } 2 -> {
33                println("Second day of the week")
34                println("Tuesday")
35            }
36            3 -> {
37                println("Third day of the week")
38                println("Wednesday")
39            }
40            4 -> println("Thursday")
41            5 -> println("Friday")
42            6 -> println("Saturday")
43            7 -> println("Sunday")
44            else -> println("Invalid day.")
45        }
46    }
```

Output

```
Monday
Weekday
Second day of the week
Tuesday

Process finished with exit code 0
```

3.	Range

Code

```
2     /*
3        Created by Rasyid Ramadhani [20104099] - 04/04/2023
4        Telkom Institute of Technology Purwokerto
5        SIB#4 Infinite Learning_Morning_Android Mobile Dev
6     */
7     fun main() {
8         val rangeInt = 1 ≤ .. ≤ 10
9         print (rangeInt.toList())
10        println("Step: " + rangeInt.step)
11
12        // Using rangeTo().
13        println("\nMenggunakan rangeTo()\t\t: ")
14        val rangeInts = 1.rangeTo( other 10)
15        println("Step: "+ rangeInts.step)
16        print (rangeInts.toList())
17
18        // Using step
19        println("\nMenggunakan step\t\t: ")
20        val rangeIntz = 1 ≤ .. ≤ 10 step 2
21        println("Step: ${rangeIntz.step}")
22        print (rangeIntz.toList())
24        // rangeTo()
25        println("\nMenggunakan rangeTo()\t\t: ")
26        var i = 4
27        if (i in 1.rangeTo( other 10)) { // equivalent of i >= 1 && i <= 10
28            println("Value 4 available in Range")
29        }
30
31        // downTo()
32        println("\nMenggunakan downTo()\t\t: ")
33        var j = 20
34        if (j in 10.downTo( to 1)) {
35            println("Value 4 is not available in Range")
36        }
37
38    }
```

Nama　　　: Rasyid Ramadhani

Sesi Kelas　: Morning

Program　　: Android Mobile Dev

Output

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]Step: 1

Menggunakan rangeTo()      :
Step: 1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Menggunakan step           :
Step: 2
[1, 3, 5, 7, 9]
Menggunakan rangeTo()      :
Value 4 available in Range

Menggunakan downTo()        :

Process finished with exit code 0
```

4.	Looping For
Code	

```
/*
        Created by Rasyid Ramadhani [20104099] - 04/04/2023
        Telkom Institute of Technology Purwokerto
        SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // Looping For untuk Range Expressions
    println("\nFor untuk Range Expressions \t\t: ")
    for (i in 1 ≤ .. ≤ 20){
        println("Value is $i")
    }

    // Looping For untuk Iterasi Array
    println("\nFor untuk Range Expressions \t\t: ")
    val days = arrayOf("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
    for (day in days) {
        println(day)
    }

    // Looping for dengan with index
    println("\nFor dengan withIndex \t\t: ")
    val dayz = arrayOf("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
    for ((index, value) in dayz.withIndex()) {
        println("value $value with index $index")
    }

    // Looping for dengan forEach
    println("\nFor dengan forEach \t\t: ")
    val daye = arrayOf("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
    daye.forEach { value-> println("Day is $value") }

    // Looping for dengan forEachIndexed
    println("\nFor dengan forEachIndexed \t\t: ")
    val dayc = arrayOf("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
    dayc.forEachIndexed { index, value -> println("Day $value with index: $index") }
}
```

| **Output** | |

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

```
For untuk Range Expressions        :
Value is 1
Value is 2
Value is 3
Value is 4
Value is 5
Value is 6
Value is 7
Value is 8
Value is 9
Value is 10
Value is 11
Value is 12
Value is 13
Value is 14
Value is 15
Value is 16
Value is 17
Value is 18
Value is 19
Value is 20

For untuk Range Expressions        :
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

For dengan withIndex       :
value Monday with index 0
value Tuesday with index 1
value Wednesday with index 2
value Thursday with index 3
value Friday with index 4
value Saturday with index 5
value Sunday with index 6

For dengan forEach       :
Day is Monday
Day is Tuesday
Day is Wednesday
Day is Thursday
Day is Friday
Day is Saturday
Day is Sunday

For dengan forEachIndexed       :
Day Monday with index: 0
Day Tuesday with index: 1
Day Wednesday with index: 2
Day Thursday with index: 3
Day Friday with index: 4
Day Saturday with index: 5
Day Sunday with index: 6

Process finished with exit code 0
```

Nama　　　: Rasyid Ramadhani

Sesi Kelas　: Morning

Program　　: Android Mobile Dev

5.	Looping While

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // Looping dengan While Do
    println("\nFor dengan While Do \t\t: ")
    var i = 0
    while (i < 10) {
        println("Loop to: $1")
        i++
    }

    // Looping dengan Do While
    println("\nFor dengan Do While \t\t: ")
    var k = 0
    do {
        println("$k Kampus Merdeka")
        k++
    } while (i <= 8)

}
```

Output

```
For dengan While Do        :
Loop to: $1
Loop to: $1
Loop to: $1
Loop to: $1
Loop to: $1
Loop to: $1
Loop to: $1
Loop to: $1
Loop to: $1
Loop to: $1

For dengan Do While        :
0 Kampus Merdeka

Process finished with exit code 0
```

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

6.	Break

Code

```kotlin
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // Example 1
    println("\nExample 1 \t\t: ")
    var i = 0
    while (true) {
        println("Break $i")
        i++
        if (i > 10) {
            break
        }
    }

    // Example 2
    val c = 2
    for (b in 1 .. 10) {
        println("Sayang...")
        if(b == c) {
            println("Putus!")
            break
        }
    }
}
```

Output

```
Example 1       :
Break 0
Break 1
Break 2
Break 3
Break 4
Break 5
Break 6
Break 7
Break 8
Break 9
Break 10
Sayang...
Sayang...
Putus!

Process finished with exit code 0
```

Nama   : Rasyid Ramadhani

Sesi Kelas : Morning

Program   : Android Mobile Dev

7.	Continue

Code

```kotlin
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // Example 1
    println("\nExample 1 \t\t: ")
    val listInt = listOf(1, 2, null, 4, 5, null, 7)
    for (i in listInt) {
        if (i == null) continue
        print(i)
    }

    // Example 2
    println("\nExample 2 \t\t: ")
    println("Print odd numbers")
    for (i in 1 ≤ .. ≤ 20) {
        if (i % 2 == 0) {
            continue
        }
        println("$i is an odd number")
    }
}
```

Output

```
Example 1      :
12457
Example 2      :
Print odd numbers
1 is an odd number
3 is an odd number
5 is an odd number
7 is an odd number
9 is an odd number
11 is an odd number
13 is an odd number
15 is an odd number
17 is an odd number
19 is an odd number

Process finished with exit code 0
```

Nama　　　 : Rasyid Ramadhani

Sesi Kelas　: Morning

Program　　: Android Mobile Dev

KOTLIN COLLECTION

1.	List

Code

```kotlin
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // listOf()
    val numberList = listOf("One", "Two", "Tree")
    println(numberList)

    // mutableListOf()
    val numberList2 = mutableListOf(1, 2, 3)
    println(numberList2)
    numberList2[2] = 5
    // error --> numberList2[2] = "Three"
    // error karena beda tipe datanya yang mau diubah
    println(numberList2)

    // List <Any>
    val numberList3 = mutableListOf<Any>(false, "One", 2, 3.5)
    println(numberList3)

    numberList3.remove( element: false)
    numberList3.set(0, 1)
    numberList3[2] = 3
    numberList3.add( index: 3,  element: 4)
    println(numberList3)
}
```

Output

```
[One, Two, Tree]
[1, 2, 3]
[1, 2, 5]
[false, One, 2, 3.5]
[1, 2, 3, 4]

Process finished with exit code 0
```

Nama　　　　: Rasyid Ramadhani

Sesi Kelas　: Morning

Program　　: Android Mobile Dev

2.	Set

Code

```kotlin
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // setOf(), Check value exist in set
    val setA = setOf<Int>(1, 2, 4, 2, 1, 5)
    val setB = setOf<Int>(1, 2, 4, 5)
    val setC = setOf<Int>(1, 3, 2, 4)
    val setD = setOf<Int>(1, 3, 2, 4, 4, 1, 2)
    println(setA == setB) // true

    val hasil = setA == setB
    println(hasil)

    val hasil2 = setC == setB
    println(hasil2)

    //  angka 4, 1, 2 di tiga terakhir tidak muncul karena bersifat tidak mencetak data duplikat (redudansi data)
    println(setD)

    // Union and Intersect
    val list1 = listOf(1, 1, 2,3, 5, 8, -1)
    val list2 = setOf(1, 1, 2, 2,3, 5)
    val list3 = mutableSetOf(6, 7)

    val intersect = list1 intersect list2
    val union = list1 union list2 union list3

    println(intersect) // Will display the same data println(union)
    println(union) // Merge data and eliminate duplicates
}
```

Output

```
true
true
false
[1, 3, 2, 4]
[1, 2, 3, 5]
[1, 2, 3, 5, 8, -1, 6, 7]

Process finished with exit code 0
```

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

3.	Map

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // Example 1
    val groupMap = mapOf<Int, String>(
        1 to "Group 1",
        2 to "Group 2",
        3 to "Group 3",
        4 to "Group 4",
        5 to "Group 5",
        6 to "Group 6",
        7 to "Group 3",
    )
    println(groupMap)
    println(groupMap[3])
    println(groupMap.getValue( key: 3))
    println(groupMap.values)
    println(groupMap.keys)

    // Example 2
    val numberList = listOf(1, 2, 3, 4, 5)
    val multipliedBy5 = numberList.map { it * 5 } // [5, 10, 15, 20, 25] print(multipliedBy5)
}
```

Output

```
{1=Group 1, 2=Group 2, 3=Group 3, 4=Group 4, 5=Group 5, 6=Group 6, 7=Group 3}
Group 3
Group 3
[Group 1, Group 2, Group 3, Group 4, Group 5, Group 6, Group 3]
[1, 2, 3, 4, 5, 6, 7]

Process finished with exit code 0
```

4.	Filter()

Code

```
package com.Rasyid.kotlinInfinite.week1.collection
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val numberList = listOf(11, 12, 13, 14, 15, 1, 2, 3, 4, 5, 6, 2, 4, 8)
    val eventList = numberList.filter {it % 2 == 0 } // [12, 14, 2, 4, 6, 2, 4, 8]
    val notEventList = numberList.filterNot {it % 2 == 0 } // [11, 13, 15, 1, 3, 5]

    println(numberList)
    println(eventList)
    println(notEventList)
}
```

Output

```
[11, 12, 13, 14, 15, 1, 2, 3, 4, 5, 6, 2, 4, 8]
[12, 14, 2, 4, 6, 2, 4, 8]
[11, 13, 15, 1, 3, 5]

Process finished with exit code 0
```

Nama   : Rasyid Ramadhani

Sesi Kelas : Morning

Program   : Android Mobile Dev

5.	Find(), firstOrNull(), dan lastOrNull()

Code

```
package com.Rasyid.kotlinInfinite.week1.collection
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    // Example 2
    val numberList2 = listOf(11, 12, 13, 14, 15, 1, 2, 3, 4, 5, 6, 2, 4, 8)
    val oddNumber = numberList2.find { it % 2 == 1}              // 11
    val firstOrNullNumber = numberList2.firstOrNull { it % 2 == 3 }    // null
    val lastOrNullNumber = numberList2.lastOrNull { it % 2 == 3 }      // null

    println(oddNumber)
    println(firstOrNullNumber)
    println(lastOrNullNumber)
}
```

Output

```
11
null
null

Process finished with exit code 0
```

6.	First() & Last()

Code

```
package com.Rasyid.kotlinInfinite.week1.collection
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val fruit = arrayOf(12, "Apple", "Orange", "Mango", "Melon", 11, "Juminten")
    val firstData = fruit.first()
    val latestData = fruit.last()

    println(firstData) // Monday
    println(latestData) // Sunday
}
```

Output

```
12
Juminten

Process finished with exit code 0
```

17

Nama : Rasyid Ramadhani

Sesi Kelas : Morning

Program : Android Mobile Dev

7.	Sum()

Code

```
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val numberList = listOf(1, 3, 2, 6, 5, 4)

    val total = numberList.sum()
    print(total) // 21
}
```

Output

```
21
Process finished with exit code 0
```

8.	Sorted()

Code

```
package com.Rasyid.kotlinInfinite.week1.collection
/*
    Created by Rasyid Ramadhani [20104099] - 04/04/2023
    Telkom Institute of Technology Purwokerto
    SIB#4 Infinite Learning_Morning_Android Mobile Dev
*/
fun main() {
    val numberList = listOf(1, 3, 2, 6, 5, 4)
    val hackerChar = listOf('h', 'a', 'c', 'k', 'e', 'r')

    // Ascending Sorted
    val ascendingSort1 = numberList.sorted()
    val ascendingSort2 = hackerChar.sorted()

    println(ascendingSort1) // [1, 2, 3, 4, 5, 6]
    println(ascendingSort2) // [a, c, e, h, k, r]

    // Descending Sorted
    val descendingSort1 = numberList.sortedDescending()
    val descendingSort2 = hackerChar.sortedDescending()

    println(descendingSort1) // [1, 2, 3, 4, 5, 6]
    println(descendingSort2) // [a, c, e, h, k, r]
}
```

Output

```
[1, 2, 3, 4, 5, 6]
[a, c, e, h, k, r]
[6, 5, 4, 3, 2, 1]
[r, k, h, e, c, a]

Process finished with exit code 0
```