

Setoran:

File Laporan Pengamatan: **PAM_LaporanPraktikum1_{NIM}.pdf**

Source Code: **PAM_SourceCode1_{NIM}.zip**

Persiapan Praktikum:

1. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android.

- Mengaktifkan Hyper-V pada Windows:

[Hyper-V](#)

- Tahapan instalasi Android Studio:

[Android Studio](#)

2. Kotlin

Kotlin adalah bahasa pemrograman modern yang dirancang untuk meningkatkan produktivitas pengembang. Dikembangkan oleh JetBrains, Kotlin berjalan di atas Java Virtual Machine (JVM) dan menawarkan sintaks yang lebih ringkas dan ekspresif.

- Menambahkan Kotlin Android Studio ke Lingkungan Windows:

[Kotlin \(Android Studio\)](#)

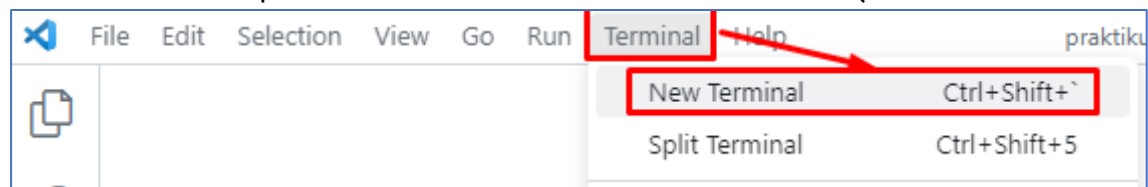
3. VSCode

VSCode (Visual Studio Code), merupakan text editor yang dikembangkan oleh Microsoft. Dirancang untuk mendukung berbagai bahasa pemrograman.

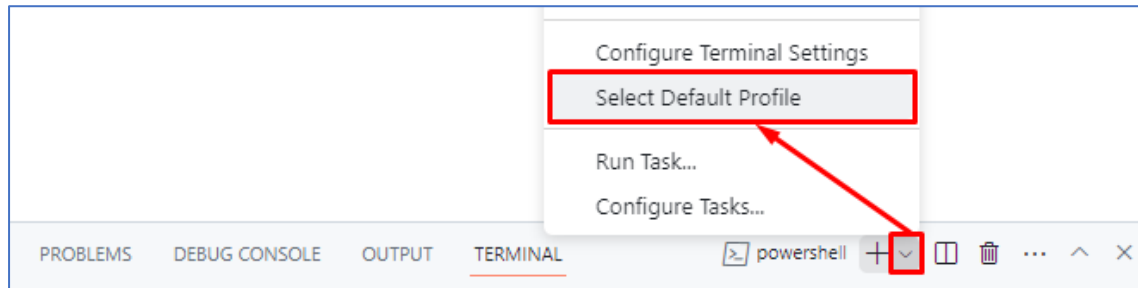
- Tahapan instalasi VSCode:

[VSCode](#)

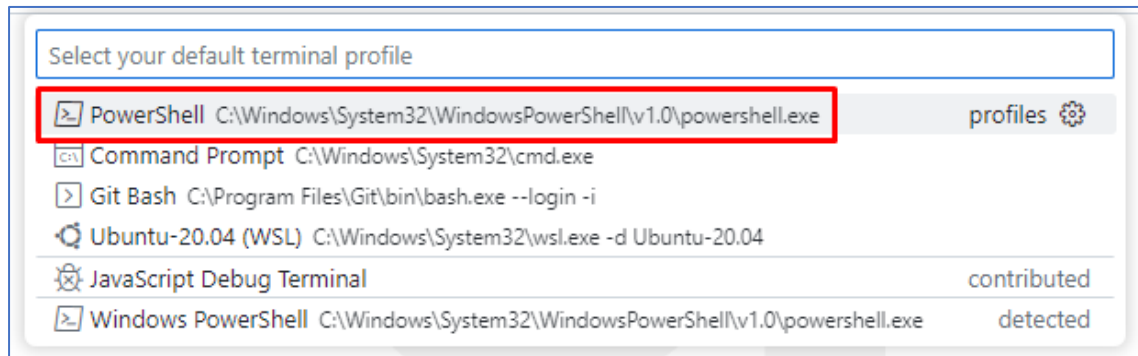
- Pada VSCode pilih menu Terminal > New Terminal (Membuka Terminal)



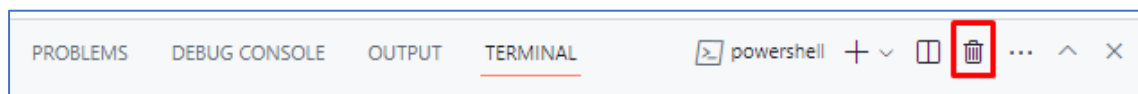
- Akan tampil panel terminal, silahkan pilih ikon "arrow-down" > Select Default Profile. (Mengubah Terminal yang Digunakan)



- Akan tampil daftar terminal profile yang dapat digunakan. Silahkan pilih pada "PowerShell".



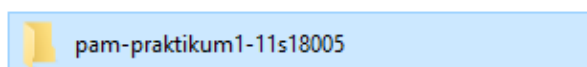
- Hapus terminal dengan menekan ikon "Keranjang Sampah" pada panel terminal.



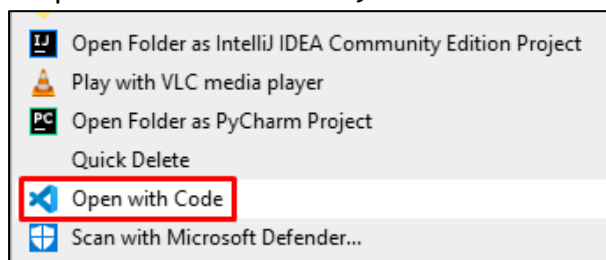
4. Workspace

Workspace pada VSCode adalah istilah yang digunakan untuk menyebut kumpulan file dan folder terkait suatu proyek atau pekerjaan yang sedang dikerjakan di editor tersebut. Workspace ini membantu kita mengelola proyek secara terorganisir dan mudah diakses

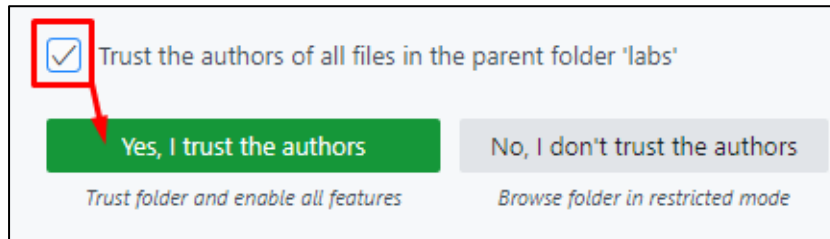
- Buat folder baru dengan format penamaan "**pam-praktikum1-{nim}**" dalam huruf kecil untuk setiap karakter, seperti berikut:



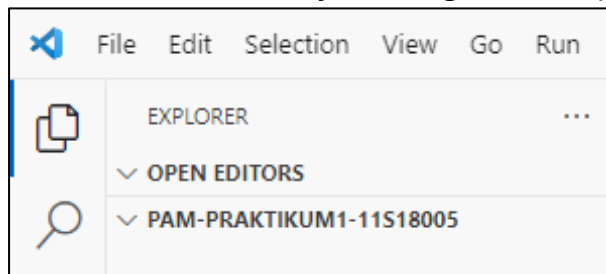
- Klik kanan pada folder yang telah dibuat sebelumnya, selanjutnya pada *context menu*, silahkan memilih opsi "**Open with Code**".



- Apabila tampil dialog, silahkan centang opsi berikut, setelah itu pilih tombol "**Yes, I trust the authors**".

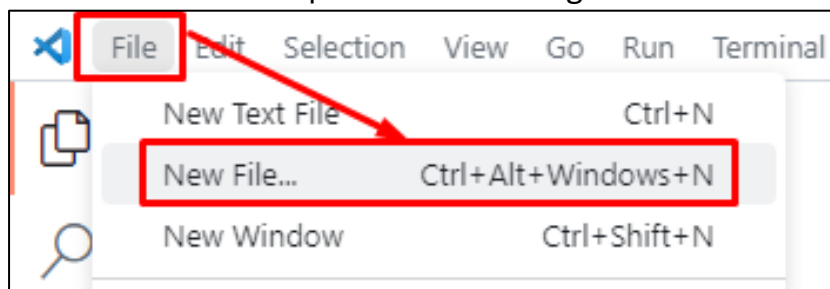


- Akan terbuka aplikasi VSCode dan menjadikan folder yang telah dibuat sebelumnya sebagai *workspace*.

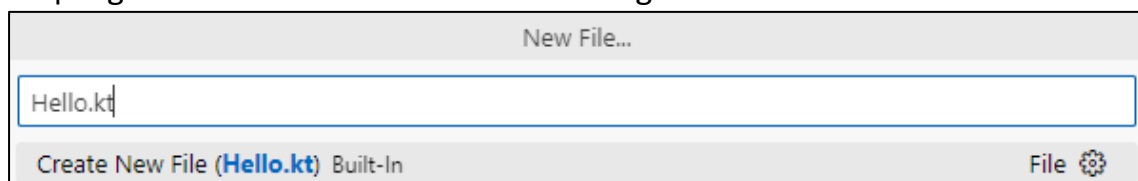


5. Compile & Run Program Kotlin

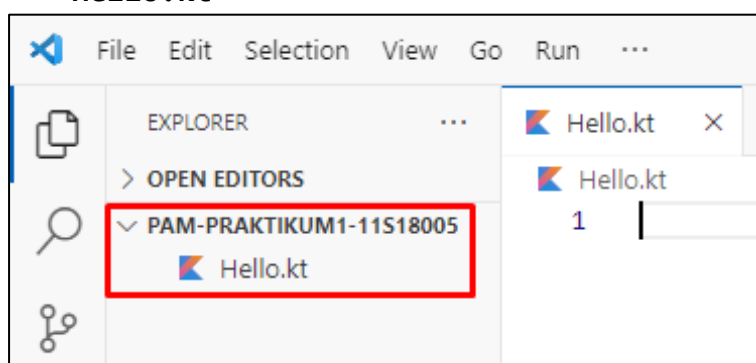
- Buat file baru pada VSCode dengan memilih menu File > New File...



- Akan tampil dialog untuk membuat file baru, pada kotak isian silahkan mengetik "Hello.kt". Silahkan dicatat bahwa setiap program Kotlin harus diakhiri dengan extension ".kt".



- Sekarang pada Workspace kita akan terdapat file baru dengan nama "Hello.kt"



- Modifikasi isi file "Hello.kt" menjadi seperti berikut

```
1 fun main() {  
2     println("Belajar Pengembangan Aplikasi Mobile")  
3 }
```

- Buka terminal pada VSCode, selanjutnya tuliskan perintah:

```
kotlinc Hello.kt
```

PROBLEMS DEBUG CONSOLE PORTS OUTPUT TERMINAL

```
PS D:\repository\del-practice-module\2024\pam\labs\pam-praktikum1-11s18005> kotlinc Hello.kt  
PS D:\repository\del-practice-module\2024\pam\labs\pam-praktikum1-11s18005>
```

- Apabila proses kompilasi berhasil maka akan terdapat 1 file baru pada workspace kita, yaitu: **HelloKt.class**.

File Edit Selection View Go

EXPLORER

> OPEN EDITORS

✓ PAM-PRAKTIKUM1-11S18005

> META-INF

Hello.kt

HelloKt.class

- Jalankan program Kotlin yang telah berhasil dikompilasi dengan menuliskan perintah:

```
kotlin HelloKt
```

PROBLEMS DEBUG CONSOLE PORTS OUTPUT TERMINAL

```
PS D:\repository\del-practice-module\2024\pam\labs\pam-praktikum1-11s18005> kotlin HelloKt  
Belajar Pengembangan Aplikasi Mobile  
PS D:\repository\del-practice-module\2024\pam\labs\pam-praktikum1-11s18005>
```

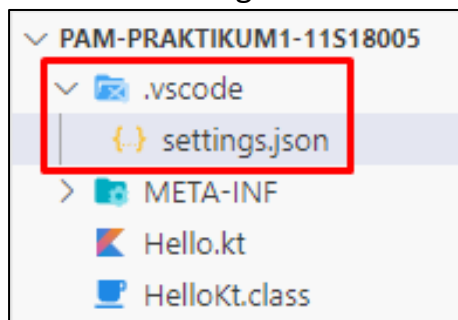
- Jika berhasil maka akan tampil pesan "Belajar Pengembangan Aplikasi Mobile".

6. Menyembunyikan File di VSCode

Menggunakan VSCode kamu dapat menyembunyikan file yang tidak ingin kamu lihat pada Explorer VSCode. Dalam kasus ini sebenarnya kita tidak perlu untuk melihat file hasil compile Kotlin seperti file dengan ekstensi ".class" dan folder "META-INF".



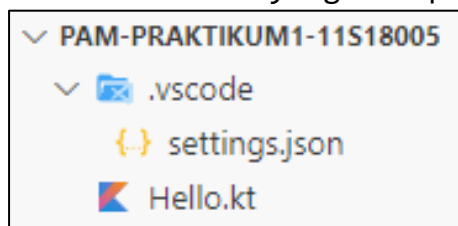
Untuk menghilangkan file hasil compile Kotlin, silahkan membuat folder baru dengan nama **".vscode"** di dalam folder tersebut buat file baru dengan nama **"settings.json"**, seperti berikut:



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 }
```

Jika berhasil maka pada Explorer VSCode hanya terlihat file dengan ekstensi **".kt"** yang merupakan file kode Haskell.



Latihan Praktikum:

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

A. Dasar Kotlin

1. Variable

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

```
Variable.kt
1  // Konstanta
2  const val PI = 3.14
3
4  fun main() {
5      // Variabel Immutable
6      val pi: Double = 3.14
7      println(pi)
8
9      // Variabel Mutable
10     var counter: Int = 0
11     counter += 1
12     println(counter)
13
14     // Tipe Data Otomatis
15     val name = "Abdullah"
16     println(name)
17
18     // Nullable Types
19     var think: String? = null
20     println(think)
21
22     // Konstanta
23     println(PI)
24 }
```

Analisis perbedaan dari masing-masing variable pada line 2, 6, 10, 15 dan 19. Silahkan melakukan compile & run program secara mandiri.

2. Tipe Data

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

```
TipeData.kt
1  fun main() {
2      // Tipe Data Angka
3      val angkaInt: Int = 42
4      val angkaDouble: Double = 3.14
5      println(angkaInt)
6      println(angkaDouble)
7
8      // Tipe Data Karakter
```

TipeData.kt

```
9    val huruf: Char = 'A'
10   println(huruf)
11
12   // Tipe Data Boolean
13   val benar: Boolean = true
14   println(benar)
15
16   // Tipe Data String
17   val teks: String = "Saya belajar Pengembangan Aplikasi Mobile"
18   println(teks)
19
20   // Tipe Data Array
21   val arr: Array<Int> = arrayOf(1, 2, 3, 4, 5)
22   println(arr)
23
24   // Tipe Data Nullable
25   var think: String? = null
26   println(think)
27
28   // Tipe Data Any
29   val obj: Any = "Ini adalah objek"
30   println(obj)
31
32   // Memanggil fungsi
33   cetakPesan();
34 }
35
36 // Tipe Data Unit (mirip dengan void di bahasa pemrograman lainnya)
37 fun cetakPesan(): Unit {
38     println("Wah, ini pesan!")
39 }
```

Analisis kegunaan dari masing-masing tipe data. Silahkan melakukan compile & run program secara mandiri.

3. Data String

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

DataString.kt

```
1  fun main() {
2      // Deklarasi string
3      val nama1: String = "Abdullah"
4      val nama2 = "Ubaid"
5
6      // Operasi string
7      // --> menggabungkan string
8      val hasil1: String = nama1 + " " + nama2
9      // --> interpolasi string
10     val hasil2: String = "$nama1 $nama2"
11     // --> manipulasi string
```

DataString.kt

```
12 val hasil3: String = "Hay, perkenalkan nama saya $nama1 $nama2"
13
14 println(hasil1);
15 println(hasil2);
16 println(hasil3);
17
18 // Fungsi pada string
19 val teks: String = "Belajar Kotlin "
20 // --> mendapatkan panjang string
21 val panjang: Int = teks.length
22 // --> mengonversi ke huruf kecil
23 val hurufKecil: String = teks.lowercase()
24 // --> Mengonversi ke huruf besar
25 val hurufBesar: String = teks.uppercase()
26 // --> memeriksa apakah string berisi suatu substring
27 val mengandungSubstring: Boolean = teks.contains("Kotlin")
28 // --> mengulangi string
29 val ulangTeks: String = teks.repeat(3)
30
31 println(teks);
32 println(panjang);
33 println(hurufKecil);
34 println(hurufBesar);
35 println(mengandungSubstring);
36 println(ulangTeks);
37
38 // String raw
39 val teksMultiline: String = """
40     Ini adalah
41     string
42     multiline
43     """
44 println(teksMultiline);
45 }
```

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

4. Data Collection

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

DataCollection.kt

```
1 fun main() {
2     // List
3     val angka: List<Int> = listOf(1, 2, 3, 1, 2, 3)
4     angka.forEach { print("${it} | ") }
5     println()
6
7     // Mutable List
8     val mutableList: MutableList<Int> = mutableListOf(1, 2, 3, 1, 2, 3)
```


DataCollection.kt

```
9 mutableList.add(4)
10 println(mutableList)
11 mutableList.forEach { print("${it} | ") }
12 println()
13
14 // Set
15 val huruf: Set<Char> = setOf('a', 'b', 'c', 'a', 'b', 'c')
16 huruf.forEach { print("${it} | ") }
17 println()
18
19 // Mutable Set
20 val mutableSet: MutableSet<String> = mutableSetOf("apple", "banana",
"orange", "apple", "banana", "orange")
21 mutableSet.add("grape")
22 mutableSet.forEach { print("${it} | ") }
23 println()
24
25 // Map
26 val map: Map<String, Int> = mapOf("satu" to 1, "dua" to 2, "tiga" to
3)
27 // Mutable Map
28 val mutableMap: MutableMap<String, Int> = mutableMapOf("one" to 1,
"two" to 2)
29 mutableMap["three"] = 3
30 map.forEach { (kunci, nilai) -> print("$kunci -> $nilai | ") }
31 println()
32
33 // Lainnya
34 // Array
35 val numbers: Array<Int> = arrayOf(1, 2, 3, 4, 5)
36 numbers.forEach { print("${it} | ") }
37 println()
38
39 // Sequence
40 val sequence: Sequence<Int> = sequenceOf(1, 2, 3, 4, 5)
41 sequence.forEach { print("${it} | ") }
42 println()
43 }
```

Analisis masing-masing kegunaan dan perbedaan dari koleksi di atas. Silahkan melakukan compile & run program secara mandiri.

5. Data List

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

DataList.kt

```
1 fun main() {
2     // Deklarasi list
3     // --> list dengan tipe data Int
4     val angka: List<Int> = listOf(1, 2, 3, 4, 5)
```

DataList.kt

```
5 // --> list dengan tipe data String
6 val kata: List<String> = listOf("satu", "dua", "tiga")
7 // --> list dengan tipe data campuran (Any)
8 val campuran: List<Any> = listOf(1, "dua", 3.0, true)
9 // --> list bernilai null
10 val listNullable: List<Int>? = null
11
12 angka.forEach { print("${it} | ") }
13 println()
14 kata.forEach { print("${it} | ") }
15 println()
16 campuran.forEach { print("${it} | ") }
17 println()
18 println(listNullable)
19
20 // List Mutable
21 val mutableList: MutableList<Int> = mutableListOf(1, 2, 3)
22 mutableList.add(4)
23 mutableList.removeAt(0)
24
25 mutableList.forEach { print("${it} | ") }
26 println()
27
28 // Mengakses elemen dalam list
29 // --> mengakses elemen pertama
30 val elemenPertama: Int = angka[0]
31 // --> mengakses elemen terakhir
32 val elementTerakhir: Int = angka[angka.size - 1]
33
34 println(elemenPertama)
35 println(elementTerakhir)
36
37 // Operasi list
38 // --> menghitung jumlah elemen
39 val jumlahElemen: Int = angka.size
40 // --> mengecek apakah List kosong
41 val isEmpty: Boolean = angka.isEmpty()
42 // --> membuat list baru dengan transformasi
43 val kaliDua: List<Int> = angka.map { it * 2 }
44
45 println(jumlahElemen)
46 println(isEmpty)
47 kaliDua.forEach { print("${it} | ") }
48 println()
49 }
```

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. Input & Output

Silahkan membuat fungsi yang menerima satu buah string dan menampilkannya inputan tersebut menggunakan huruf kapital pada terminal.

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

Tantangan1.kt

```
fun main() {  
    val name = readLine()  
  
    // TODO: tuliskan kode kamu disini  
}
```

Hasil yang diharapkan sebagai berikut:

Input	Output
Abdullah	ABDULLAH
Abdullah Ubaid	ABDULLAH UBAID