LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : PEMROGRAMAN KOTLIN
NAMA : MUHAMMAD RIZQON MAULANA
NIM : 215150201111022
TANGGAL : 11/09/2023
ASISTEN : - ANAK AGUNG GDE AGASTYA MAHESWARA
- GEDE PRAMANANDA KUSUMA WISESA

TUGAS 1

A. Soal

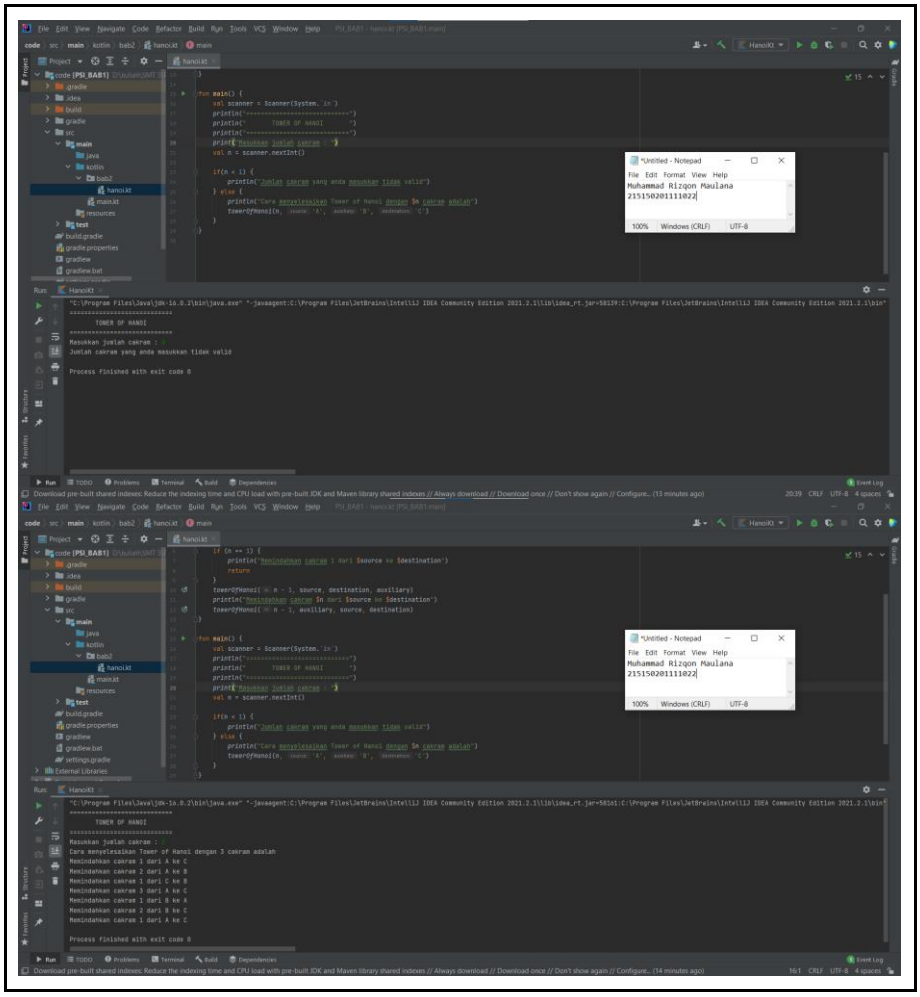
Buatlah program interaktif untuk menyelesaikan permasalahan menara Hanoi

B. Source Code

hanoi.kt

```
1  package bab2
2
3  import java.util.Scanner
4
5  fun towerOfHanoi(n: Int, source: Char, auxiliary:
6  Char, destination: Char) {
7      if (n == 1) {
8          println("Memindahkan cakram 1 dari $source
9  ke $destination")
10         return
11     }
12     towerOfHanoi(n - 1, source, destination,
13 auxiliary)
14     println("Memindahkan cakram $n dari $source ke
15 $destination")
16     towerOfHanoi(n - 1, auxiliary, source,
17 destination)
18 }
19
20 fun main() {
21     val scanner = Scanner(System.`in`)
22     println("============================")
23     println("       TOWER OF HANOI       ")
24     println("============================")
25     print("Masukkan jumlah cakram : ")
26     val n = scanner.nextInt()
27
28     if(n < 1) {
29         println("Jumlah cakram yang anda masukkan
30 tidak valid")
31     } else {
32         println("Cara menyelesaikan Tower of Hanoi
33 dengan $n cakram adalah")
         towerOfHanoi(n, 'A', 'B', 'C')
     }
 }
```

C. Screenshot



D. Penjelasan

Program ini memungkinkan pengguna untuk memasukkan jumlah cakram yang ingin dipindahkan dalam permainan Tower of Hanoi. Setelah input diterima, program akan mencetak langkah-langkah yang diperlukan untuk menyelesaikan masalah tersebut. Program menggunakan rekursif untuk memecahkan masalah ini. Jika jumlah cakram adalah 1, program langsung mencetak langkah pemindahan cakram. Jika tidak, program memecahkan masalah menjadi tiga langkah: pertama, memindahkan n-1 cakram dari tiang awal ke tiang bantu dengan menggunakan tiang tujuan sebagai tiang bantu; kedua, memindahkan cakram ke-n dari tiang awal ke tiang tujuan; dan ketiga, memindahkan kembali n-1 cakram dari tiang bantu ke tiang tujuan dengan menggunakan tiang awal sebagai tiang bantu. Setiap langkah pemindahan cakram dicetak ke layar sehingga pengguna dapat melihat cara menyelesaikan masalah Tower of Hanoi dengan jumlah cakram yang dimasukkan.

TUGAS 2

A. Soal

Client meminta program pendataan buku. Dengan fitur menambahkan buku dan melihat seluruh daftar buku yang ada. Selesaikan program tersebut dengan pendekatan object oriented dan interaktif

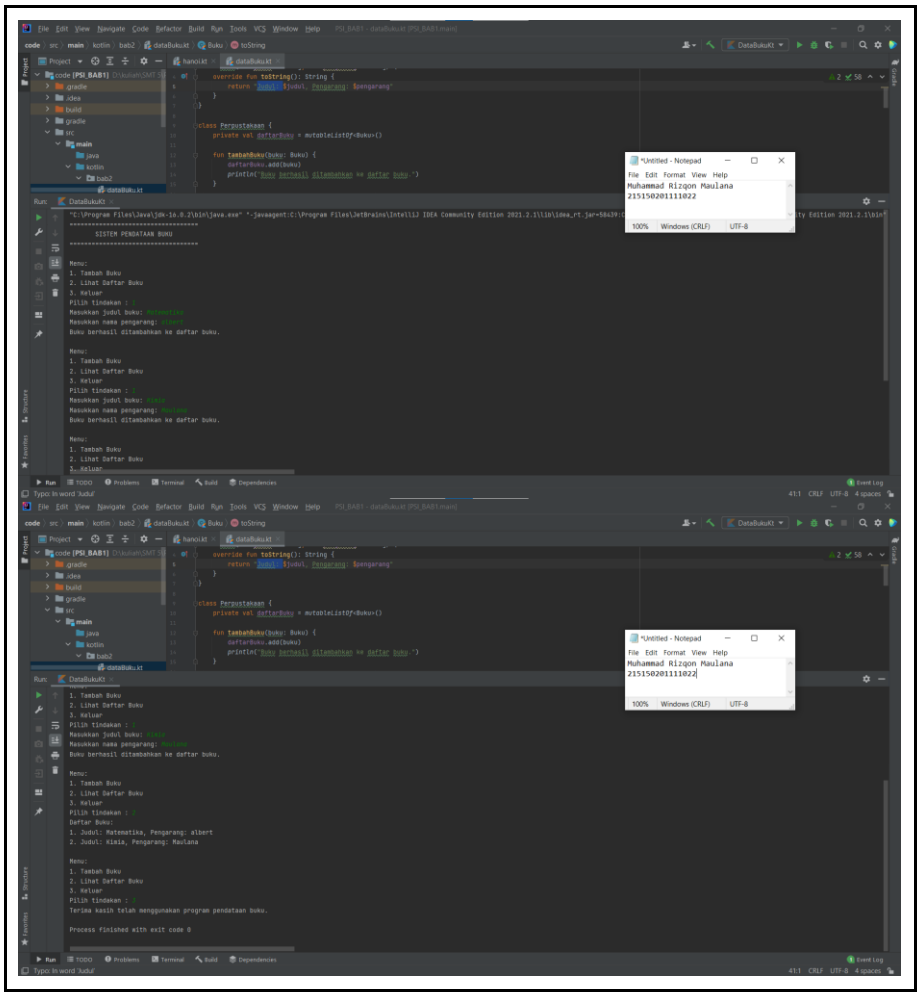
B. Source Code

dataBuku.kt

```
1  package bab2
2
3  class Buku(val judul: String, val pengarang:
4  String) {
5      override fun toString(): String {
6          return "Judul: $judul, Pengarang:
7  $pengarang"
8      }
9  }
10
11 class Perpustakaan {
12     private val daftarBuku =
13 mutableListOf<Buku>()
14
15     fun tambahBuku(buku: Buku) {
16         daftarBuku.add(buku)
17         println("Buku berhasil ditambahkan ke
18 daftar buku.")
19     }
20
21     fun lihatDaftarBuku() {
22         if (daftarBuku.isEmpty()) {
23             println("daftar buku kosong.")
24         } else {
25             println("Daftar Buku:")
26             for ((index, buku) in
27 daftarBuku.withIndex()) {
28                 println("${index + 1}. $buku")
29             }
30         }
31     }
32 }
33
34 fun main() {
35
36 println("==================================")
37     println("       SISTEM PENDATAAN BUKU
38 ")
39
40 println("==================================")
41
42     val perpustakaan = Perpustakaan()
```

```
43
44      while (true) {
45          println("\nMenu:")
46          println("1. Tambah Buku")
47          println("2. Lihat Daftar Buku")
48          println("3. Keluar")
49          print("Pilih tindakan : ")
50
51          when (readLine()?.toIntOrNull()) {
52              1 -> {
53                  print("Masukkan judul buku: ")
54                  val judul = readLine()
55                  print("Masukkan nama
56  pengarang: ")
57                  val pengarang = readLine()
58                  if (judul != null &&
59  pengarang != null) {
60                      val buku = Buku(judul,
61  pengarang)
62
63  perpustakaan.tambahBuku(buku)
64                  } else {
65                      println("Input tidak
66  valid.")
67                  }
68              }
69              2 ->
70  perpustakaan.lihatDaftarBuku()
71              3 -> {
72                  println("Terima kasih telah
73  menggunakan program pendataan buku.")
74                  return
75              }
76              else -> println("Pilihan tidak
77  valid. Silakan pilih lagi.")
78          }
79      }
80  }
```

C. Screenshot



D. Penjelasan

Program ini mengikuti paradigma object-oriented dengan dua kelas utama, yaitu Buku dan Perpustakaan. Kelas Buku digunakan untuk merepresentasikan informasi buku, sedangkan kelas Perpustakaan bertanggung jawab untuk mengelola daftar buku yang mana terdapat 2 fungsi utama yaitu tambahBuku dan lihatDaftarBuku. Program ini interaktif dengan menggunakan looping, memungkinkan pengguna untuk menambahkan buku baru atau melihat daftar buku yang ada secara terus-menerus sampai pengguna keluar dari program. Selama interaksi, program juga memberikan pesan-pesan yang informatif. Ini memanfaatkan fitur-fitur Kotlin seperti classes, functions, dan control flow untuk memberikan pengalaman yang sederhana namun efektif dalam mengelola daftar buku.

TUGAS 3

A. Soal

Buatlah suatu program sederhana yang dapat menyeleksi bilangan prima pada rentang 1 – 100 dengan menggunakan minimal satu fitur pada functional programming. Jelaskan bagian kode mana yang menunjukkan bahwa kode tersebut termasuk fitur dari functional programming. Output harus seperti gambar berikut:

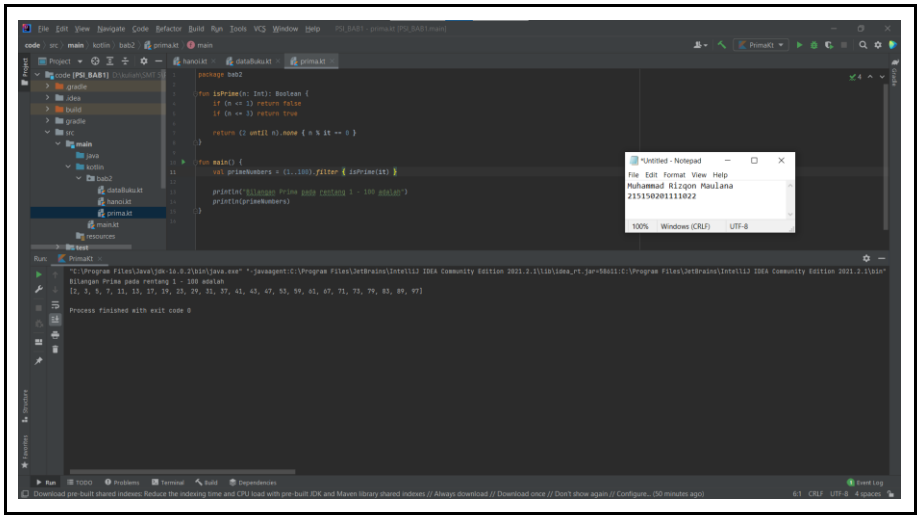
```
"C:\Program Files\Java\jdk1.8.0_141\bin\java.exe" ...
Bilangan Prima antara 1 - 100 adalah
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

Process finished with exit code 0
```

B. Source Code

Main.kt

```
1   package bab2
2
3   fun isPrime(n: Int): Boolean {
4       if (n <= 1) return false
5       if (n <= 3) return true
6
7       return (2 until n).none { n % it == 0 }
8   }
9
10  fun main() {
11      val primeNumbers = (1..100).filter
12  { isPrime(it) }
13
14      println("Bilangan Prima pada rentang 1 - 100
15  adalah")
16      println(primeNumbers)
17  }
```

C. Screenshot



D. Penjelasan

Kode di atas mencerminkan paradigma functional programming melalui penggunaan beberapa konsep utama. Salah satu fitur yang digunakan adalah penggunaan fungsi tingkat tinggi (higher-order function), seperti fungsi `filter`, yang dapat menerima fungsi lain sebagai argumen atau mengembalikan fungsi sebagai hasil. Dalam hal ini, kita menggunakan fungsi `filter` untuk memfilter elemen-elemen dalam rentang `(1..100)` berdasarkan kriteria yang didefinisikan dalam fungsi lain, yaitu `isPrime`. Dengan kata lain, kita meneruskan fungsi `isPrime` sebagai argumen ke fungsi `filter`, yang memungkinkan kita untuk dengan mudah menerapkan aturan pemfilteran.

Selanjutnya, kode tersebut menggunakan lambda expression pada baris `(1..100).filter { isPrime(it) }`. Lambda expression adalah cara untuk mendefinisikan fungsi anonim dengan cepat dan ringkas. Di sini, kita mendefinisikan fungsi anonim yang mengambil satu argumen (dalam hal ini, `it` adalah bilangan dari rentang) dan memeriksa apakah itu adalah bilangan prima dengan memanggil `isPrime(it)`. Penggunaan lambda expression memungkinkan kita untuk dengan mudah menentukan perilaku fungsi tanpa harus menulisnya secara eksplisit sebagai fungsi terpisah.

Dengan kombinasi fungsi tingkat tinggi, penggunaan fungsi sebagai argumen, dan lambda expression, kita dapat memanfaatkan konsep-konsep fundamental dalam paradigma functional programming untuk menyaring bilangan prima dalam rentang 1 hingga 100 dengan cara yang lebih ringkas dan ekspresif.