

Nama: Harry LBI
NIM: A11.2023.15332
Kelompok: A11.4405

TUGAS PALINDROME

```
1  package palindrome;
2
3  import java.util.Scanner;
4
5  public class Palindrome {
6
7      public static void main(String[] args) {
8          TextGame game = new TextGame();
9          Scanner scanner = new Scanner(System.in);
10
11         char isLanjut;
12         boolean lanjut;
13
14         do {
15             // Meminta input dari pengguna
16             System.out.print("Masukkan kata: ");
17             String input = scanner.nextLine();
18             // Menampilkan hasil
19             game.displayResult(input);
20
21             // Menanyakan apakah ingin lanjut
22             System.out.print("Apakah ingin lanjut (Y/N)? ");
23             isLanjut = scanner.nextLine().charAt(0);
24
25             if (isLanjut == 'Y' || isLanjut == 'y') {
26                 lanjut = true;
27             } else {
28                 lanjut = false;
29             }
30
31         } while (lanjut);
32
33         // Menutup scanner
34         scanner.close();
35     }
36 }
37
```

Untuk membuat program lebih dinamis, fitur untuk menangkap input dari pengguna ditambahkan menggunakan kelas `Scanner`. Loop `do-while` digunakan agar program

dapat meminta input secara berulang sampai pengguna memutuskan untuk berhenti. Berikut penjelasan langkah-langkahnya:

1. Mengimpor Kelas **Scanner**:

Sebelum digunakan, kelas **Scanner** diimpor:

```
import java.util.Scanner;
```

-
- Kelas ini memungkinkan program membaca input dari keyboard (**System.in**).

2. Membuat Objek **Scanner**:

Objek **scanner** dibuat untuk membaca input pengguna:

```
Scanner scanner = new Scanner(System.in);
```

-

3. Penggunaan Loop **do-while**:

Program menggunakan loop **do-while** agar pengguna dapat memasukkan teks berulang kali hingga memilih untuk berhenti:

```
do {  
  
    // Proses input  
  
} while (lanjut);
```

-
- Loop ini memastikan bahwa setidaknya satu kali input akan diproses, bahkan jika pengguna memutuskan untuk tidak melanjutkan setelahnya.

4. Membaca Input Pengguna:

Metode `nextLine()` digunakan untuk membaca teks dari pengguna:

```
String input = scanner.nextLine();
```

-
- Input ini diteruskan ke metode `displayResult()` untuk diproses.

5. Meminta Konfirmasi Pengguna:

Setelah hasil diproses, pengguna ditanya apakah ingin melanjutkan:

```
System.out.print("Apakah ingin lanjut (Y/N)? ");
```

```
isLanjut = scanner.nextLine().charAt(0);
```

-
- Pilihan pengguna (karakter pertama) digunakan untuk menentukan apakah loop akan diulang.

6. Menentukan Kelanjutan Program:

Berdasarkan input pengguna (`Y/y` untuk lanjut, selain itu berhenti), variabel `lanjut` diatur:

```
if (isLanjut == 'Y' || isLanjut == 'y') {  
  
    lanjut = true;  
  
} else {  
  
    lanjut = false;  
  
}
```

-

7. Menutup Scanner:

Setelah pengguna memilih berhenti, scanner ditutup:

```
scanner.close();
```

○

Manfaat

- **Interaktif:** Program menjadi lebih interaktif dengan meminta input langsung dari pengguna.
- **Fleksibel:** Dengan loop `do-while`, pengguna dapat memasukkan teks berulang kali tanpa harus menjalankan ulang program.
- **Efisien:** Scanner ditutup setelah selesai digunakan untuk menghemat sumber daya.

Dengan cara ini, pengguna dapat dengan mudah mengetikkan teks yang ingin diperiksa, melihat hasilnya, dan memilih untuk melanjutkan atau keluar dari program.

Hasil dari program palindrome

```
PBO/palindrome on □ main [!?] via ↻ v23.0.2
> java Palindrome.java
Masukkan kata: malam
Kata: malam adalah Palindrome
Apakah ingin lanjut (Y/N)? y
Masukkan kata: pagi
Kata: pagi bukan Palindrome
Apakah ingin lanjut (Y/N)? n

PBO/palindrome on □ main [!?] via ↻ v23.0.2 took 21s
> □
```