JOBSHEET III  
ARRAY OF OBJECTS

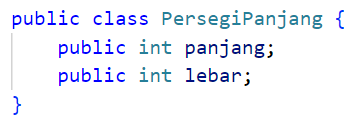
1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

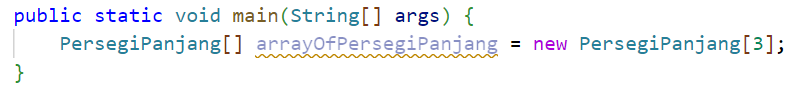
1. Memahami dan menjelaskan fungsi array yang berisikan variabel object.
2. Mahasiswa mampu menerapkan instansiasi array of objects dalam Java
3. Mahasiswa mampu melakukan operasi terhadap elemen dalam array of objects
4. Membuat Array dari Object, Mengisi dan Menampilkan

Didalam praktikum ini, kita akan mempraktekkan bagaimana membuat array of objects, kemudian mengisi dan menampilkan array tersebut.

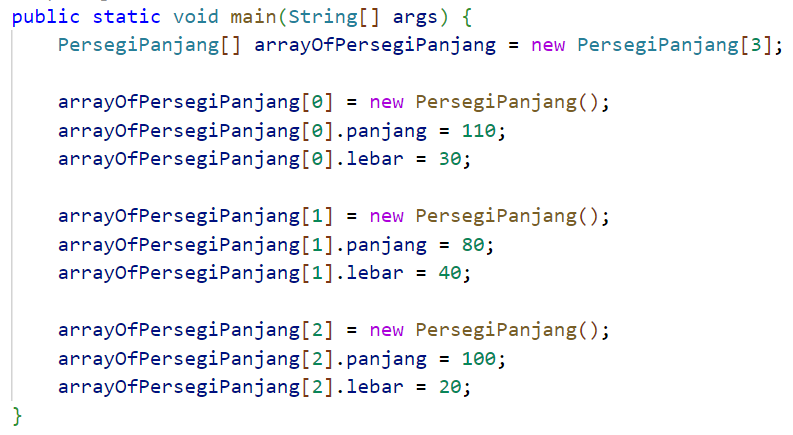
1. Langkah-langkah Percobaan
2. Buat folder baru dengan nama Praktikum03.
3. Buat class **PersegiPanjang**:



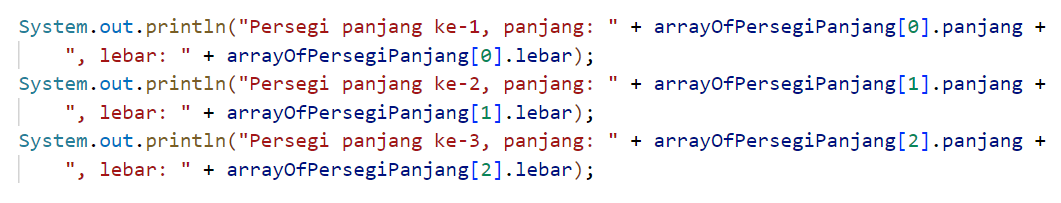
1. Buat class PersegiPanjangDemo kemudian tambahkan fungsi main sebagai berikut



1. Kemudian isikan masing-masing atributnya:

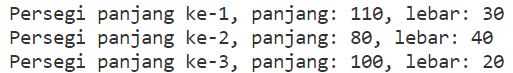


1. Cetak ke layar semua atribut dari objek **ppArray**:



1. Run program dan amati hasilnya.
2. Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.





1. Pertanyaan
2. Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method? Jelaskan!

**Jawab:**

Tidak selalu. Pada kode di atas, class PersegiPanjang memiliki atribut (panjang dan lebar) dan method (main). Namun, class yang akan dibuat array of object tidak selalu harus memiliki atribut dan sekaligus method.

1. Apa yang dilakukan oleh kode program berikut?



**Jawab:**

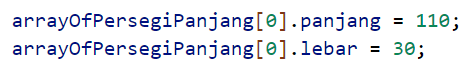
Creating an object of array

1. Apakah class PersegiPanjang memiliki konstruktor? Jika tidak, kenapa bisa dilakukan pemanggilan konstruktur pada baris program berikut?



**Jawab:** ya, class PersegiPanjang memiliki konstruktor. Karena pada baris pemograman tsb terdapat pemanggilan konstruktor. Jika class PersegiPanjang tidak mendefinisikan konstruktor secara eksplisit, Java akan secara otomatis membuatkan konstruktor default.

1. Apa yang dilakukan oleh kode program berikut?



**Jawab:** Kode program ini membuat array objek PersegiPanjang, menginisialisasi nilai propertinya, dan kemudian mencetak informasi tentang setiap objek.

1. Mengapa class PersegiPanjang dan PersegiPanjangDemo dipisahkan pada uji coba 3.2?

**Jawab:** agar class PersegiPanjang dapat digunakan kembali di program lain dan bersifat global. Selain itu membuat kode lebih mudah dipahami, diubah, dan diuji.

1. Menerima Input Isian Array Menggunakan Looping

Pada praktikum ini kita akan mengubah hasil program dari praktikum 3.2 sehingga program dapat menerima input dan menggunakan looping untuk mengisikan atribut dari semua persegi panjang

* + 1. Langkah-langkah Percobaan

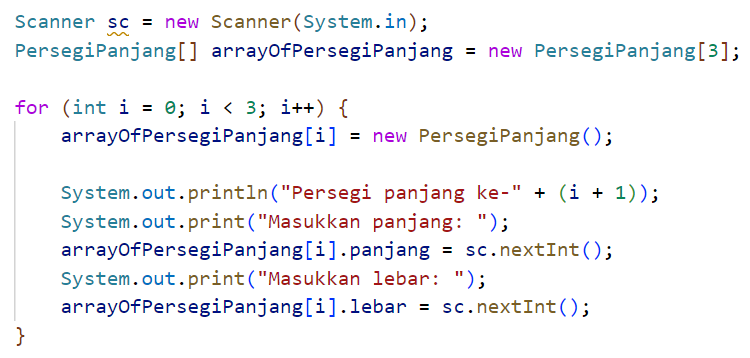
1. Import scanner pada class **PersegiPanjangDemo**.

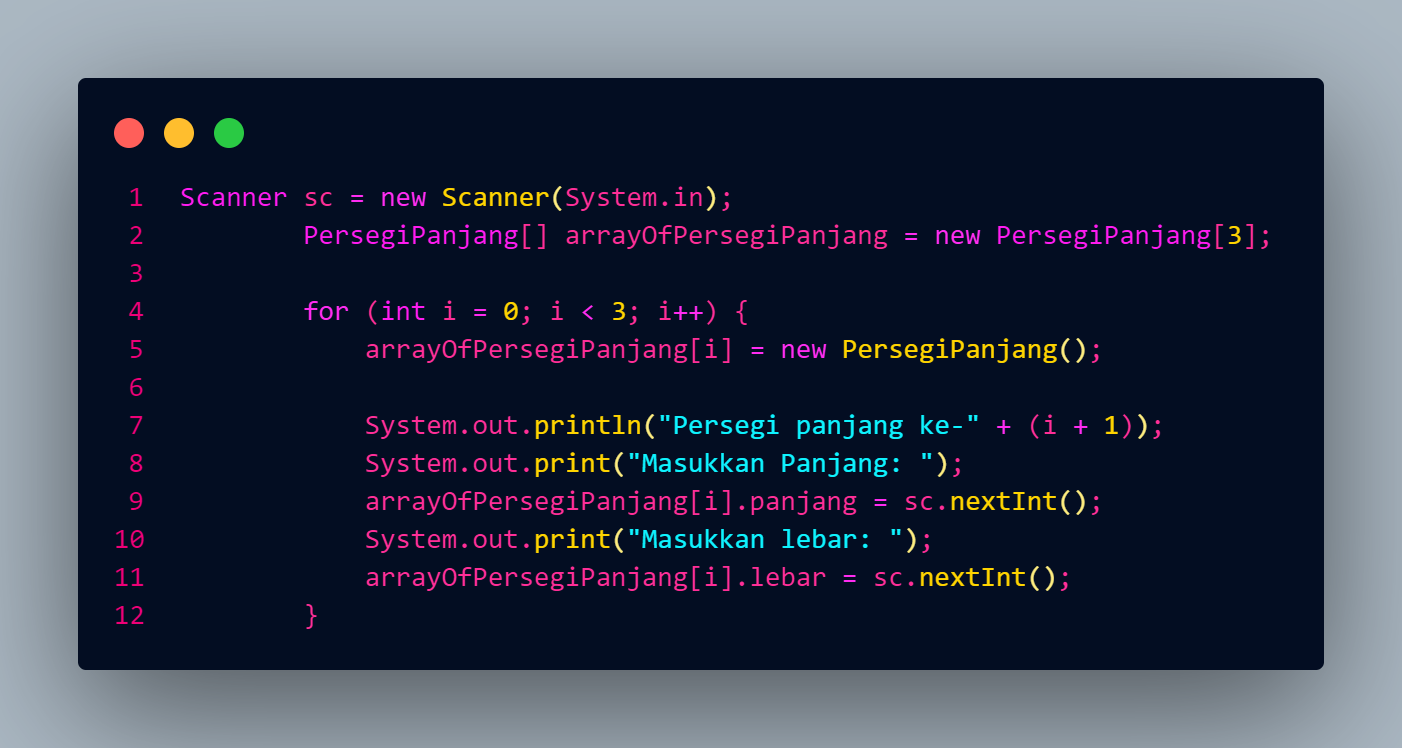


Note: Letakkan kode import dibawah kode package (jika ada).

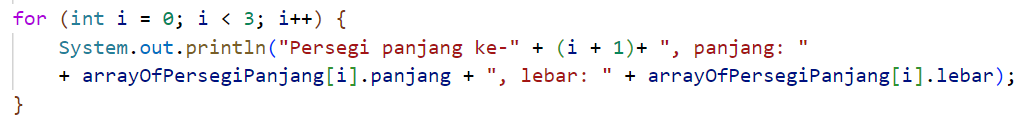


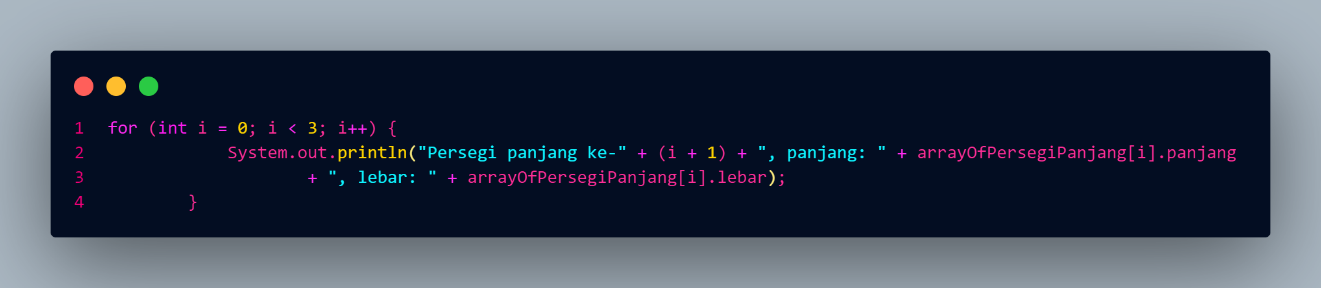
1. Pada praktikum 3.2 poin nomor 4, modifikasi kode program sebagai berikut. Buat objek **Scanner** untuk menerima input, kemudian lakukan looping untuk menerima informasi panjang dan lebar:





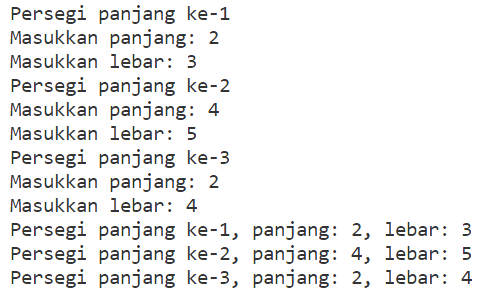
1. Pada praktikum 3.2 poin nomor 5, modifikasi kode program sebagai berikut. Lakukan looping untuk mengakses elemen array dan menampilkan informasinya ke layar:

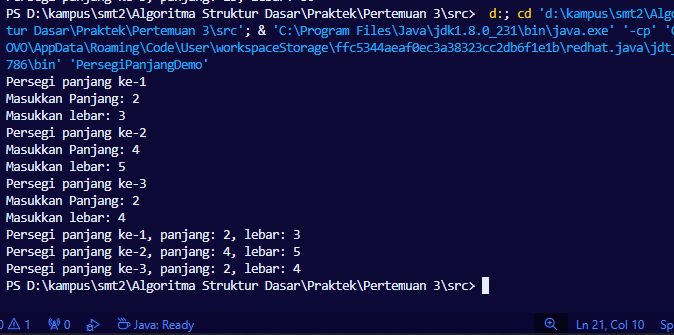




1. Run program dan amati hasilnya.
   * 1. Verifikasi Hasil Percobaan

Contoh verifikasi hasil percobaan ini.

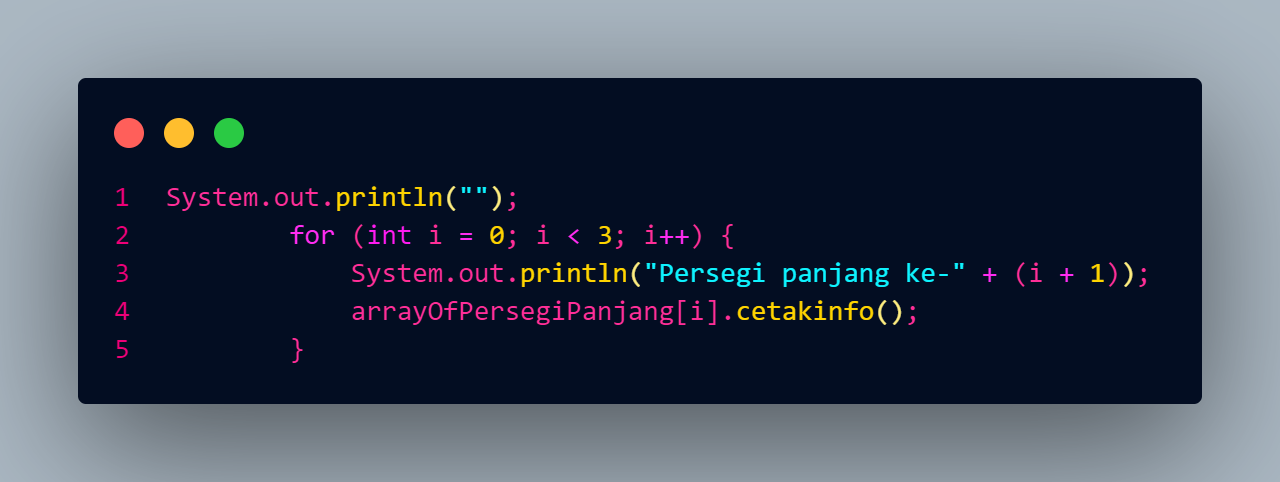


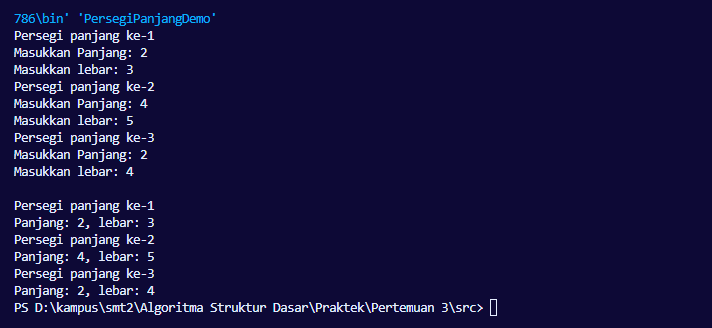


* + 1. Pertanyaan

1. Tambahkan method cetakInfo() pada class PersegiPanjang kemudian modifikasi kode program pada langkah no 3.







1. Misalkan Anda punya **array baru** bertipe array of PersegiPanjang dengan nama **myArrayOfPersegiPanjang**. Mengapa kode berikut menyebabkan error?

PersegiPanjang[] myArrayOfPersegiPanjang = new PersegiPanjang[100];

myArrayOfPersegiPanjang[5].panjang = 20;

**Jawab:**

* Array myArrayOfPersegiPanjang belum diinisialisasi.
* Saat kode dijalankan dan mencoba mengakses elemen array yang belum diinisialisasi, Java akan menghasilkan error.

1. Constructor Berparameter

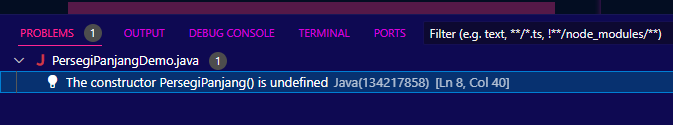
Pada praktikum ini kita akan melakukan pengoperasian matematika beberapa atribut pada masing-masing anggota array.

* + 1. Langkah-langkah Percobaan

1. Pada class PersegiPanjang, tambahkan constructor berparameter

|  |
| --- |
|  |

1. Run program kemudian amati hasilnya.

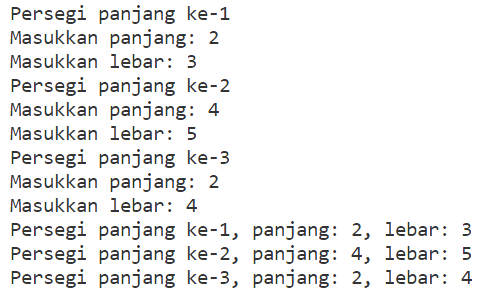
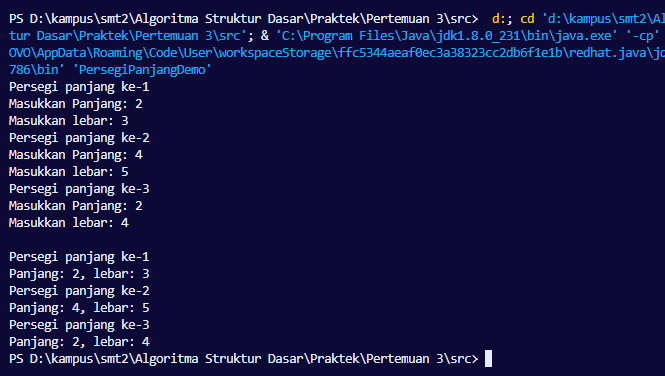


1. Modifikasi class PersegiPanjangDemo sehingga instansiasi dilakukan menggunakan constructor berparameter

|  |
| --- |
|  |

1. Run progam dan amati hasilnya.
   * 1. Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



* + 1. Pertanyaan

1. Mengapa terjadi compile error pada langkah no 2?

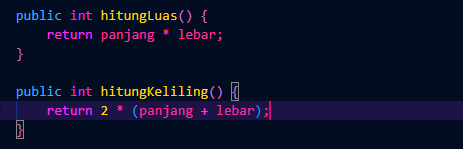
**Jawab:** karena memiliki konstruktor dengan argumen, tetapi mencoba menyebutnya tanpa argument, perlu memanggilnya dengan memberikan argumen yang diperlukan saat membuat objek kelas.

1. Apakah suatu class dapat memiliki lebih dari 1 constructor? Jika iya, berikan contohnya

**Jawab:** ya, suatu class dapat memiliki lebih dari satu constructor. Contohnya seperti class PersegiPanjang Constructor tanpa parameter, yang menginisialisasi panjang dan lebar dengan nilai default. Constructor dengan dua parameter, yang menginisialisasi panjang dan lebar dengan nilai yang diberikan.

1. Tambahkan method hitungLuas() dan hitungKeliling()

**Jawab:**

****

1. Lakukan modifikasi kode program untuk menampilkan luas dan keliling dengan contoh output sebagai berikut:

|  |
| --- |
|  |

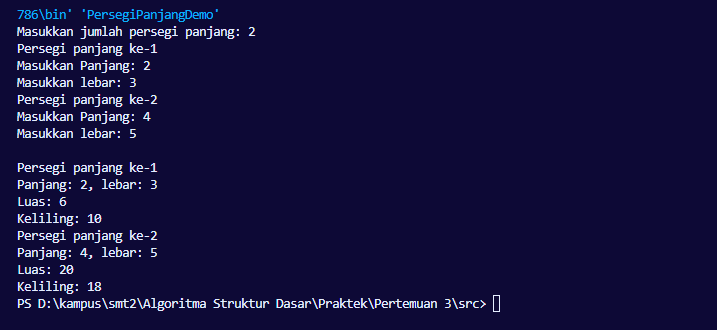
**Jawab:**

****

1. Modifikasi kode program pada praktikum 3.3 agar length array ditentukan dari user melalui input dengan Scanner

**Jawab:**





* 1. Tugas

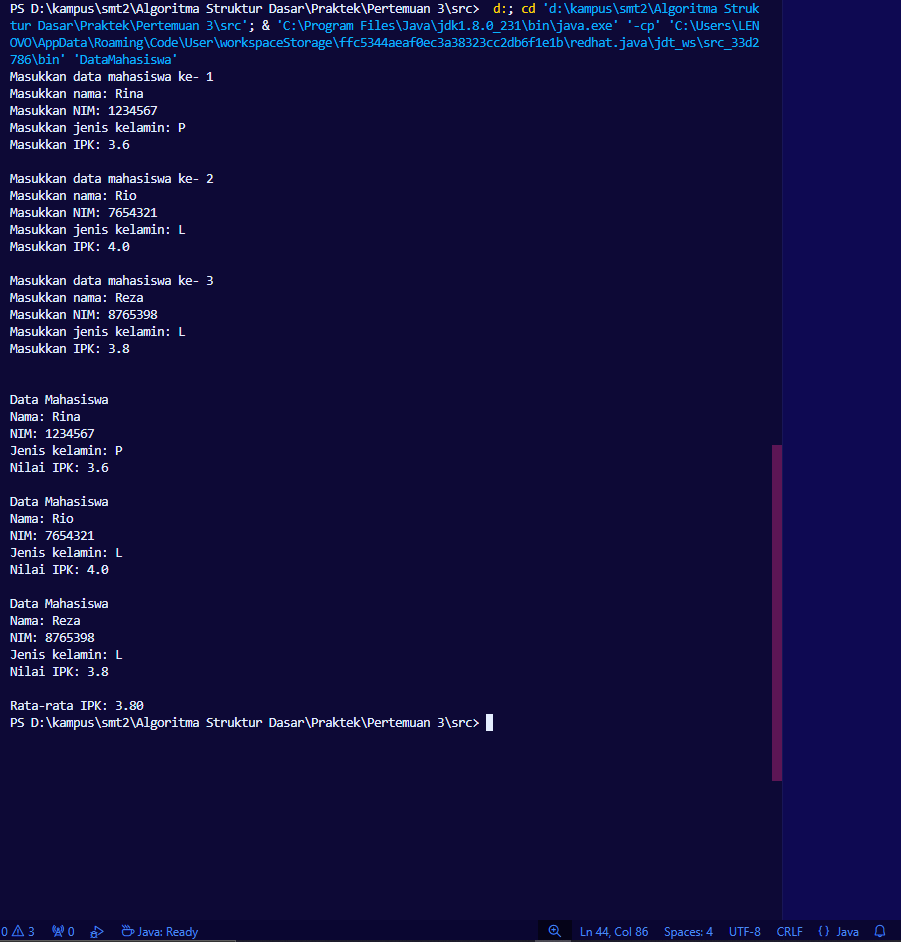
1. Sebuah kampus membutuhkan program untuk menampilkan informasi mahasiswa berupa nama, NIM, jenis kelamin dan juga IPK mahasiswa. Program dapat menerima input se mua informasi tersebut, kemudian menampilkanya kembali ke user. Tambahkan informasi rata-rata IPK pada bagian akhir. Gunakan looping dengan **FOR** untuk pembuatan object. Gunakan looping dengan **FOREACH** untuk pencetakan data ke layar.

Contoh output sebagai berikut:

|  |
| --- |
| Masukkan data mahasiswa ke- 1  Masukkan nama:Rina  Masukkan NIM:1234567  Masukkan jenis kelamin:P  Masukkan IPK:3.6  Masukkan data mahasiswa ke- 2  Masukkan nama:Rio  Masukkan NIM:7654321  Masukkan jenis kelamin:L  Masukkan IPK:4.0  Masukkan data mahasiswa ke- 3  Masukkan nama:Reza  Masukkan NIM:8765398  Masukkan jenis kelamin:L  Masukkan IPK:3.8  Data Mahasiswa ke-1  Nama: Rina  Nim: 1234567  Jenis kelamin: P  Nilai IPK: 3.6  Data Mahasiswa ke-2  nama: Rio  NIM: 7654321  Jenis kelamin: L  Nilai IPK: 4.0  Data Mahasiswa ke-3  nama: Reza  NIM: 8765398  Jenis kelamin: L  Nilai IPK: 3.8  Rata-rata IPK: 3.8 |

**** **Jawab:**





1. Implementasikan tugas ASD Teori no 1 ke dalam kode program. Buatlah array of objects berdasarkan input dari user dan tampilkan data seperti pada Tugas no 1.
2. package ClinicSystem;
3. import java.util.\*;
4. public class clinic {
5. private final Scanner input;
6. private static final int MAX\_PASIEN = 100;
7. private Pasien[] pasienList = new Pasien[MAX\_PASIEN];
8. private static UserCredentials[] credentials;
9. private int jumlahPasien = 0;
10. private boolean lanjutInput = true;
11. private static int idOtomatis = 1;
12. public clinic(Scanner input) {
13. this.input = input;
14. credentials = new UserCredentials[] { *// Initialize credentials array (optional)*
15. new UserCredentials("admin", "password")
16. };
17. }
18. public static class UserCredentials {
19. private String username;
20. private String password;
21. public UserCredentials(String username, String password) {
22. this.username = username;
23. this.password = password;
24. }
25. public String getUsername() {
26. return username;
27. }
28. public String getPassword() {
29. return password;
30. }
31. }
32. public static class Pasien {
33. private String noKartu;
34. private String nama;
35. private String noKtp;
36. private String alamat;
37. private String noTelp;
38. public Pasien(String noKartu, String nama, String noKtp, String alamat, String noTelp) {
39. this.noKartu = noKartu;
40. this.nama = nama;
41. this.noKtp = noKtp;
42. this.alamat = alamat;
43. this.noTelp = noTelp;
44. }
45. public String getNoKartu() {
46. return noKartu;
47. }
48. public String getNama() {
49. return nama;
50. }
51. public String getNoKtp() {
52. return noKtp;
53. }
54. public String getAlamat() {
55. return alamat;
56. }
57. public String getNoTelp() {
58. return noTelp;
59. }
60. }
61. boolean adminLogin() {
62. System.out.println("==========================================================================");
63. System.out.print("Masukkan username admin: ");
64. String inputUsername = input.nextLine();
65. System.out.print("Masukkan password admin: ");
66. String inputPassword = input.nextLine();
67. System.out.println("==========================================================================");
68. if (credentials == null || credentials.length == 0) { *// Handle empty credentials array*
69. return false;
70. }
71. for (UserCredentials credential : credentials) {
72. if (credential.getUsername().equals(inputUsername) && credential.getPassword().equals(inputPassword)) {
73. return true;
74. }
75. }
76. return false;
77. }
78. void registrasiPasienBaru() {
79. System.out.println("");
80. System.out.println("");
81. System.out.println("KLINIK X - REGISTRASI PASIEN");
82. while (lanjutInput) {
83. if (jumlahPasien >= MAX\_PASIEN) {
84. System.out.println("Jumlah pasien sudah mencapai batas maksimal.");
85. return;
86. }
87. String noKartu = generateKartuNumber();
88. System.out.println("==========================================================================");
89. System.out.println("No Kartu: " + noKartu);
90. Pasien newPasien = createPasienObject(noKartu);
91. System.out.println("--------------------------------------------------");
92. System.out.println("DATA BERHASIL DISIMPAN!");
93. printPasienDetails(newPasien);
94. System.out.println("==========================================================================");
95. idOtomatis++;
96. lanjutInput = lanjutInputPasien();
97. }
98. }
99. boolean lanjutInputPasien() {
100. System.out.println("Apakah ingin memasukkan data pasien lagi? (Y/T)");
101. String pilihan = input.nextLine().toUpperCase();
102. return pilihan.equals("Y");
103. }
104. Pasien createPasienObject(String noKartu) {
105. System.out.print("Nama: ");
106. String nama = input.nextLine();
107. System.out.print("No KTP: ");
108. String noKtp = input.nextLine();
109. System.out.print("Alamat: ");
110. String alamat = input.nextLine();
111. System.out.print("No. Telp: ");
112. String noTelp = input.nextLine();
113. return new Pasien(noKartu, nama, noKtp, alamat, noTelp);
114. }
115. String generateKartuNumber() {
116. return "P" + String.format("%04d", idOtomatis);
117. }
118. void printPasienDetails(Pasien pasien) {
119. System.out.println("No Kartu: " + pasien.getNoKartu());
120. System.out.println("Nama: " + pasien.getNama());
121. System.out.println("No KTP: " + pasien.getNoKtp());
122. System.out.println("Alamat: " + pasien.getAlamat());
123. System.out.println("No. Telp: " + pasien.getNoTelp());
124. }
125. }

