

---

Nama	: Muhammad Hazel Zahran Saleh
NIM	: 245150707111004
Kelas	: TI - A
Bab	: Constructor dan Instance Method
Asisten	: 1. Ketut Bagus Wedanta Ananda Murti 2. Gantang Satria Yudha.

---

## LANGKAH 1

### A. Soal

Tugas Praktikum

Dari jawaban bab 1, implementasikan hal-hal berikut di dalam class :

1. Tambahkan attribute ‘sinopsis’ (minimal 10 kata),
2. Constructor pada semua attribute yang ada,
3. Method hitung jumlah kata pada atribut synopsis,
4. Method cek tingkat kesamaan, dengan parameter object lain dan nilai kembalian 0-100% sesuai kesamaan attribute kedua object,
5. Method copy yang mengembalikan object yang nilainya sama.

### B. Screenshoot

```

class Jual {
    private String jJudul;
    private String jPenulis;
    private String jKategori;
    private String jSinopsis;
    private Buku next;
}

public void tambahJual(String jJudul, String jPenulis, String jKategori, String jSinopsis) {
    this.jJudul = jJudul;
    this.jPenulis = jPenulis;
    this.jKategori = jKategori;
    this.jSinopsis = jSinopsis;
    this.next = null;
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public void setNext(Buku next) {
    this.next = next;
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public Buku getNext() {
    return next;
}
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public void tampilkanInfo() {
    System.out.println("Judul: " + jJudul);
    System.out.println("Penulis: " + jPenulis);
    System.out.println("Kategori: " + jKategori);
    System.out.println("Sinopsis: " + jSinopsis);
    System.out.println("Jumlah kata dalam sinopsis: " + hitungJumlahKata(jSinopsis));
    System.out.println(".....");
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public String getJatuhTempat() {
    return jKategori;
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public int hitungJumlahKata(String s) {
    return s.split(" ").length;
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public double cekRataan(Buku buku) {
    int totalBintang = 0; // Jumlah penulis, kategori, sinopsis
    int bintang = 0;

    if (buku.jJudul.equals(bukubuku.jJudul)) bintang++;
    if (buku.jPenulis.equals(bukubuku.jPenulis)) bintang++;
    if (buku.jKategori.equals(bukubuku.jKategori)) bintang++;
    if (buku.jSinopsis.equals(bukubuku.jSinopsis)) bintang++;

    return (bintang / (double) totalBintang) * 100;
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public Buku copy() {
    return new Buku(jJudul, jPenulis, jKategori, jSinopsis);
}
}

Cukup Refactor | Sipilin
class Buku {
    private String nama;
    private Buku firstBuku;

    public Category(string nama) {
        this.nama = nama;
        this.firstBuku = null;
    }

    Cukup Refactor | Sipilin Generate JavaDoc | X
    public string getNama() {
        return nama;
    }

    Cukup Refactor | Sipilin Generate JavaDoc | X
    public void tambahBuku(Buku buku) {
        if (firstBuku == null) {
            firstBuku = buku;
        } else {
            Buku temp = firstBuku;
            while (temp.getNext() != null) {
                temp = temp.getNext();
            }
            temp.setNext(buku);
        }
    }
}

Cukup Refactor | Sipilin Generate JavaDoc | X
public void tampilkanInfo() {
    System.out.println("Judul: " + nama);
    System.out.println(".....");
    Buku temp = firstBuku;
    while (temp != null) {
        System.out.println(temp);
        temp = temp.getNext();
    }
}
}

Cukup Refactor | Sipilin
class Persestakan {
    private Category[] daftarKategori;
    private int jumlahKategori;

    public Persestakan() {
        daftarKategori = new Category[10];
        jumlahKategori = 0;
    }

    Cukup Refactor | Sipilin Generate JavaDoc | X
    public void tambahKategori(Category masingKategori) {
        if (jumlahKategori < daftarKategori.length) {
            daftarKategori[jumlahKategori] = new Category(masingKategori);
            jumlahKategori++;
        }
    }

    Cukup Refactor | Sipilin Generate JavaDoc | X
    public void tambahBukuKategori(String masingKategori, Buku buku) {
        for (int i = 0; i < jumlahKategori; i++) {
            if (daftarKategori[i].getNama().equals(masingKategori)) {
                daftarKategori[i].tambahBuku(buku);
                return;
            }
        }
    }
}

```

```

    Codename Refactor | English | Generate JavaDoc | X
    public void tampilkanSemua() {
        for (int i = 0; i < jumlahKategori; i++) {
            daftarKategori[i].tampilkanInfo();
        }
    }

    Codename Refactor | English | Generate JavaDoc | X
    public class PraktisPematerihali {
        Codename Refactor | English | Generate JavaDoc | X
        public static void main(String[] args) {
            Perpustakaan perpustakaan = new Perpustakaan();

            String[] kategoriBuku = {"Teknologi", "Vilasfat", "Sejarah", "Agama", "Psikologi", "Politik", "Fiksi"};
            for (String kategori : kategoriBuku) {
                perpustakaan.tambahKategori(kategori);
            }

            Buku[] daftarBuku = [
                new Buku("Kecerdasan Buatan", "John McCarthy", "Teknologi", "Teknologi AI berkembang pesat dalam era modern ini."),
                new Buku("Data Science", "Jake Vanderplas", "Teknologi", "Analisis data menjadi elemen penting dalam dunia digital."),
                new Buku("Programmme Java", "Ivanec S. Tocino", "Teknologi", "Java adalah bahasa pemrograman yang sangat populer."),
                new Buku("Taringan Komputer", "Andrew S. Tanenbaum", "Teknologi", "Taringan komputer menjadi tulang punggung komunikasi modern."),
                new Buku("Komputer Bisnis", "Axel Gantek", "Teknologi", "Taringan komputer menjadi tulang punggung aseli abt."),
                //ini teknologi
                new Buku("Hid dan water", "Martin Heidegger", "Filosofat", "Kecintuan hidup dalam ketidakmodernan."),
                new Buku("Republik", "Plato", "Filosofat", "Filosofi tentang kehidupan dalam dunia segera."),
                new Buku("al arikie azmi", "Dilan", "Filosofat", "gutau alesie pegeen belli truk."),
                new Buku("apa itu apa?", "Pitraus", "Filosofat", "manan buah khaldi itu erak utz aseli."),
                new Buku("aplaiki ya?", "Jokowi", "Filosofat", "Bingung nethamp filosofat itu apa."),
                //filosofat
                new Buku("Sejarah osiris", "E. H. Krombach", "Sejarah", "Ringkasan sejarah dunia dalam berbagai era."),
                new Buku("Sapiens", "Yuval Noah Harari", "Sejarah", "Sejarah umat manusia dari zanes purba hingga kini."),
                new Buku("Sipari peduli pria-wati", "Sipari", "Sejarah", "Memiliki peduli tapi ga bilang bilang."),
                new Buku("Sipari peduli hutnik", "Sipari", "Sejarah", "Sejarah yang capra ternyata punya hati."),
                new Buku("Sipari ada typo", "Sipari", "Sejarah", "Sejarah umat manusia garong itu sekerewa ada typo."),
                //sejarah
                new Buku("Kitab Taufiq", "Muhammad bin Abdul Wahab", "Agama", "Perjelanan tentang konsep taufiq dalam Islam."),
                new Buku("Tiyaduk Shalihin", "Abu Zakaria ar-Ramli", "Agama", "Kumpulan hadits tentang moral dan etika Islam."),
                new Buku("Taufiq ibnu Kathir", "Ibnu Kathir", "Agama", "Penafsiran Al-Qur'an yang asli dan akurat."),
                new Buku("Al-Mu'awin", "Ismail Syifii", "Agama", "Mudahah berbagai aspek hukum Islam (fiqh), mulai dari ibadah, mazhab, hingga sistem peradilan Islam."),
                new Buku("Asal Muslim setara-kartu", "Arzada Shafiq Maslamah", "Agama", "Panduan praktis tentang islam turis dengan mudah, simak dalam Islam, dan turut serta hidup Islam."),
                //agama
                new Buku("Psikologi sosial", "David Myers", "Psikologi", "Anggapan interaksi sosial mempengaruhi perilaku individu."),
                new Buku("Discret", "Carol S. Dweck", "Psikologi", "Pentingnya pola pikir diri seorang keberhasilan."),
                new Buku("Stan", "Chris K. Wylie", "Psikologi", "Pentingnya diri dan searah keberhasilan."),
                new Buku("Silay", "Izzah bin Zainal", "Psikologi", "Testing tujuan dan cita-cita hidup."),
                new Buku("Kitchen House Mystery", "Elizabeth Haynes", "Psikologi", "Kisah prioritas dalam menjalankan suatu kebiasaan"),
                //psikologis
                new Buku("The Prince", "Niccolò Machiavelli", "Politik", "Strategi kepemimpinan dalam dunia politik."),
                new Buku("Das Kapital", "Karl Marx", "Politik", "Analisis monasi dan sosial dalam sistem kapitalisme."),
                new Buku("Tana Yang Selap Butan Indonesia", "Luket Pardiyana", "Politik", "Alasan-alasan Luket atas ekonomi kolonial yang ada."),
                new Buku("Tata Tidak Berdati", "Nelgekt", "Politik", "Cerita tentang perjuangan-pelabuh turang cilik-nasional."),
                new Buku("Pilih Hunting Yo Waria Merah", "Yosep Amara", "Politik", "Analisa partai politik di Indonesia."),
                //politik
                new Buku("Harry Potter", "J.K. Rowling", "Fiksi", "Petualangan magis di dunia仙语 yang luar biasa."),
                new Buku("Silent Songong Cinta Indigo", "Tok Sabang", "Fiksi", "Mengikuti cerita hubungan cintanya dengan yang tidak baik-baik saja."),
                new Buku("Tembaga-Kembang Bawali", "Hsi", "Fiksi", "Andara kendar yang menyayangi teman teman menjadi superhero untuk menuntaskan kejahatan."),
                new Buku("Parable", "Kiran Thapar", "Fiksi", "Kisah seorang yang mengayuh nasihat hidup yang berlalu."),
                new Buku("Siti Tergelap Sugra", "Wulan Thalina", "Fiksi", "Mengikuti tentang cinta gelap batin metropolitan tercinta yaitu Jakarta."),
                //fiksi
            ];
        }

        for (Buku buku : daftarBuku) {
            perpustakaan.tambahBukuKeKategori(buku.getKategori(), buku);
        }

        perpustakaan.tampilkanSemua();
    }
}

```

## C. Syntax

	<pre> class Buku {     private String judul;     private String penulis;     private String kategori;     private String sinopsis;     private Buku next;      public Buku(String judul, String penulis, String kategori, String sinopsis) {         this.judul = judul;         this.penulis = penulis;         this.kategori = kategori;         this.sinopsis = sinopsis;         this.next = null;     }      public void setNext(Buku next) {         this.next = next;     }      public Buku getNext() {         return next;     }      public void tampilkanInfo() {         System.out.println("Judul: " + judul);     } } </pre>
--	---

```

        System.out.println("Penulis: " + penulis);
        System.out.println("Kategori: " + kategori);
        System.out.println("Sinopsis: " + sinopsis);
        System.out.println("Jumlah Kata dalam Sinopsis: " +
hitungJumlahKataSinopsis());
        System.out.println("-----");
    }

    public String getKategori() {
        return kategori;
    }

    public int hitungJumlahKataSinopsis() {
        return sinopsis.split(" ").length;
    }

    public double cekKesamaan(Buku bukuLain) {
        int totalAtribut = 4; // Judul, Penulis, Kategori,
Sinopsis
        int kesamaan = 0;

        if (this.judul.equals(bukuLain.judul)) kesamaan++;
        if (this.penulis.equals(bukuLain.penulis))
kesamaan++;
        if (this.kategori.equals(bukuLain.kategori))
kesamaan++;
        if (this.sinopsis.equals(bukuLain.sinopsis))
kesamaan++;

        return (kesamaan / (double) totalAtribut) * 100;
    }

    public Buku copy() {
        return new Buku(judul, penulis, kategori, sinopsis);
    }
}

class Kategori {
    private String nama;
    private Buku firstBuku;

    public Kategori(String nama) {
        this.nama = nama;
        this.firstBuku = null;
    }

    public String getNama() {
        return nama;
    }

    public void tambahBuku(Buku buku) {
        if (firstBuku == null) {
            firstBuku = buku;
        } else {
            Buku temp = firstBuku;
            while (temp.getNext() != null) {
                temp = temp.getNext();
            }
            temp.setNext(buku);
        }
    }

    public void tampilkanBuku() {

```

```

        System.out.println("Kategori: " + nama);
        System.out.println("=====");
        Buku temp = firstBuku;
        while (temp != null) {
            temp.tampilkanInfo();
            temp = temp.getNext();
        }
    }

    class Perpustakaan {
        private Kategori[] daftarKategori;
        private int jumlahKategori;

        public Perpustakaan() {
            daftarKategori = new Kategori[10];
            jumlahKategori = 0;
        }

        public void tambahKategori(String namaKategori) {
            if (jumlahKategori < daftarKategori.length) {
                daftarKategori[jumlahKategori] = new
Kategori(namaKategori);
                jumlahKategori++;
            }
        }

        public void tambahBukuKeKategori(String namaKategori,
Buku buku) {
            for (int i = 0; i < jumlahKategori; i++) {
                if
(daftarKategori[i].getNama().equals(namaKategori)) {
                    daftarKategori[i].tambahBuku(buku);
                    return;
                }
            }
            System.out.println("Kategori " + namaKategori + "
tidak ditemukan!");
        }

        public void tampilkanSemuaBuku() {
            for (int i = 0; i < jumlahKategori; i++) {
                daftarKategori[i].tampilkanBuku();
            }
        }
    }

    public class PraktikumPemlanAwal2 {
        public static void main(String[] args) {
            Perpustakaan perpustakaan = new Perpustakaan();

            String[] kategoriBuku = {"Teknologi", "Filsafat",
"Sejarah", "Agama", "Psikologi", "Politik", "Fiksi"};

            for (String kategori : kategoriBuku) {
                perpustakaan.tambahKategori(kategori);
            }

            Buku[] daftarBuku = {
                new Buku("Kecerdasan Buatan", "John McCarthy",
"Teknologi", "Teknologi AI berkembang pesat dalam era modern ini."),
                new Buku("Data Science", "Jake VanderPlas",
"Data Science", "Data Science")
            };
        }
    }
}

```

"Teknologi", "Analisis data menjadi elemen penting dalam dunia digital."),  
    new Buku("Pemrograman Java", "James Gosling",  
"Teknologi", "Java adalah bahasa pemrograman yang sangat populer."),  
    new Buku("Jaringan Komputer", "Andrew S. Tanenbaum", "Teknologi", "Jaringan komputer menjadi tulang punggung komunikasi modern."),  
    new Buku("komputer bingung", "hazel gantenek",  
"Teknologi", "Jaringan komputer menjadi bingung aseli dah."),  
    //ini teknologi  
    new Buku("Ada dan Waktu", "Martin Heidegger",  
"Filsafat", "Eksistensialisme dalam kehidupan modern."),  
    new Buku("Republik", "Plato", "Filsafat",  
"Filosofi tentang keadilan dalam suatu negara."),  
    new Buku("al asikin azzah", "dilan", "Filsafat",  
"gatau males pengen beli truk."),  
    new Buku("apa itu apa?", "Piraun", "Filsafat",  
"makan buah khuldi itu enak bat aseli."),  
    new Buku("apalagi ya?", "jospar", "Filsafat",  
"bingung terhadap filsafat itu apa."),  
    //filsafat  
    new Buku("Sejarah Dunia", "E. H. Gombrich",  
"Sejarah", "Ringkasan sejarah dunia dalam berbagai era."),  
    new Buku("Sapiens", "Yuval Noah Harari",  
"Sejarah", "Sejarah umat manusia dari zaman purba hingga kini."),  
    new Buku("Sapri pemilik piramid", "sapri",  
"Sejarah", "memiliki piramid tapi ga bilang bilang."),  
    new Buku("Sapra pemilik hatiku", "sapra",  
"Sejarah", "Sejarah sang sapra ternyata punya hati."),  
    new Buku("Sapru suka typo", "sapru", "Sejarah",  
"Sejarah umat manusia ganteng itu sebenarnya suka typo."),  
    //sejarah  
    new Buku("Kitab Tauhid", "Muhammad bin Abdul Wahhab", "Agama", "Penjelasan tentang konsep tauhid dalam Islam."),  
    new Buku("Riyadhus Shalihin", "Abu Zakaria an-Nawawi", "Agama", "Kumpulan hadits tentang moral dan etika Islam."),  
    new Buku("Tafsir Ibnu Katsir", "Ibnu Katsir",  
"Agama", "Penafsiran Al-Quran yang mendalam."),  
    new Buku("Al-Umm", "Imam Syafi'i", "Agama",  
"membahas berbagai aspek hukum Islam (fiqh), mulai dari ibadah, muamalah, hingga sistem peradilan Islam."),  
    new Buku("Bekal Muslim Sehari-hari", "Ustadz Khalid Basalamah", "Agama", "panduan praktis tentang ibadah harian seorang Muslim, termasuk doa-doa sunnah, adab dalam Islam, dan tuntunan hidup Islami."),  
    //agama  
    new Buku("Psikologi Sosial", "David Myers",  
"Psikologi", "Bagaimana interaksi sosial mempengaruhi perilaku individu."),  
    new Buku("Mindset", "Carol S. Dweck",  
"Psikologi", "Pentingnya pola pikir dalam meraih kesuksesan."),  
    new Buku("otaku", "shin kam pri", "Psikologi",  
"Pentingnya dirimu dalam meraih kesuksesan."),  
    new Buku("ikigay", "saskeh bin narto",  
"Psikologi", "Tentang tujuan dan cita-cita hidup."),  
    new Buku("Eishen hower matriks", "eishen hower",  
"Psikologi", "skala prioritas dalam menjalankan suatu

```

        kewajiban"),
                //psikologggggg
                new Buku("The Prince", "Niccolò Machiavelli",
"Politik", "Strategi kepemimpinan dalam dunia politik."),
                new Buku("Das Kapital", "Karl Marx", "Politik",
"Analisis ekonomi dan sosial dalam sistem kapitalisme."),
                new Buku("Kamu Yang Gelap Bukan Indonesia",
"Luah pandjaitan", "Politik", "Alasan-alasan luhut atas
seluruh kebijakan yang ada."),
                new Buku("tikus tikus berdasi", "zelgans",
"Politik", "Kritikan terhadap pejabat-pejabat korup
diindonesia."),
                new Buku("pilih banteng ya warna merah", "roman
kamaru", "Politik", "Analisis partai politik di
indonesia."),
                //politikk
                new Buku("Harry Potter", "J.K. Rowling",
"Fiksi", "Petualangan magis di dunia sihir yang luar
biasa."),
                new Buku("Sikancil songong sama buaya", "Tok
Dalang", "Fiksi", "Mengisahkan cerita hubungan kancil dan
buaya yang tidak baik-baik saja."),
                new Buku("Kembara-Kembar Nackal", "Ros",
"Fiksi", "Saudara kembar yang mempunyai tekad untuk menjadi
superhero untuk menuntaskan kejahatan."),
                new Buku("Parable", "Brian Khrisna", "Fiksi",
"Kisah seseorang yang mempunyai masalah hidup yang
berliku."),
                new Buku("Sisi Tergelap Surga", "Brian Khrisna",
"Fiksi", "Mengisahkan tentang sisi gelap kota metropolitan
ternama yaitu jakarta."),
                //fiksi
            };
        }

        for (Buku buku : daftarBuku) {
            perpustakaan.tambahBukuKeKategori(buku.getKatego
ri(), buku);
        }

        perpustakaan.tampilkanSemuaBuku();
    }
}

```

## D. Penjelasan

```

Codeium: Refactor | Explain
1 class Buku {
2     private String judul;
3     private String penulis;
4     private String kategori;
5     private String sinopsis;
6     private Buku next;
7
8     public Buku(String judul, String penulis, String kategori, String sinopsis) {
9         this.judul = judul;
10        this.penulis = penulis;
11        this.kategori = kategori;
12        this.sinopsis = sinopsis;
13        this.next = null;
14    }
15

```

### Penjelasan:

- Kelas Buku digunakan untuk menyimpan informasi mengenai sebuah buku seperti **judul**, **penulis**, **kategori**, dan **sinopsis**.
- **next** adalah penunjuk ke buku berikutnya (menggunakan **Linked List**).
- Constructor ini digunakan untuk membuat objek Buku baru dengan data yang diberikan.

```
Codeium: Refactor | Explain | Generate Javadoc | X
24 public void tampilkanInfo() {
25     System.out.println("Judul: " + judul);
26     System.out.println("Penulis: " + penulis);
27     System.out.println("Kategori: " + kategori);
28     System.out.println("Sinopsis: " + sinopsis);
29     System.out.println("Jumlah Kata dalam Sinopsis: " + hitungJumlahKataSinopsis());
30     System.out.println("-----");
31 }
```

### Penjelasan:

- Metode ini digunakan untuk menampilkan informasi lengkap tentang buku.
- Menghitung jumlah kata dalam sinopsis dengan metode **hitungJumlahKataSinopsis()**.

```
Codeium: Refactor | Explain
58 class Kategori {
59     private String nama;
60     private Buku firstBuku;
61
62     public Kategori(String nama) {
63         this.nama = nama;
64         this.firstBuku = null;
65     }
}
```

### Penjelasan:

- Kelas ini menyimpan daftar buku berdasarkan **kategori**.
- **firstBuku** adalah pointer ke buku pertama dalam kategori tersebut (**Linked List**).

```
Codeium: Refactor | Explain | Generate Javadoc | X
71 public void tambahBuku(Buku buku) {
72     if (firstBuku == null) {
73         firstBuku = buku;
74     } else {
75         Buku temp = firstBuku;
76         while (temp.getNext() != null) {
77             temp = temp.getNext();
78         }
79         temp.setNext(buku);
80     }
81 }
```

### Penjelasan:

- Jika kategori masih kosong, buku pertama langsung diisi.
- Jika sudah ada buku, buku baru ditambahkan ke **bagian akhir daftar**.

```
Codeium: Refactor | Explain | Generate Javadoc | X
83 public void tampilkanBuku() {
84     System.out.println("Kategori: " + nama);
85     System.out.println("=====");
86     Buku temp = firstBuku;
87     while (temp != null) {
88         temp.tampilkanInfo();
89         temp = temp.getNext();
90     }
91 }
92 }
```

### Penjelasan:

- Metode ini akan **menampilkan semua buku** dalam kategori secara berurutan.
- Menggunakan **while loop** untuk membaca setiap buku dalam **Linked List**.

```
Codeium: Refactor | Explain
4 class Perpustakaan {
5     private Kategori[] daftarKategori;
6     private int jumlahKategori;
7
8     public Perpustakaan() {
9         daftarKategori = new Kategori[10];
10        jumlahKategori = 0;
11    }
12 }
```

### Penjelasan:

- Kelas ini mengelola semua kategori buku yang ada di perpustakaan.
- **daftarKategori** menyimpan **maksimal 10 kategori**.
- **jumlahKategori** menyimpan jumlah kategori yang telah dimasukkan.

```
Codeium: Refactor | Explain | Generate Javadoc | X
103 public void tambahKategori(String namaKategori) {
104     if (jumlahKategori < daftarKategori.length) {
105         daftarKategori[jumlahKategori] = new Kategori(namaKategori);
106         jumlahKategori++;
107     }
108 }
109 }
```

### Penjelasan:

- Metode ini menambahkan kategori baru ke array daftarKategori.
- Jika array belum penuh, kategori akan disimpan di indeks yang tersedia.

```

Codeium: Refactor | Explain | Generate Javadoc | X
10  public void tambahBukuKeKategori(String namaKategori, Buku buku) {
11      for (int i = 0; i < jumlahKategori; i++) {
12          if (daftarKategori[i].getNama().equals(namaKategori)) {
13              daftarKategori[i].tambahBuku(buku);
14              return;
15          }
16      }
17      System.out.println("Kategori " + namaKategori + " tidak ditemukan!");
18  }
19

```

### Penjelasan:

- Metode ini mencari kategori berdasarkan nama.
- Jika ditemukan, buku akan ditambahkan ke kategori tersebut.
- Jika tidak, akan muncul pesan "**Kategori tidak ditemukan!**".

```

Codeium: Refactor | Explain | Generate Javadoc | X
120  public void tampilkanSemuaBuku() {
121      for (int i = 0; i < jumlahKategori; i++) {
122          daftarKategori[i].tampilkanBuku();
123      }
124  }
125

```

### Penjelasan:

- Metode ini akan menampilkan semua buku dari setiap kategori di perpustakaan.

```

Codeium: Refactor | Explain
127  public class PraktikumPemlanAwal2 {
128      public static void main(String[] args) {
129          Perpustakaan perpustakaan = new Perpustakaan();
130
131          String[] kategoriBuku = {"Teknologi", "Filsafat", "Sejarah", "Agama", "Psikologi", "Politik", "Fiksi"};
132
133          for (String kategori : kategoriBuku) {
134              perpustakaan.tambahKategori(kategori);
135          }

```

### Penjelasan:

- Membuat objek **Perpustakaan**.
- Menambahkan daftar kategori ke perpustakaan.

```

137      Buku[] daftarBuku = {
138          new Buku("Kecerdasan Buatan", "John McCarthy", "Teknologi", "Teknologi AI berkembang pesat dalam era modern ini."),
139          new Buku("Data Science", "Jake VanderPlas", "Teknologi", "Analisis data menjadi elemen penting dalam dunia digital."),
140          new Buku("Pemrograman Java", "James Gosling", "Teknologi", "Java adalah bahasa pemrograman yang sangat populer."),
141          new Buku("Jaringan Komputer", "Andrew S. Tanenbaum", "Teknologi", "Jaringan komputer menjadi tulang punggung komunikasi modern."),
142          new Buku("komputer bingung", "hazel ganteng", "Teknologi", "Jaringan komputer menjadi bingung aseli dah."),
143          //ini teknologi
144          new Buku("Ada dan Waktu", "Martin Heidegger", "Filsafat", "Eksistensialisme dalam kehidupan modern."),
145          new Buku("Republik", "Plato", "Filsafat", "Filosofi tentang keadilan dalam suatu negara."),
146          new Buku("al asikin azzah", "dilan", "Filsafat", "gatau males pengen beli truk."),
147          new Buku("apa itu apa?", "piraun", "Filsafat", "makna buah khuldi itu enak bat aseli."),
148          new Buku("apalagi ya?", "jospar", "Filsafat", "bingung terhadap filsafat itu apa."),
149          //filsafat

```

### Penjelasan:

- Membuat daftar buku dan menambahkannya ke perpustakaan berdasarkan kategori.