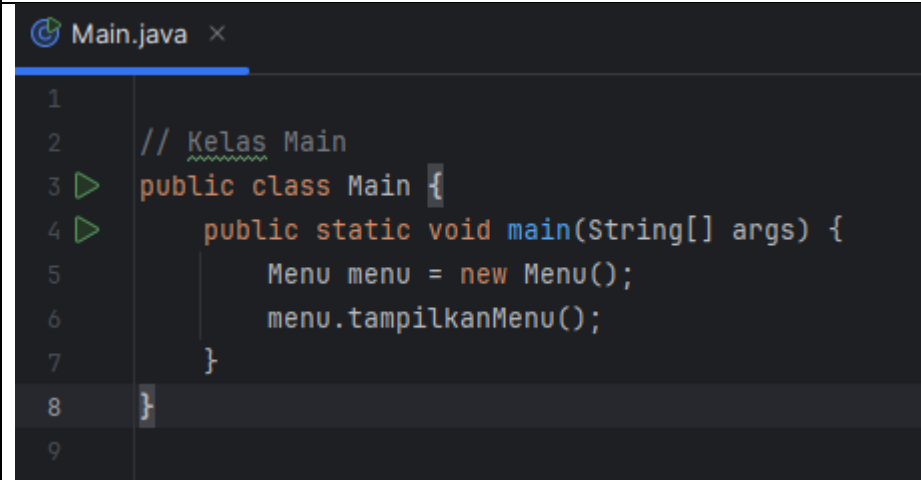
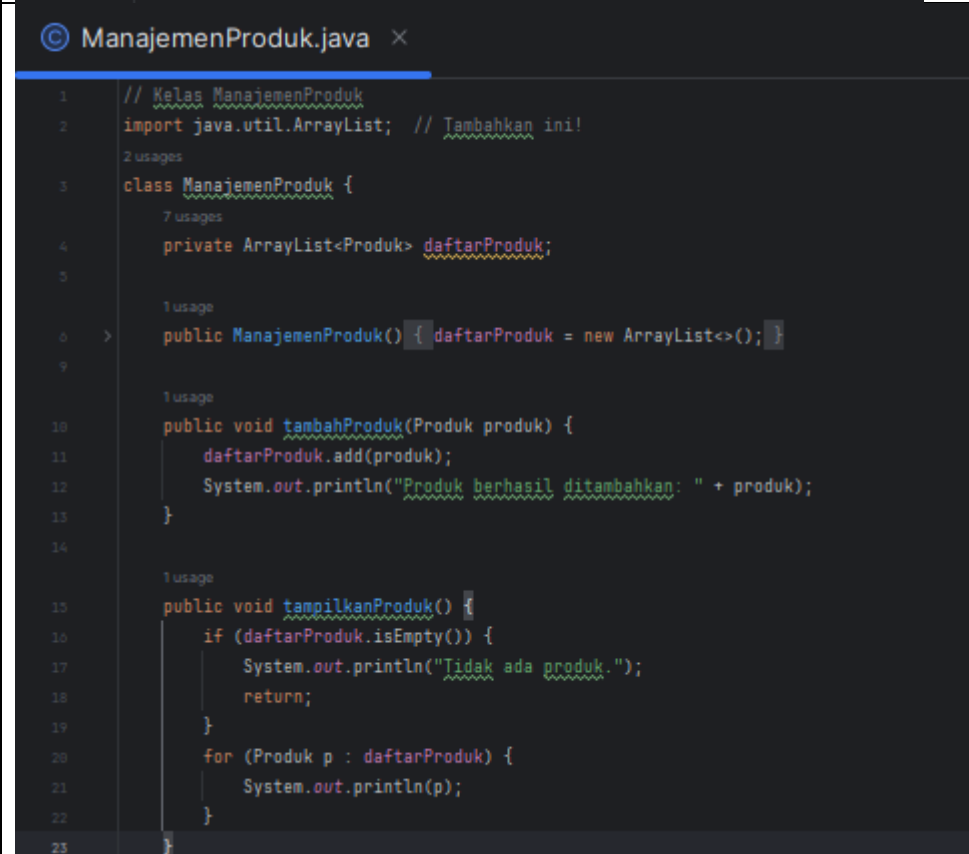


NAMA : MUHAMMAD SHANDY ALFARIZAL

NIM : 239106106

KELAS : C1'23

## SS PROGRAM


SCREENSHOT		KETERANGAN
 <pre>1 2 // Kelas Main 3 public class Main { 4     public static void main(String[] args) { 5         Menu menu = new Menu(); 6         menu.tampilkanMenu(); 7     } 8 } 9</pre>		CLASS MAIN
 <pre>1 // Kelas ManajemenProduk 2 import java.util.ArrayList; // Tambahkan ini! 3 class ManajemenProduk { 4     private ArrayList&lt;Produk&gt; daftarProduk; 5 6     public ManajemenProduk() { daftarProduk = new ArrayList&lt;&gt;(); } 7 8     public void tambahProduk(Produk produk) { 9         daftarProduk.add(produk); 10        System.out.println("Produk berhasil ditambahkan: " + produk); 11    } 12 13    public void tampilkanProduk() { 14        if (daftarProduk.isEmpty()) { 15            System.out.println("Tidak ada produk."); 16            return; 17        } 18        for (Produk p : daftarProduk) { 19            System.out.println(p); 20        } 21    } 22 } 23</pre>		CLASS MANAJEMEN PRODUK

```

1 usage
public boolean updateProduk(String kode, String namaBaru, double hargaBaru) {
    for (Produk p : daftarProduk) {
        if (p.getKode().equals(kode)) {
            p.setNama(namaBaru);
            p.setHarga(hargaBaru);
            System.out.println("Produk berhasil diperbarui.");
            return true;
        }
    }
    System.out.println("Produk tidak ditemukan.");
    return false;
}

1 usage
public boolean hapusProduk(String kode) {
    for (Produk p : daftarProduk) {
        if (p.getKode().equals(kode)) {
            daftarProduk.remove(p);
            System.out.println("Produk berhasil dihapus.");
            return true;
        }
    }
    System.out.println("Produk tidak ditemukan.");
    return false;
}
}

```

<div data-bbox="228 210 442 246">  Menu.java × </div> <pre> 1  import java.util.Scanner; 2 3  class Menu { 4      private Scanner scanner; 5      private ManajemenProduk manajemen; 6 7      public Menu() { 8          scanner = new Scanner(System.in); 9          manajemen = new ManajemenProduk(); 10     } 11 12     public void tampilkanMenu() { 13         while (true) { 14             System.out.println("\n1. Tambah Produk"); 15             System.out.println("2. Lihat Produk"); 16             System.out.println("3. Update Produk"); 17             System.out.println("4. Hapus Produk"); 18             System.out.println("5. Keluar"); 19             System.out.print("Pilih menu: "); 20             int pilihan = scanner.nextInt(); 21             scanner.nextLine(); 22 23             switch (pilihan) { 24                 case 1: 25                     tambahProduk(); 26                     break; 27                 case 2: 28                     manajemen.tampilkanProduk(); 29                     break; 30                 case 3: 31                     updateProduk(); 32                     break; </pre>	<div data-bbox="1228 194 1329 268"> CLASS MENU </div>
---	---

```

33         case 4:
34             hapusProduk();
35             break;
36         case 5:
37             System.out.println("Keluar...");
38             scanner.close();
39             return;
40         default:
41             System.out.println("Pilihan tidak valid.");
42     }
43 }
44 }
45
46 1 usage
47 private void tambahProduk() {
48     System.out.print("Masukkan kode produk: ");
49     String kode = scanner.nextLine();
50     System.out.print("Masukkan nama produk: ");
51     String nama = scanner.nextLine();
52     System.out.print("Masukkan harga produk: ");
53     double harga = scanner.nextDouble();
54     scanner.nextLine();
55
56     manajemen.tambahProduk(new Produk(kode, nama, harga));
57 }
58
59 1 usage
60 private void updateProduk() {
61     System.out.print("Masukkan kode produk yang ingin diperbarui: ");
62     String kode = scanner.nextLine();
63     System.out.print("Masukkan nama baru: ");
64     String namaBaru = scanner.nextLine();
65     System.out.print("Masukkan harga baru: ");
66     double hargaBaru = scanner.nextDouble();
67     scanner.nextLine();
68
69     manajemen.updateProduk(kode, namaBaru, hargaBaru);
70 }
71
72 1 usage
73 private void hapusProduk() {
74     System.out.print("Masukkan kode produk yang ingin dihapus: ");
75     String kode = scanner.nextLine();
76
77     manajemen.hapusProduk(kode);
78 }
79 }

```

<div><div>© Produk.java ×</div><div><div>1</div><div><div><div>❏</div></div><div>class <u>Produk</u> {</div></div></div><div><div>2</div><div><div>3 usages</div><div>private String <u>kode</u>;</div></div></div><div><div>3</div><div><div>4 usages</div><div>private String <u>nama</u>;</div></div></div><div><div>4</div><div><div>5 usages</div><div>private double <u>harga</u>;</div></div></div><div><div>5</div><div><div>4 usages</div><div>public <u>Produk</u>(String <u>kode</u>, String <u>nama</u>, double <u>harga</u>) {</div></div></div><div><div>6</div><div><div>7</div><div>    this.kode = kode;</div></div></div><div><div>7</div><div><div>8</div><div>    this.nama = nama;</div></div></div><div><div>8</div><div><div>9</div><div>    this.harga = harga;</div></div></div><div><div>9</div><div><div>10</div><div>}</div></div></div><div><div>10</div><div><div>3 usages</div><div>public <u>Produk</u>(String <u>kode</u>, String <u>nama</u>) {</div></div></div><div><div>11</div><div><div>12</div><div>    this.kode = kode;</div></div></div><div><div>12</div><div><div>13</div><div>    this.nama = nama;</div></div></div><div><div>13</div><div><div>14</div><div>    this.harga = 0.0;</div></div></div><div><div>14</div><div><div>15</div><div>}</div></div></div><div><div>15</div><div><div>1 usage</div><div>public void <u>setHarga</u>(double <u>harga</u>) {</div></div></div><div><div>16</div><div><div>17</div><div>    this.harga = harga;</div></div></div><div><div>17</div><div><div>18</div><div>}</div></div></div><div><div>18</div><div><div>no usages</div><div>public void <u>setHarga</u>(int <u>harga</u>) {</div></div></div><div><div>19</div><div><div>20</div><div>    this.harga = (double) harga;</div></div></div><div><div>20</div><div><div>21</div><div>}</div></div></div><div><div>21</div><div><div>1 usage</div><div>public void <u>setNama</u>(String <u>nama</u>) {</div></div></div><div><div>22</div><div><div>23</div><div>    this.nama = nama;</div></div></div><div><div>23</div><div><div>24</div><div>}</div></div></div><div><div>24</div><div><div>25</div><div></div></div></div><div><div>25</div><div><div>26</div><div></div></div></div><div><div>26</div><div><div>27</div><div></div></div></div><div><div>27</div><div><div>28</div><div></div></div></div><div><div>28</div><div><div>29</div><div></div></div></div></div>
--

CLASS  
PRODUK

```

30     public String getKode() {
31         return kode;
32     }
33
34     3 usages
35     public String getNama() {
36         return nama;
37     }
38
39     3 usages
40     public double getHarga() {
41         return harga;
42     }
43
44     // toString tetap diisi meskipun hanya placeholder
45     3 overrides
46     public String toString() {
47         return "Produk";
48     }
49
50     }
51
52     // Subclass BatikTulis
53     no usages
54     class BatikTulis extends Produk {
55         4 usages
56         private String teknik;
57
58         no usages
59         public BatikTulis(String kode, String nama, double harga, String teknik) {
60             super(kode, nama, harga);
61             this.teknik = teknik;
62         }
63     }

```

```

no usages
58     public BatikTulis(String kode, String nama, String teknik) {
59         super(kode, nama);
60         this.teknik = teknik;
61     }
62
63     no usages
64     public String getTeknik() {
65         return teknik;
66     }
67
68     @Override
69     public String toString() {
70         return "Kode: " + getKode() + ", Nama: " + getName() + ", Harga: " + getHarga() + ", 
71     }
72
73     // Subclass BatikCap
74     no usages
75     class BatikCap extends Produk {
76         4 usages
77         private String jenisCetakan;
78
79         no usages
80         public BatikCap(String kode, String nama, double harga, String jenisCetakan) {
81             super(kode, nama, harga);
82             this.jenisCetakan = jenisCetakan;
83         }
84
85         no usages
86         public BatikCap(String kode, String nama, String jenisCetakan) {
87             super(kode, nama);
88             this.jenisCetakan = jenisCetakan;
89         }
90
91         no usages
92         public String getJenisCetakan() { return jenisCetakan; }
93
94         @Override
95         public String toString() {
96             return "Kode: " + getKode() + ", Nama: " + getName() + ", Harga: " + getHarga() + ", Jenis Cetakan: " + jenisCetakan;
97         }
98     }
99
100     // Subclass BatikPrint
101     no usages
102     class BatikPrint extends Produk {
103         4 usages
104         private String bahanKain;
105
106         no usages
107         public BatikPrint(String kode, String nama, double harga, String bahanKain) {
108             super(kode, nama, harga);
109             this.bahanKain = bahanKain;
110         }
111
112         no usages
113         public BatikPrint(String kode, String nama, String bahanKain) {
114             super(kode, nama);
115             this.bahanKain = bahanKain;
116         }
117
118         no usages
119         public String getBahanKain() { return bahanKain; }
120
121         @Override
122         public String toString() {
123             return "Kode: " + getKode() + ", Nama: " + getName() + ", Harga: " + getHarga() + ", Bahan Kain: " + bahanKain;
124         }
125     }

```