Lembar Kerja Peserta Didik  
Modul 5 - OOP Lanjutan

short line

# PETUNJUK PENGERJAAN

1. LKPD ini dikerjakan secara individu namun boleh berdiskusi secara berkelompok.
2. Kelompok menggunakan susunan anggota kelompok yang sudah ditetapkan di awal.
3. Sebelum kalian mengerjakan LKPD ini, pastikan anda sudah bergabung di google classroom
4. LKPD diisi sesuai dengan perintah yang telah dijelaskan pada setiap soalnya.
5. Jika kalian belum memahami instruksi yang diberikan di dalam LKPD, mintalah penjelasan dari bapak/ibu guru.

# POIN TAMBAHAN

1. Tambahkan penanganan kesalahan dan validasi input (contoh: tidak bisa beli stok negatif atau lebih dari yang tersedia).
2. Tambahkan fitur diskon otomatis saat pembelian besar, atau peringatan stok kritis jika stok rendah.
3. Buat antarmuka pengguna sederhana (CLI) untuk memilih tindakan (beli, tambah produk, lihat stok).
4. Gunakan list untuk menyimpan beberapa produk dan tambahkan fitur untuk menambahkan produk baru secara dinamis.
5. Buat menu navigasi, fungsi pencarian produk, dan gunakan struktur data tambahan untuk efisiensi.

# PETUNJUK PENGUMPULAN

1. Tugas dikumpulkan secara **INDIVIDU**.
2. File yang dikumpulkan cukup dokumen ini saja dengan format .pdf.
   1. Cara mendownload dokumen dengan format pdf:
      1. Buka opsi file yang ada di pojok kiri atas layar.
      2. Kemudian pilih opsi download.
      3. Selanjutnya pilih tipe file PDF Documents.
3. Lalu kumpulkan file pada *Google Classroom*.

# SOAL



Anda ditugaskan untuk membuat sebuah program yang mengelola produk di sebuah toko. Buatlah sebuah class bernama **Produk** dengan **contoh atribut-atribut** sebagai berikut:

1. **nama: String** - untuk menyimpan nama produk.
2. **harga: Double** - untuk menyimpan harga produk.
3. **stok: Int** - untuk menyimpan jumlah stok produk yang tersedia.
4. Dan sebagainya, yang biasanya ada pada sebuah produk.. **(Semakin banyak detail maka semakin bagus poinnya)**

Selanjutnya, buat dua fungsi di dalam class **Produk,** yang pertama yakni **untuk menampilkan informasi detail produk,** dan yang kedua **beli produk untuk mengurangi stok produk saat produk dibeli** (jumlah stok yang akan dikurangi sesuai dengan parameter jumlah yang diterima).

Kemudian, buat fungsi di **main()** yang dapat melakukan hal-hal berikut:

1. Membuat sebuah objek dari class **Produk**.
2. Memanggil fungsi untuk menampilkan informasi produk.
3. Melakukan simulasi pembelian produk dan mengurangi stok dengan memanggil fungsi **beliProduk()**.

# LEMBAR JAWABAN

Nama : Kevin Maulana Ibrahim

Nomor Absen : 18

**Studi kasus 1 - Sistem manajemen produk toko**

1. Lampirkan gambar *screenshot* kode kalian dan sertakan notepad yang berisi nama anggota kelompok kalian!

|  |
| --- |

1. Lampirkan kode yang kalian buat.

| class produk(  val nama: String, var harga: Double, var stok: Int, var berat: Double, var merk: String, var deskripsi: String) {    fun informasiproduk() {  println("informasi produk:")  println("----------------------")  println("nama produk: $nama")  println("harga produk: $harga")  println("stok produk: $stok")  println("berat produk: $berat")  println("merk produk: $merk")  println("deskripsi produk: $deskripsi")  }  fun beliproduk(jumlah: Int) {  if (jumlah > stok) {  println("stok tidak cukup untuk melakukan pembelian.")  } else {  stok -= jumlah  println("pembelian berhasil, stok sekarang: $stok")  }  }  }  fun main() {  val produk1 = produk("beras", 300000.0, 50, 25.0, "padimas", "produk beras yang dikenal karena kualitasnya yang baik dan harganya terjangkau")  produk1.informasiproduk()    println("======================")  println("pembelian:")  println("----------------------")  produk1.beliproduk(1)  produk1.beliproduk(51)  }  **HASIL 👍**  informasi produk:  ----------------------  nama produk: beras  harga produk: 300000.0  stok produk: 50  berat produk: 25.0  merk produk: padimas  deskripsi produk: produk beras yang dikenal karena kualitasnya yang baik dan harganya terjangkau  ======================  pembelian:  ----------------------  pembelian berhasil, stok sekarang: 49  stok tidak cukup untuk melakukan pembelian |
| --- |

1. Selanjutnya jelaskan secara singkat cara kerja kode kalian!

| Kode tersebut merupakan implementasi kelas Produk dalam bahasa Kotlin, yang mendefinisikan atribut produk seperti nama, harga, stok, berat, merk, dan deskripsi, serta menyediakan metode untuk menampilkan informasi produk dan melakukan pembelian. Dalam fungsi utama, sebuah instance produk bernama "Beras" dibuat, informasi produk ditampilkan, dan dua percobaan pembelian dilakukan—satu berhasil dan satu gagal karena stok tidak mencukupi. Kode ini berfungsi untuk mengelola informasi dan transaksi produk secara sederhana. |
| --- |