**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 4**

**Object Oriented Programming Java 1**

Logo

Description automatically generated with medium confidence

**Disusun Oleh:**

**Gerald Eberhard**

**(105223002)**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN KOMPUTER**

**UNIVERSITAS PERTAMINA**

**2025**

1. **Pendahuluan**

Studi kasus praktikum ini adalah pembuatan sistem transaksi penjualan sederhana untuk sebuah toko. Sistem ini dirancang untuk mengelola data pelanggan, produk, dan transaksi penjualan menggunakan konsep Pemrograman Berorientasi Objek (PBO) dalam bahasa Java. Setiap transaksi mencakup informasi pelanggan (nama dan ID), produk (nama produk dan kode produk), dan detail transaksi (jumlah pembelian dan total harga). Program ini harus mampu memvalidasi input, memperbarui stok produk setelah transaksi, dan menampilkan laporan transaksi secara terstruktur.

Analisis kebutuhan sistem meliputi:

* Pengelolaan entitas yang saling berhubungan (pelanggan, produk, transaksi).
* Penggunaan kelas dan objek untuk merepresentasikan entitas tersebut.
* Penerapan prinsip PBO yang telah dipelajari seperti *encapsulation*, *inheritance*, dan *polymorphism*.
* Validasi input untuk memastikan data yang dimasukkan pengguna akurat.

Metode yang dipilih meliputi:

* **Kelas dan Objek**: Untuk memodelkan entitas nyata (produk, pelanggan, transaksi).
* **Encapsulation**: Menggunakan akses modifier (private, public) untuk melindungi data dan menyediakan akses melalui getter/setter.
* **Array atau Koleksi**: Untuk menyimpan daftar produk atau transaksi.
* **Struktur Kontrol dan Perulangan**: Untuk mengelola input dan proses transaksi.

Pemilihan metode ini dilandaskan berdasarkan materi yang telah dipelajari selama praktikum. Kemudian karena tidak ada detail spesifik tentang harga produk atau aturan bisnis lainnya dalam modul, saya membuat asumsi sendiri untuk nilai harga produk dan jumlah stok.

1. **Variabel**

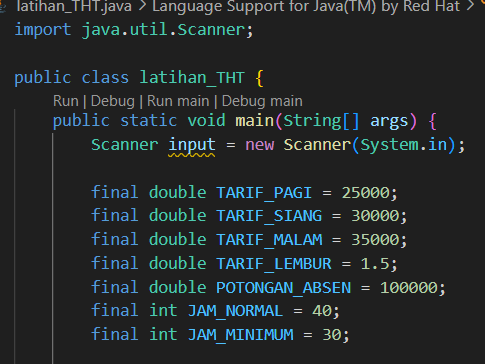
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Variabel** | **Tipe Data** | **Fungsi** |
| 1 | idKaryawan | String | Menyimpan array ID karyawan untuk identifikasi unik. |
| 2 | namaKaryawan | String | Menyimpan array nama karyawan untuk informasi identitas. |
| 3 | shiftKaryawan | Int | Menyimpan array shift kerja (1=Pagi, 2=Siang, 3=Malam) untuk menentukan tarif. |
| 4 | jamKerja | Int | Menyimpan array total jam kerja per minggu untuk perhitungan gaji. |
| 5 | hariAbsen | Int | Menyimpan array jumlah hari absen untuk menghitung potongan gaji. |
| 6 | gajiTotal | double | Menyimpan array total gaji setelah perhitungan tarif, lembur, dan potongan. |
| 7 | TARIF\_PAGI | Final Double | Konstanta untuk tarif per jam shift pagi (Rp 25.000). |
| 8 | TARIF\_SIANG | Final Double | Konstanta untuk tarif per jam shift siang (Rp 30.000). |
| 9 | TARIF\_MALAM | Final Double | Konstanta untuk tarif per jam shift malam (Rp 35.000) |
| 10 | TARIF\_LEMBUR | Final Double | Konstanta untuk pengali tarif lembur (1.5 kali tarif normal). |
| 11 | POTONGAN\_ABSEN | Final Double | Konstanta untuk potongan gaji per hari absen (Rp 100.000). |
| 12 | JAM\_NORMAL | Final Double | Konstanta untuk batas jam kerja normal (40 jam). |
| 13 | JAM\_MINIMUM | Final Double | Konstanta untuk batas minimum jam kerja (30 jam). |
| 14 | input | Scanner | Objek untuk membaca input dari pengguna. |
| 15 | jumlahKaryawan | Int | Variabel untuk menyimpan jumlah karyawan yang akan diinput. |
| 16 | i | Int | Variabel indeks untuk perulangan pengolahan data karyawan. |
| 17 | tarifPerJam | Double | Variabel sementara untuk menyimpan tarif per jam berdasarkan shift. |
| 18 | gajiNormal | Double | Variabel untuk menyimpan gaji dari jam kerja normal. |
| 19 | gajiLembur | Double | Variabel untuk menyimpan gaji dari jam lembur. |
| 20 | totalGaji | Double | Variabel sementara untuk menghitung total gaji sebelum disimpan. |
| 21. | shiftText | String | Variabel lokal tempat konversi nilai shift untuk kemudian akan di print |

1. **Constructor dan Method**

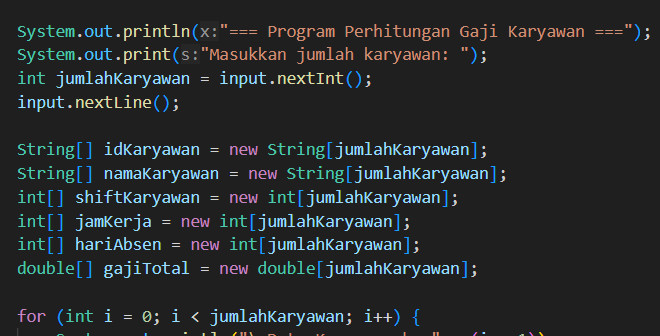
Program ini itu merupakan versi yang sangat pemula jadi tidak ada konsep PBO yang diterapkan di dalam codingan ini sehingga hanya ada satu file dan satu method yaitu:

|  |  |  |
| --- | --- | --- |
| **Nama Method** | **Jenis Method** | **Fungsi** |
| main(String[] args) | Public Static | Fungsi utama yang menjalankan program, mengelola input data karyawan, perhitungan gaji, dan menampilkan laporan. |

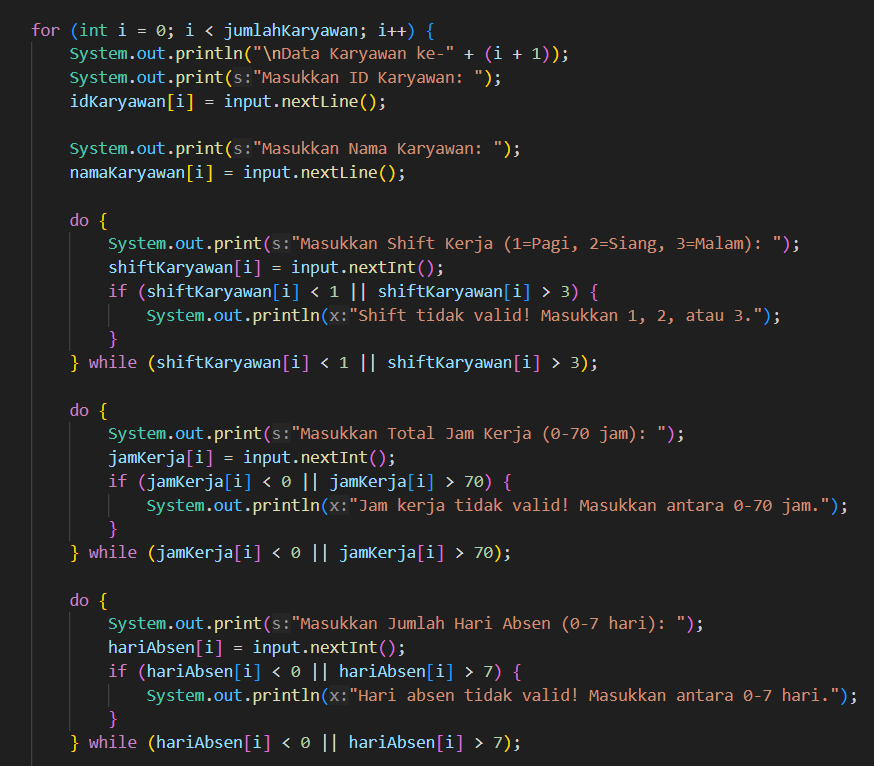
1. **Dokumentasi dan Pembahasan Code**

****

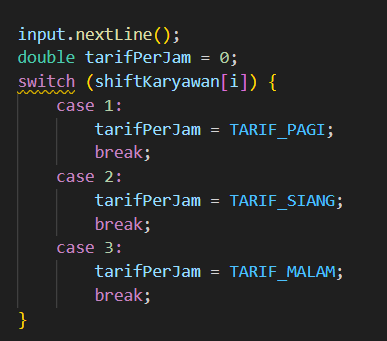
* Bagian Pertama ada import java.util.Scanner untuk menggunakan library scanner sehingga bisa meminta input dari user. Kemudian ada deklarasi final atau kalo di c++ itu biasa disebut variabel konstan yang artinya tidak bisa diubah-ubah nilainya oleh operasi aritmatika apapun. Karena ini menggunakan java maka syntaksnya berubah menggunakan final diikuti dengan tipe data dan nama variabel serta nilai dari variabel tersebut.



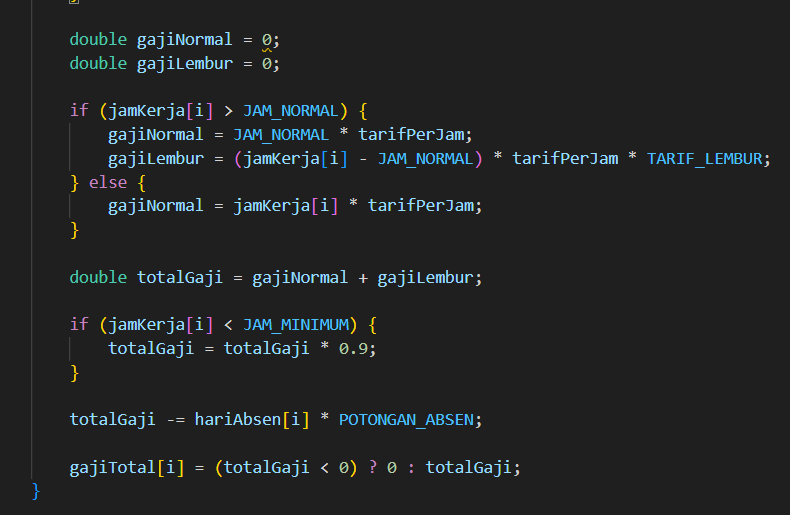
* Berikutnya setelah mendeklarasikan variabel-variabel konstannya program akan berlanjut dengan meminta inputan jumlah karyawan yang ingin di data. Dan kemudian membuat array sebanyak jumlah karyawan untuk menyimpan ide, nama, shift, jam, berapa hari absen, dan gaji total.
* Kemudian, setelah ini program akan berlanjut ke dalam looping yang sangat panjang dimana hampir semua proses penghitungan akan berlangsung di dalam looping ini sehingga yang tersisa untuk dilakukan pada saat selesai melakukan looping adalah untuk melakukan print hasil dengan rapih



* Bagian berikut adalah melakukan looping untuk meminta inputan file oleh user. Pada kasus ini saya menggunakan do-while untuk memastikan inputan user telah sesuai. Mengapa do-while karena saya ingin user memasukkan inputannya terlebih dahulu kemudian akan melakukan pengecekan. Oleh karena itu do-while akan sangat efektif.



* Berikutnya ada switch-case untuk memberikan nilai tarif per jam untuk menghitung gaji karyawan. Berdasarkan pilihan shift 1, 2, atau 3 yang telah diminta sebelumnya kepada user.



* Pada akhir looping akan dilakukan perhitungan lebih lanjut untuk mencari gaji total [i]. Pada awalnya akan digunakan variabel sementar yaitu total gaji. Total gaji sendiri di dapat dari perhitungan gaji lembur yang kemudian dijumlahkan dengan gaji normal.

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

* Codingan pun diakhiri dengan menggunakan syntaks printf ” %-10s %-20s %-10s %-15s %-10s %-15s\n” untuk mengatur kerapihan dari tabel yang ingin di print. Printf bertujuan untuk mengeluarkan output dalam format tertentu.

Arti dari masing-masing simbol pada syntaks yang digunakan untuk merapihkan tabel tersebut adalah sebagai berikut:

* S= itu untuk string, d itu untuk integer, dan f itu untuk float.
* % menandakan awal mula formating
* - menunjukkan agar teks dibuat rata kiri
* 10 itu menandakan lebar karakter
* Khusus apabila tipe datanya float kita bisa menambahkan .2f dibelakangnya untuk menyingkronkan ketelitian berapa angka di belakang komanya. Jika mau 2 angka dibelakang koma cukup gunakan .2f
* Kemudian ada juga looping dengan for loop untuk mengubah nilai dari shiftkaryawan yang tadinya berupa angka menjadi huruf di translate sesuai dengan huruf yang diinputkan oleh user. Kemudian pada akhirnya setelah diterjemahkan maka shiftText akan di print beserta dengan id karyawan, nama karyawan, hari absen, jam kerja, dan total gaji.

1. **Kesimpulan**

Praktikum ini berhasil mengimplementasikan program perhitungan gaji karyawan menggunakan konsep pemrograman Java dasar seperti struktur kontrol (if-else, switch-case), perulangan (for, do-while), array, dan konstanta (final). Program memenuhi kebutuhan studi kasus dengan menghitung gaji berdasarkan shift, lembur, dan potongan absen, serta menampilkan laporan yang terstruktur. Penggunaan validasi input meningkatkan keandalan, meskipun masih bisa ditingkatkan dengan penanganan error lebih lanjut. Pengalaman ini memperkuat pemahaman tentang penerapan logika pemrograman dalam menyelesaikan masalah praktis. Sekian dan Terimakasih

1. **Daftar Pustaka**

Modul 3: Pemograman Jawa Dasar II