**JOBSHEET 7 DASAR PEMROGRAMAN**

Nama : Jonathan Abdiel Haryono

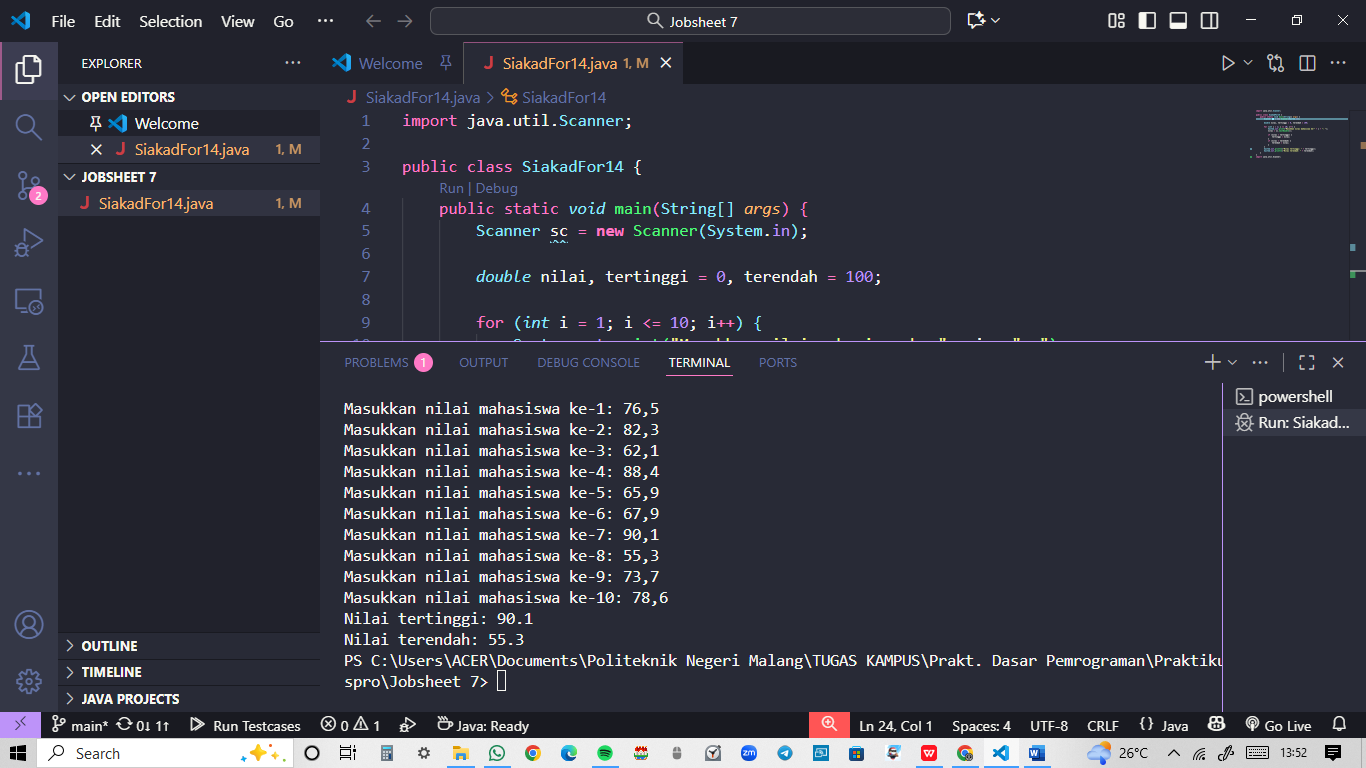
NIM : 254107020036

Absen : 14

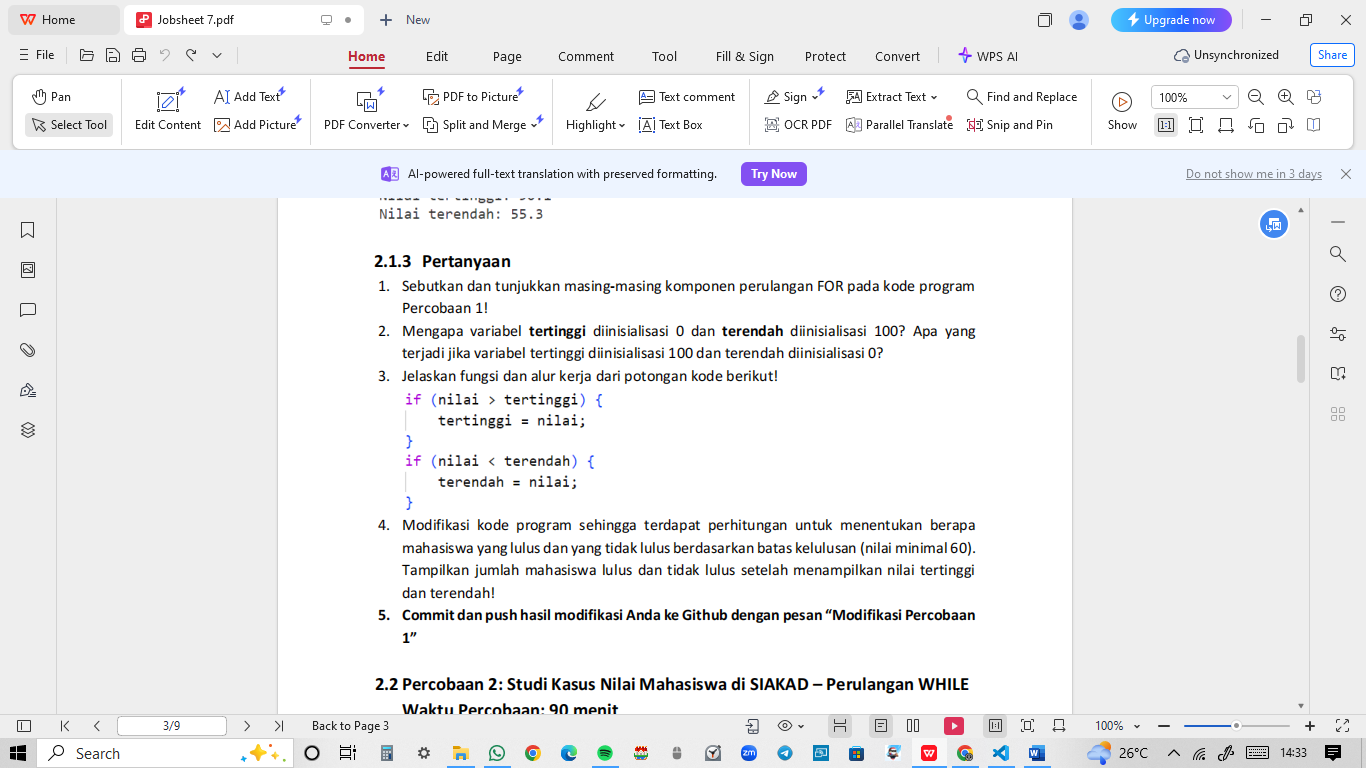
Kelas : 1G

**Percobaan 1**

****

****

**Pertanyaan 1**



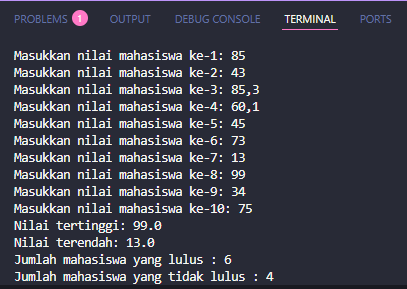
**Jawaban 1**

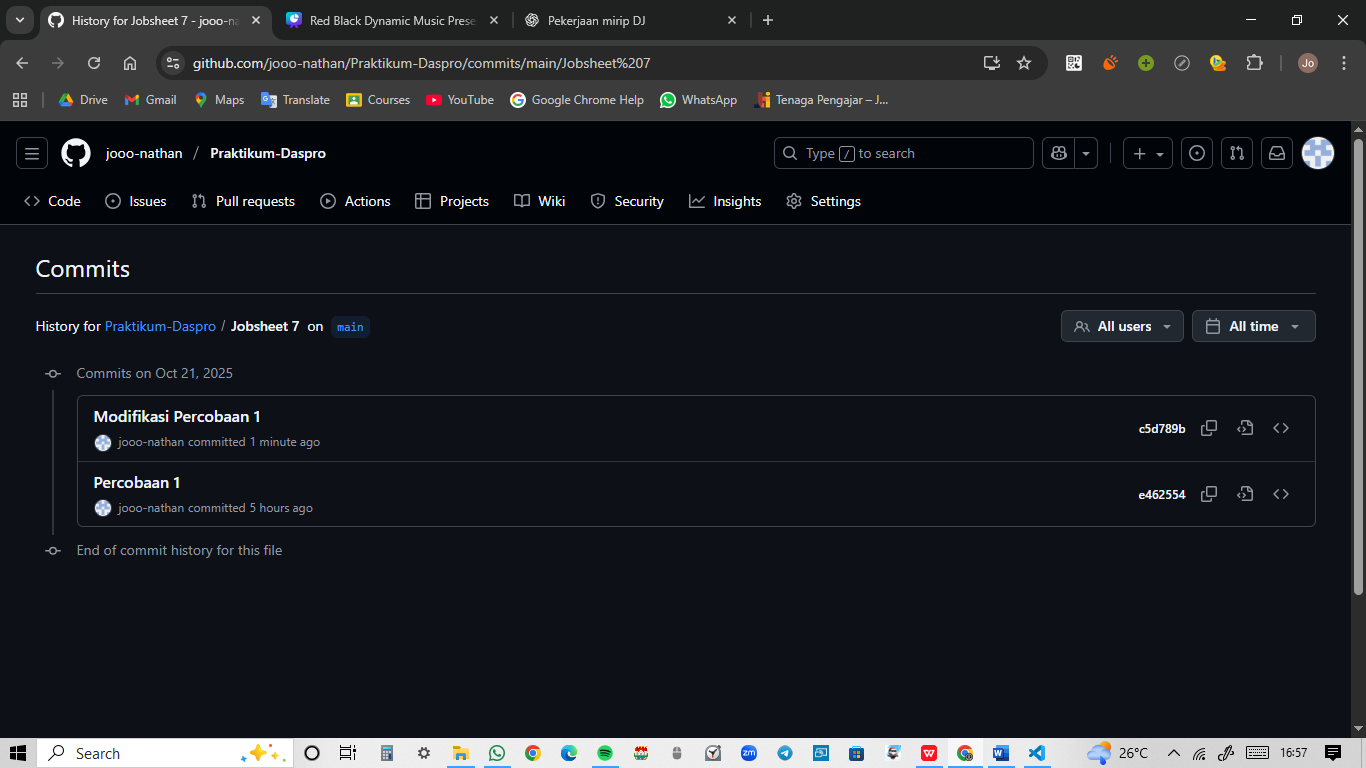


Komponen FOR

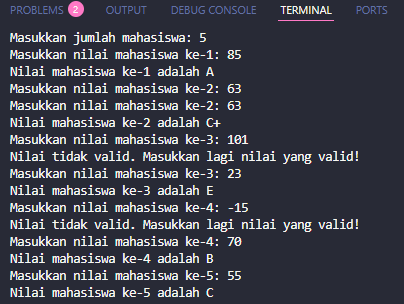
* int i = 1 🡪 Inisialisasi
* i <= 10 🡪 Kondisi
* i++ 🡪 Update
* { … } 🡪 Statement (yang di dalam kurung kurawal)

1. Variabel **tertinggi** diinisialisasi 0 dan variabel **terendah** diinisialisasi 100 karena kedua variabel tersebut akan digunakan sebagai pembanding dengan masukan sehingga dibutuhkan kemungkinan terburuknya yakni ketika nilai tertingginya 0 dan nilai terendahnya 100 supaya ketika dibandingkan, masukan akan dicek apakah ia lebih dari nilai tertinggi (0), kemudian apakah kurang dari nilai terendah (100). Demikian seterusnya hingga ditemukan nilai yang paling tepat. Apabila variabel **tertinggi** diinisialisasi 100 dan variabel **terendah** diinisialisasi 0, maka berapapun masukan yang diterima, program akan selalu mengeluarkan 0 untuk nilai terendah dan 100 untuk nilai tertinggi karena ketika kedua nilai tersebut dibandingkan dengan masukan, tak akan ditemukan kondisi dimana masukan lebih dari 100 atau kurang dari 0 (kecuali masukan tidak sesuai, misalnya -1, 101).
2. Fungsi potongan kode tersebut adalah untuk mengecek apakah variabel **nilai** merupakan yang terendah atau tertinggi. Alurnya adalah sebagai berikut. Pertama, program akan mengecek apakah nilai variabel **nilai** lebih dari nilai variabel **tertinggi** saat ini (mula-mula 0). Jika ya, maka variabel **tertinggi** akan diisi dengan nilai dari variabel **nilai**. Kemudian, program akan mengecek apakah nilai variabel **nilai** kurang dari nilai variabel **terendah** saat ini (mula-mula 100). Jika ya, maka variabel **terendah** akan diisi dengan nilai dari variabel **nilai**. Begitu seterusnya untuk setiap masukan yang diterima sehingga akan ditemukan nilai terendah dan tertinggi dari suatu kumpulan bilangan yang dimasukkan.

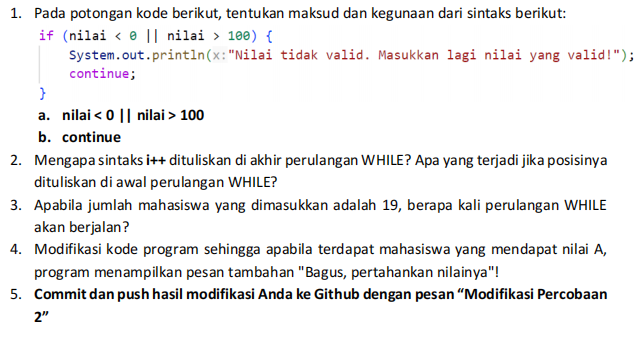


2. 

**Percobaan 2**



**Pertanyaan 2**

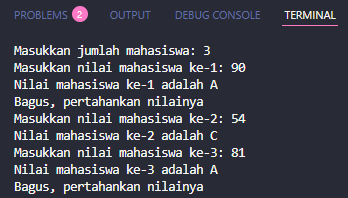


**Jawaban 2**

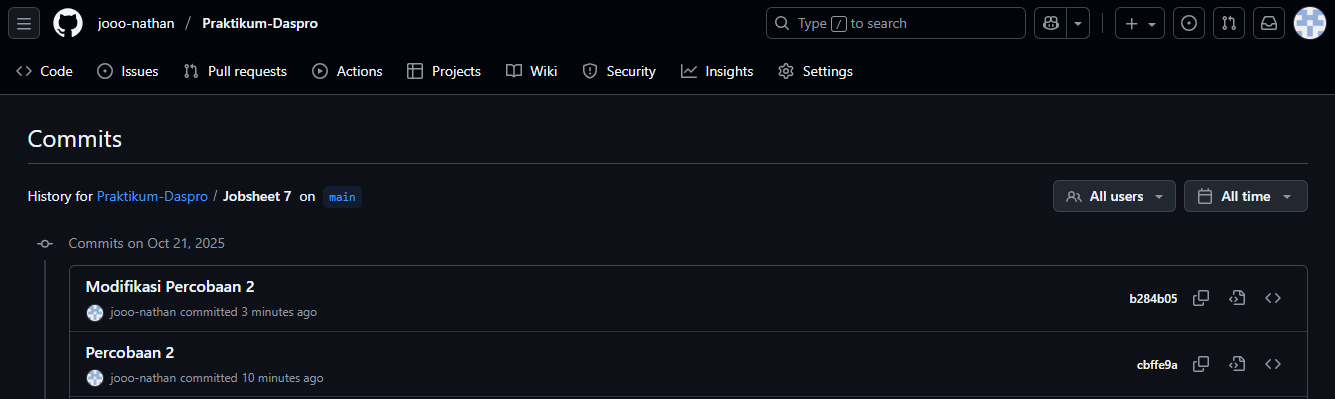
1. Sintaks **nilai < 0 || nilai > 100** maksudnya adalah kondisi tersebut bernilai **true** jika variabel **nilai** bernilai kurang dari 0 atau variabel **nilai** bernilai lebih dari 100, yang mana artinya akan terpenuhi (bernilai **true**)jika salah satu atau kedua kondisi bernilai **true**. Kegunaannya adalah untuk menentukan apakah variabel **nilai** sebagai nilai mahasiswa ternyata valid atau tidak (valid jika berada dalam rentang 0 hingga 100).
2. Sintaks **continue** maksudnya adalah melewati 1 iterasi saat ini dengan tidak melakukan seluruh arahan tepat setelah sintaks **continue** yang ada dalam perulangan melainkan langsung melanjutkan ke iterasi selanjutnya. Kegunaannya adalah untuk melewati suatu iterasi tertentu sesuai dengan kondisi yang diberikan, misalnya pada potongan kode tersebut adalah untuk melewati iterasi jika nilainya tidak valid.
3. Sintaks **i++** dituliskan di akhir perulangan WHILE karena sebelum melakukan *update* (melalui sintaks **i++**)*,* program perlu melakukan pengecekan terlebih dahulu apakah nilai yang dimasukkan sudah valid atau belum. Jika sudah, program akan berjalan sebagaimana mestinya, tanpa peringatan bahwa nilai tidak valid, dan melakukan *update* di akhir program. Namun jika belum, program akan mengeluarkan peringatan bahwa nilai tidak valid, tidak akan melakukan *update*, dan mengulangi iterasi pada i saat itu hingga nilai yang dimasukkan valid.

Jika posisinya dituliskan di awal perulangan WHILE, ada 2 hal yang akan terjadi. Pertama, keluaran untuk **(i + 1)** akan dimulai dari 2, bukan 1 sehingga menyebabkan kebingungan oleh pengguna (kecuali jika sintaks **(i + 1)** juga disesuaikan dengan menghilangkan **+1**). Kedua, apabila nilai yang dimasukkan tidak valid, *update* akan terus berjalan tanpa mempedulikan apakah nilai tersebut valid atau tidak sehingga tidak ada iterasi tambahan untuk nilai yang tidak valid (jika nilai yang dimasukkan valid semua, tidak akan ada perbedaan antara **i++** di awal atau akhir). Intinya, ada error, namun bila sintaks **(i + 1)** diubah menjadi **(i)** dan masukan semuanya valid, tidak akan ditemukan error.

1. Dengan jumlah mahasiswa yang dimasukkan adalah sebanyak 19, apabila nilai yang dimasukkan valid semua, maka perulangan WHILE akan berjalan sebanyak 19 kali pula. Namun, apabila nilai yang dimasukkan ada yang tidak valid, maka perulangan WHILE akan berjalan sebanyak 19 + banyaknya nilai yang tidak valid.

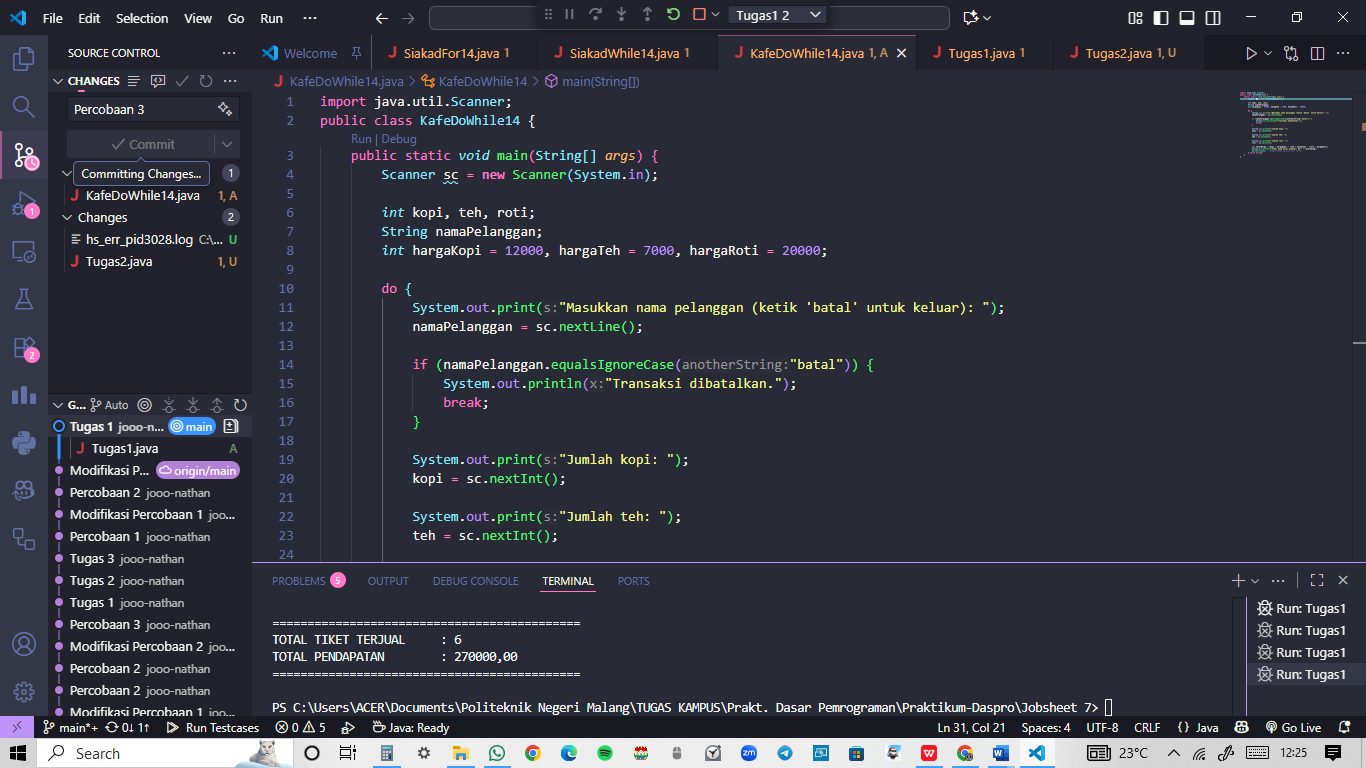
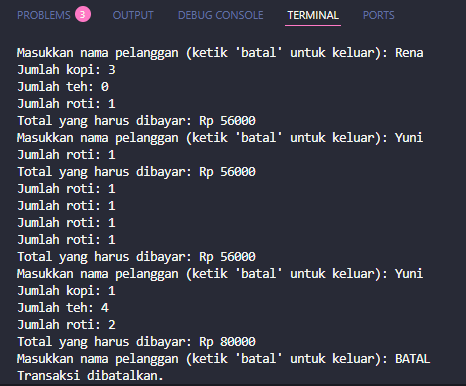




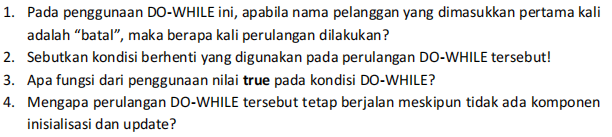


**Percobaan 3**

****

****

**Pertanyaan 3**



**Jawaban 3**

1. Dengan memasukkan “batal” sebagai nama pelanggan, perulangan terhitung sudah dilakukan sebanyak 1x karena perulangan DO-WHILE akan melakukan perintah terlebih dahulu baru melakukan pengecekan yang artinya apapun kondisinya nanti, dengan melakukan perulangan DO-WHILE, program terhitung sudah melakukan 1x perulangan.
2. Kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut adalah ketika pengguna memasukkan kata “batal” pada masukan nama pangguna (termasuk jenis menambahkan pertanyaan karena perulangan berdasarkan jawaban pengguna).
3. Nilai **true** digunakan untuk mengulang program dengan terus-menerus tanpa berhenti (kecuali dihentikan oleh **break**) karena nilai **true** akan tetap selamanya **true**.
4. Karena komponen inisialisasi dan *update* memang sudah tidak diperlukan untuk kasus ini, karena kasus ini ingin agar pengguna bisa menjalankan program terus menerus selama nama yang dimasukkan bukan kata “batal”. Namun, sebenarnya, komponen-komponen ini masih ada, hanya saja fungsinya sudah digantikan oleh hal yang lain, yakni oleh aturan bahwa **true** akan selamanya **true**.

**TUGAS**

1. Seorang pengelola bioskop ingin membuat program untuk menghitung total penjualan tiket dalam satu hari. Tiket dijual dengan harga Rp 50.000 per tiket. Program harus menghitung total tiket yang terjual dan total harga penjualan tiket selama satu hari dengan ketentuan sebagai berikut:

* Jika pelanggan membeli lebih dari 4 tiket, pelanggan mendapatkan diskon 10%.
* Jika pelanggan membeli lebih dari 10 tiket, pelanggan mendapatkan diskon 15%.
* Jika input jumlah tiket tidak valid (negatif), program akan mengabaikan input

tersebut dan meminta input ulang.

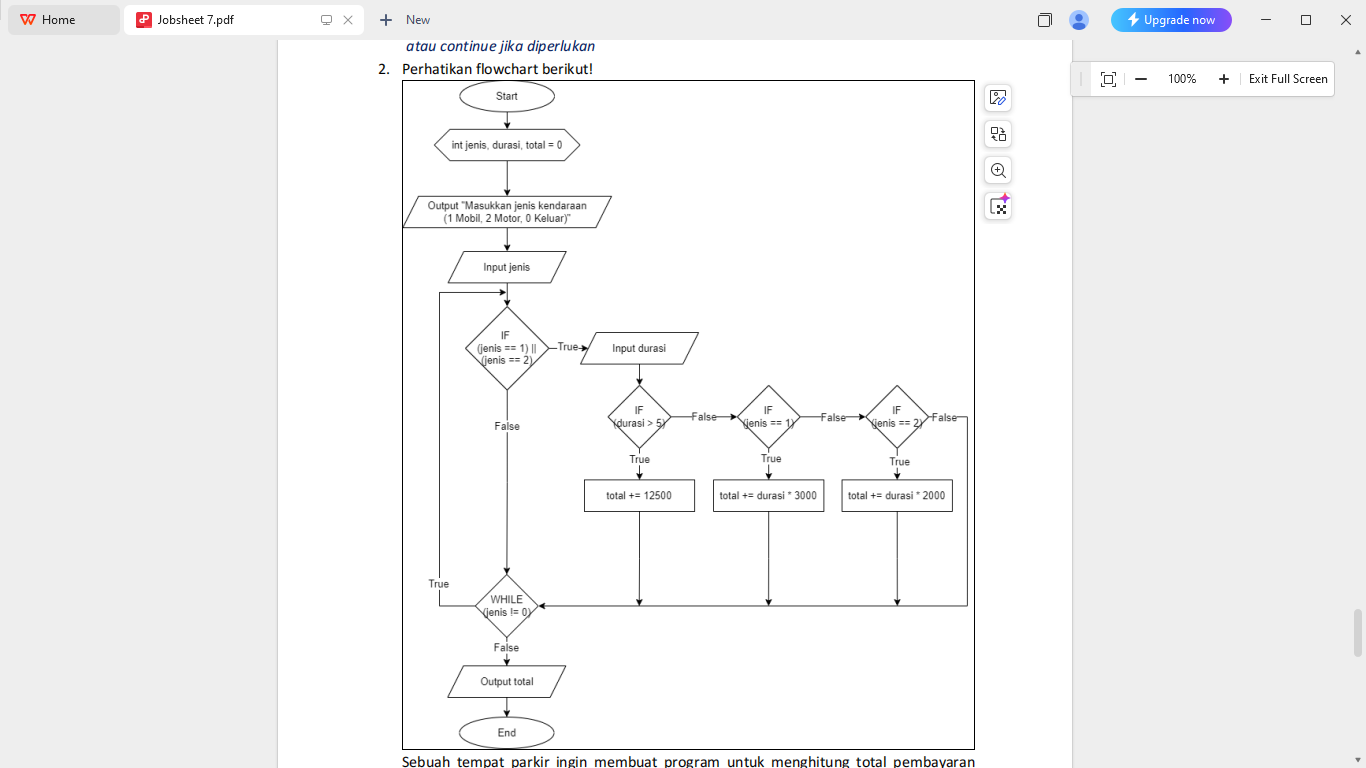
**Commit dan push program Anda ke Github dengan pesan “Tugas 1”**

*Catatan: Perulangan dapat menggunakan for, while, atau do-while. Penambahan break atau continue jika diperlukan*

**Jawaban**

****

1. Perhatikan flowchart berikut!



Sebuah tempat parkir ingin membuat program untuk menghitung total pembayaran parkir dari beberapa kendaraan. Tarif parkir adalah Rp 3.000 per jam untuk mobil dan Rp 2.000 per jam untuk motor. Namun, jika durasi parkir lebih dari 5 jam, diberikan tarif tetap sebesar Rp 12.500 untuk semua kendaraan. Program akan terus meminta masukan selama input bukan 0. Implementasikan flowchart tersebut ke dalam bentuk kode program Java! **Commit dan push program Anda ke Github dengan pesan “Tugas 2”.**

**Jawaban :**

