**LAPORAN PRAKTIKUM**

**TEKNIK PEMROGRAMAN**

**CLEAN CODE SONARQUBE TEST**

MINGGU KE-10



NAMA: FAUZI ISMAIL

NIM: 241524042

KELAS: D4-1B

**PROGRAM STUDI SARJANA TERAPAN**

**TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2025**

## DAFTAR ISI

[DAFTAR ISI 2](#_Toc197386877)

[https://github.com/mailvlous/teknikPemrograman/tree/main/Week9/flight-management-app 3](#_Toc197386878)

[FLIGHT MANAGEMENT APP CLEAN CODE TEST ANALYSIS USING SONARQUBE 3](#_Toc197386879)

[1. SEBELUM DI REFACTOR 3](#_Toc197386880)

[2. REFACTORING 7](#_Toc197386881)

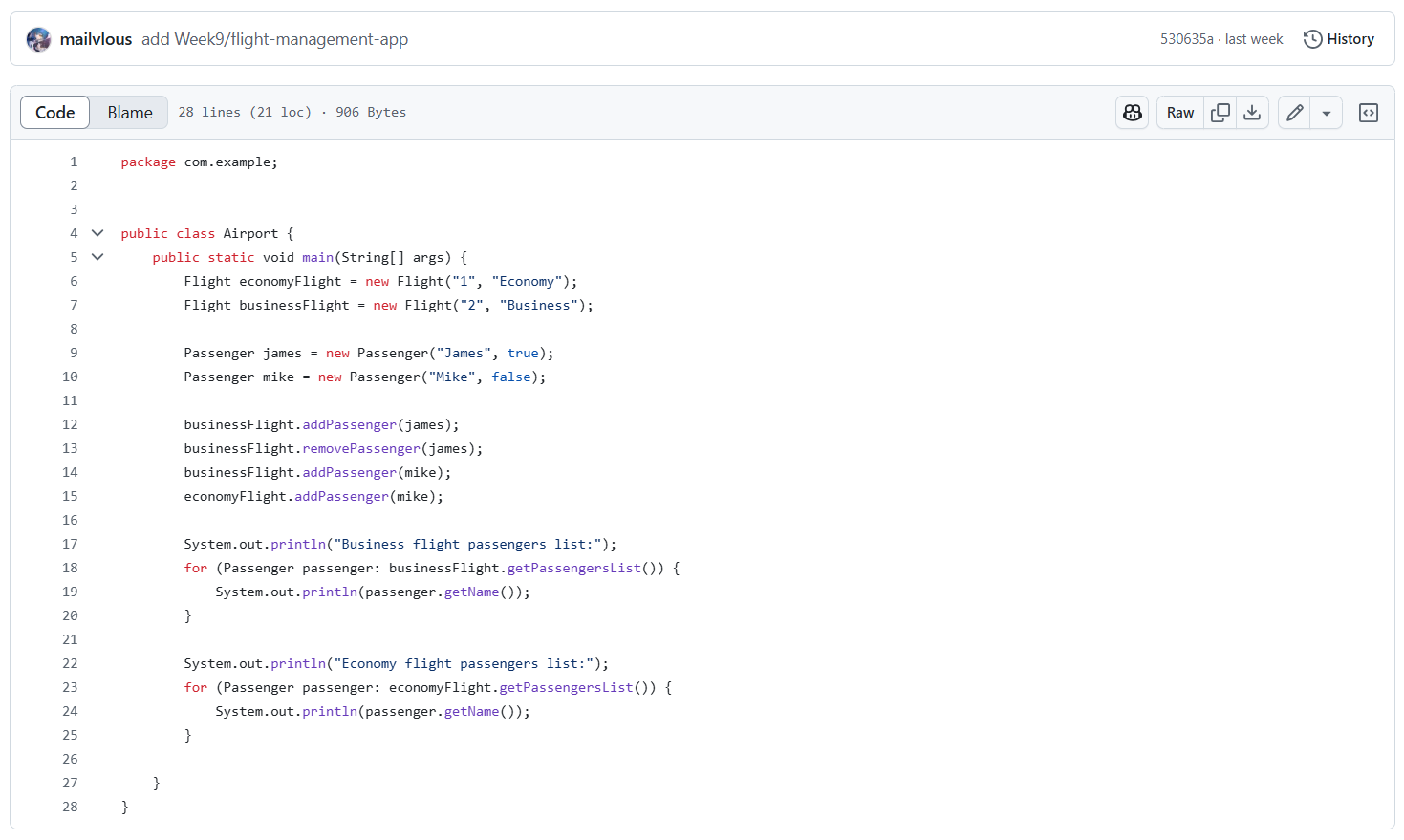
[3. SETELAH REFACTOR 8](#_Toc197386882)

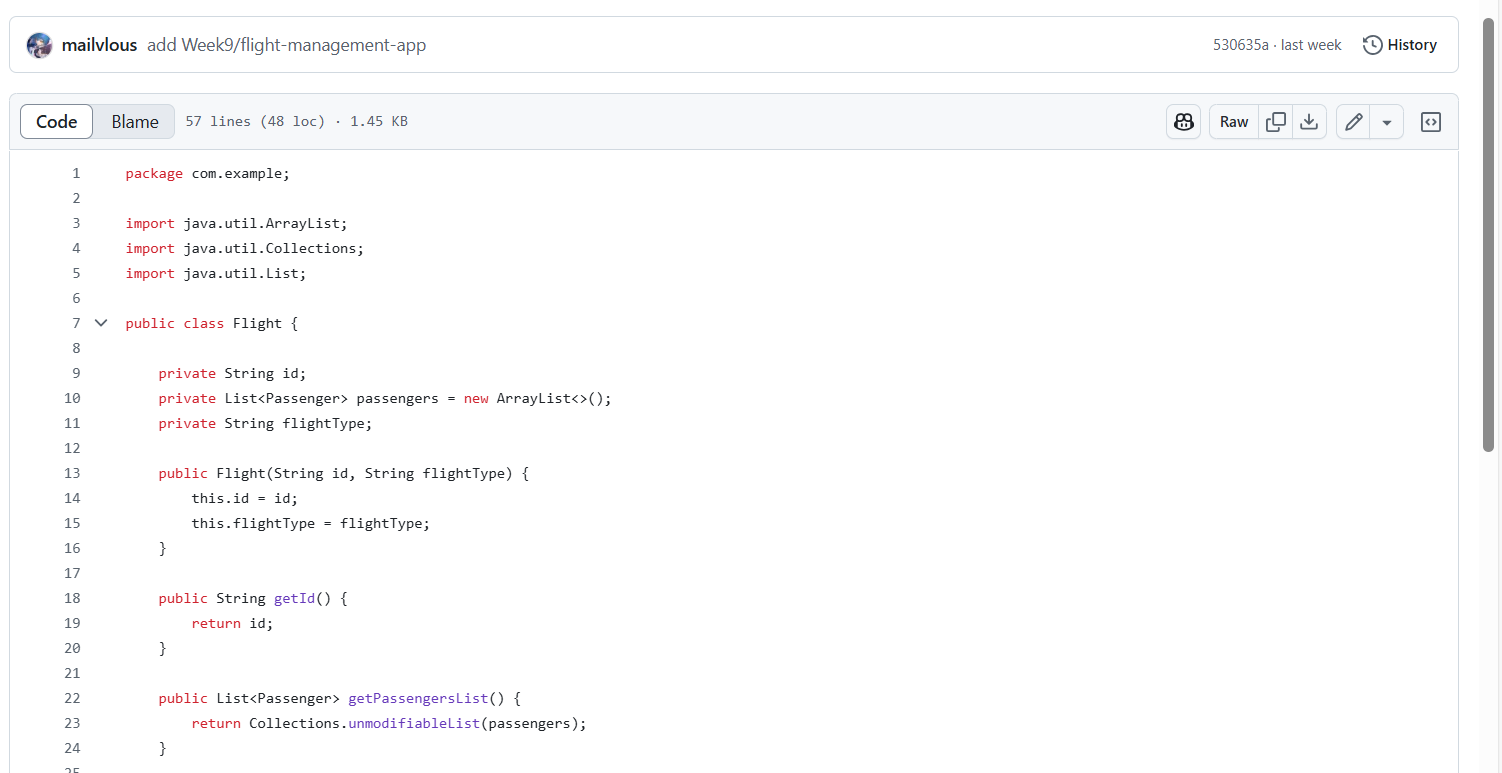
## <https://github.com/mailvlous/teknikPemrograman/tree/main/Week9/flight-management-app>

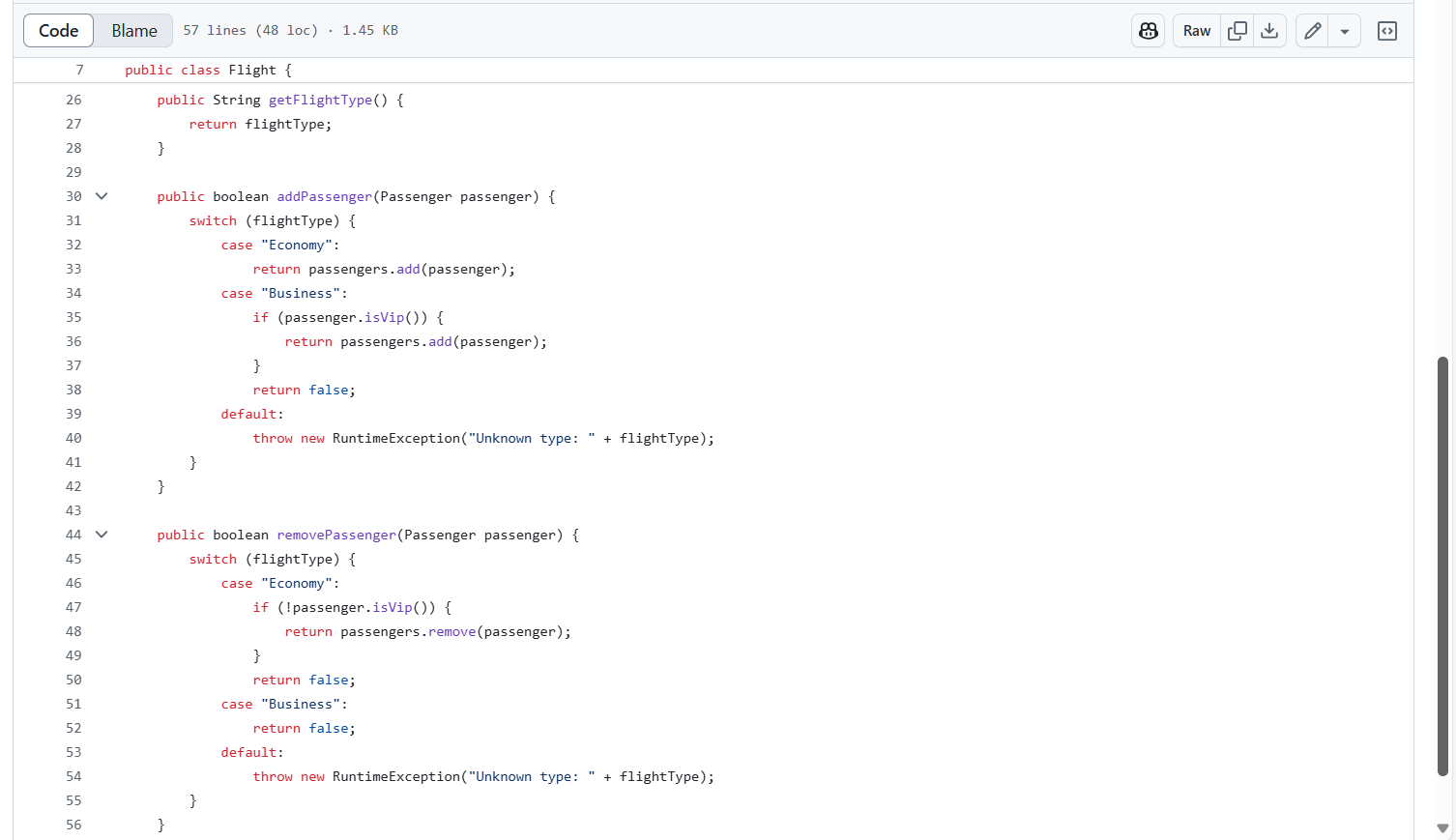
## FLIGHT MANAGEMENT APP CLEAN CODE TEST ANALYSIS USING SONARQUBE

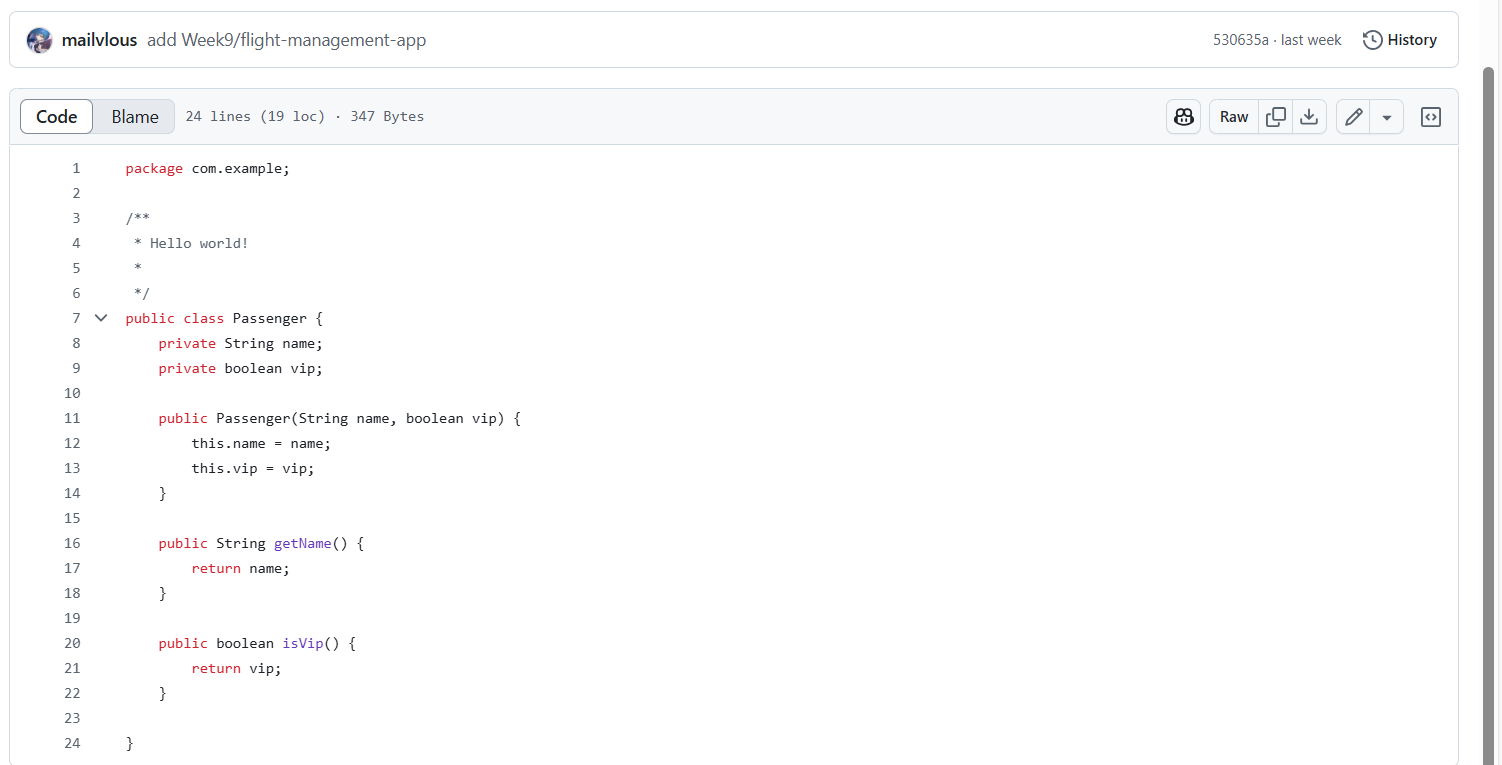
### 1. SEBELUM DI REFACTOR

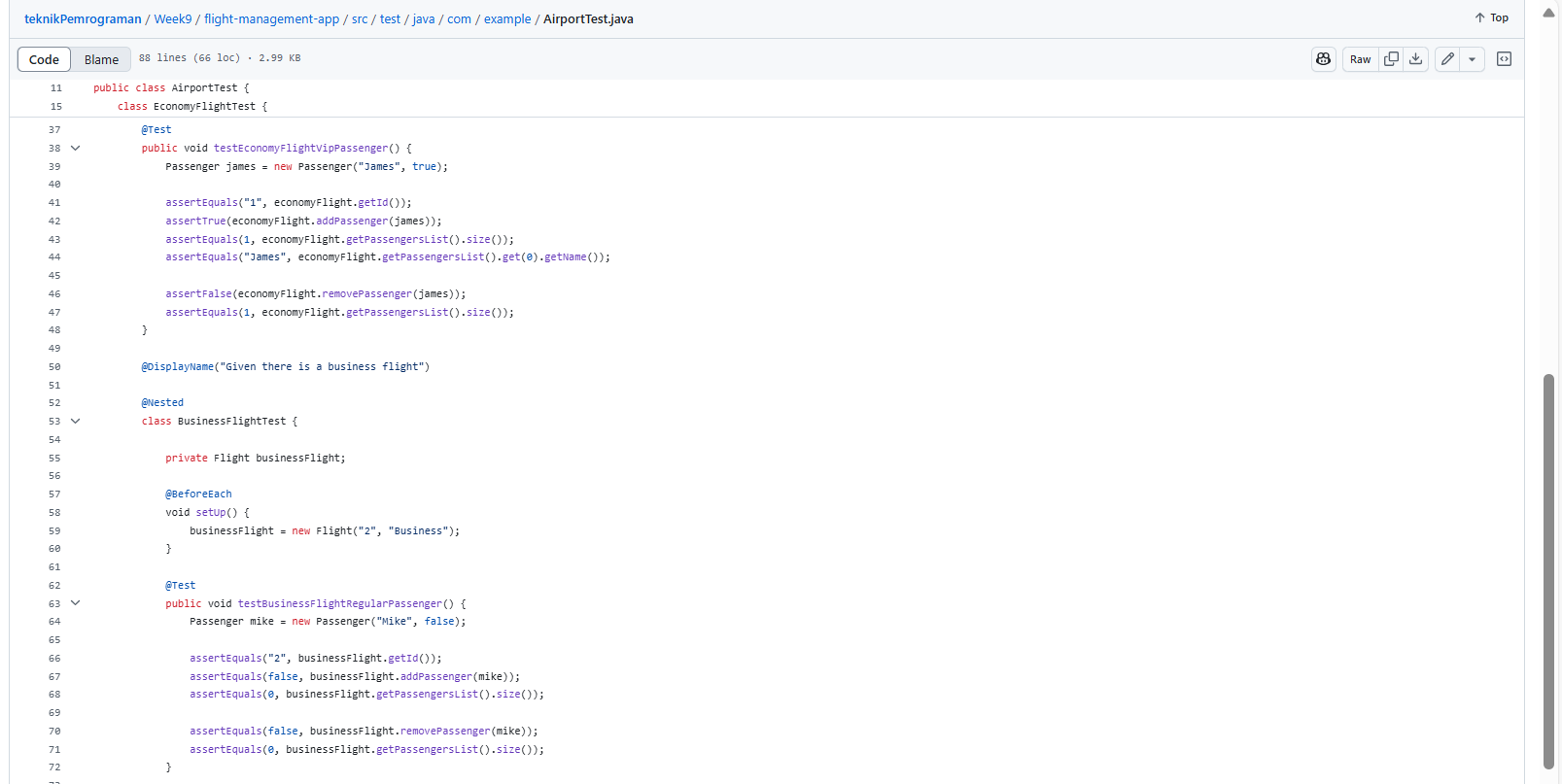
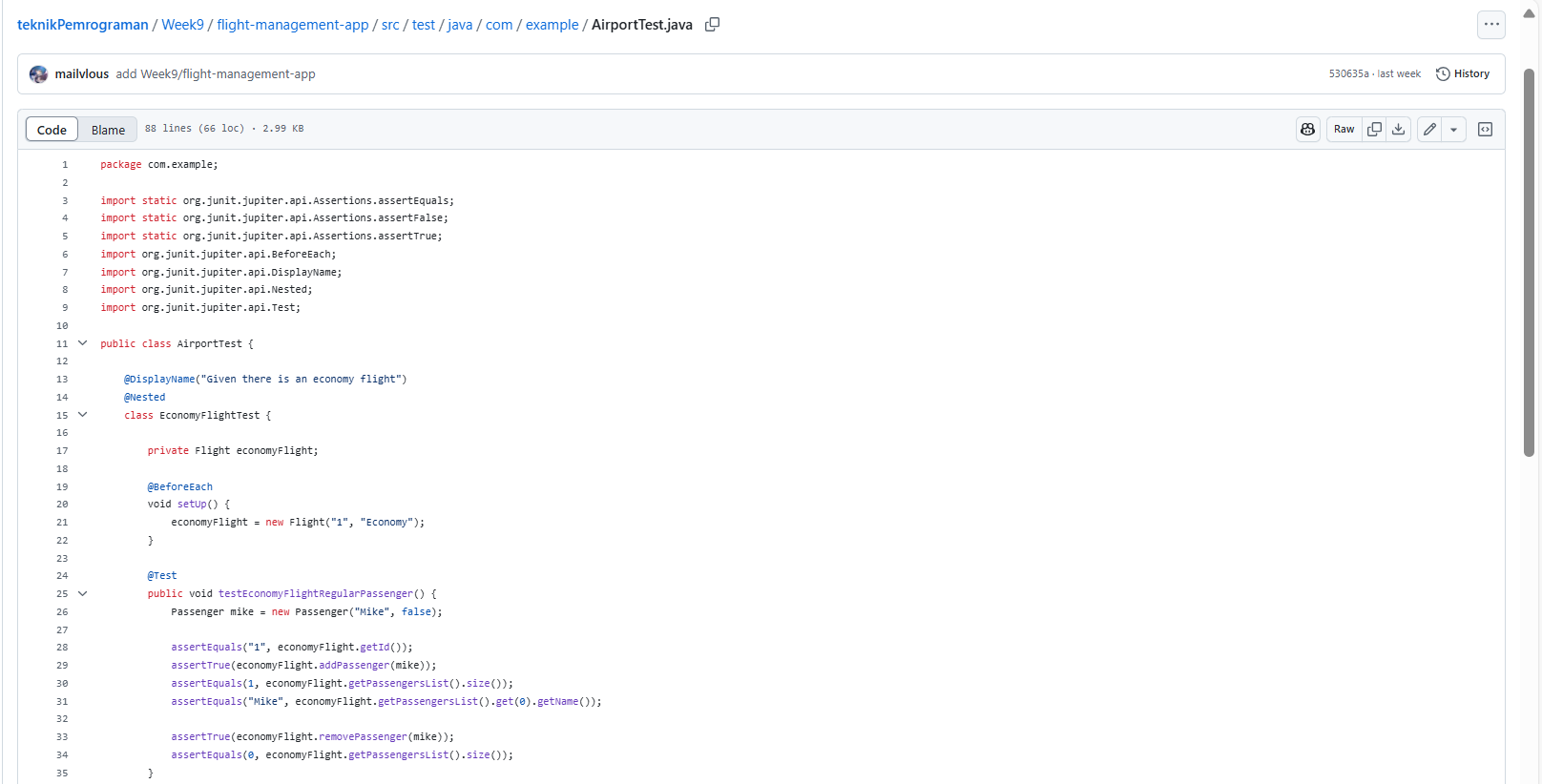
Airport.java



Flight.java



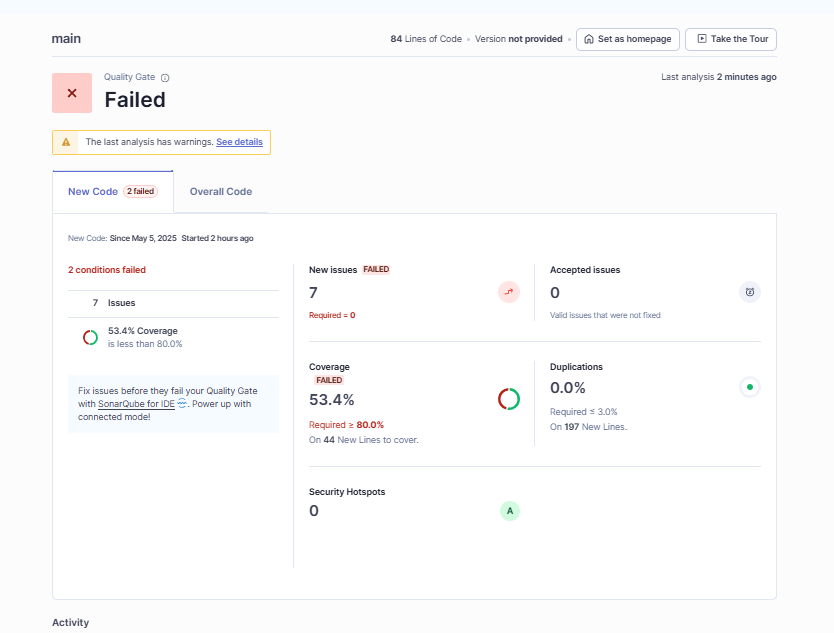
Passenger.java

AirportTest.java



### 2. REFACTORING

SEBELUM DI REFACTOR DI SONARQUBE ANALYSIS



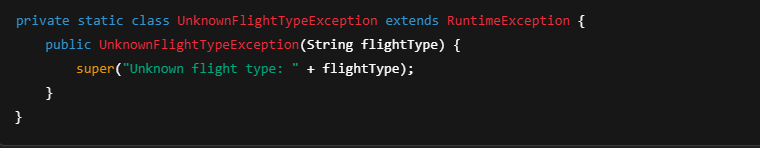
Yang saya lakukan adalah:

1. Mengganti System.out.println() dengan Logger

Menggunakan: 

Sehingga: 

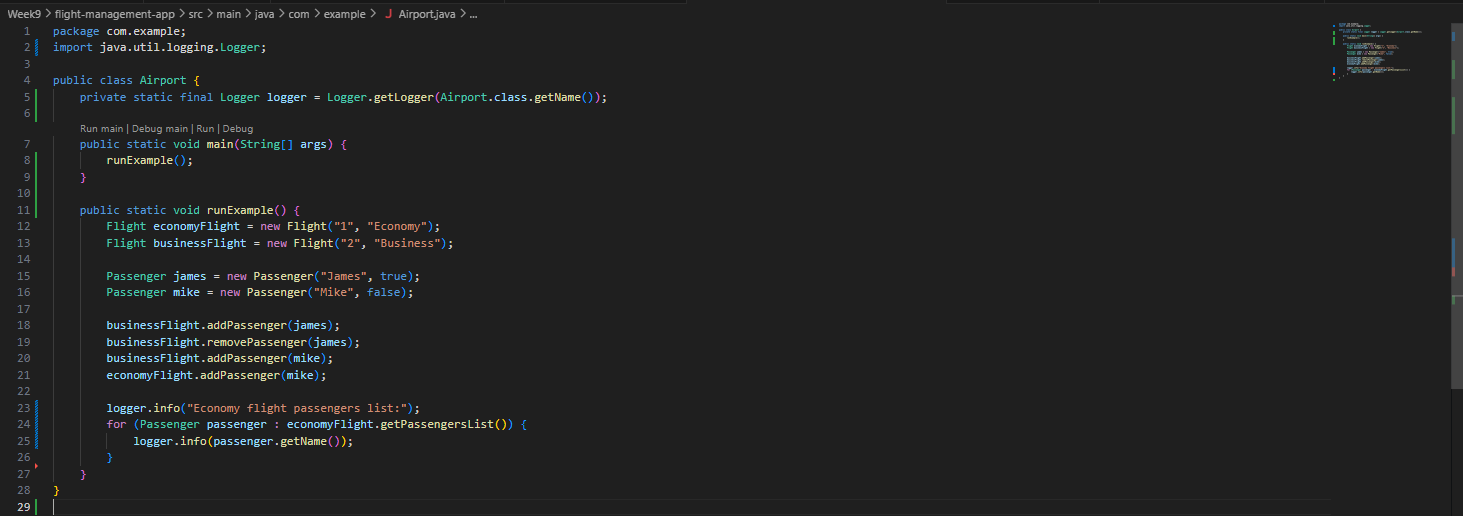
2. Menghindari RuntimeException generic

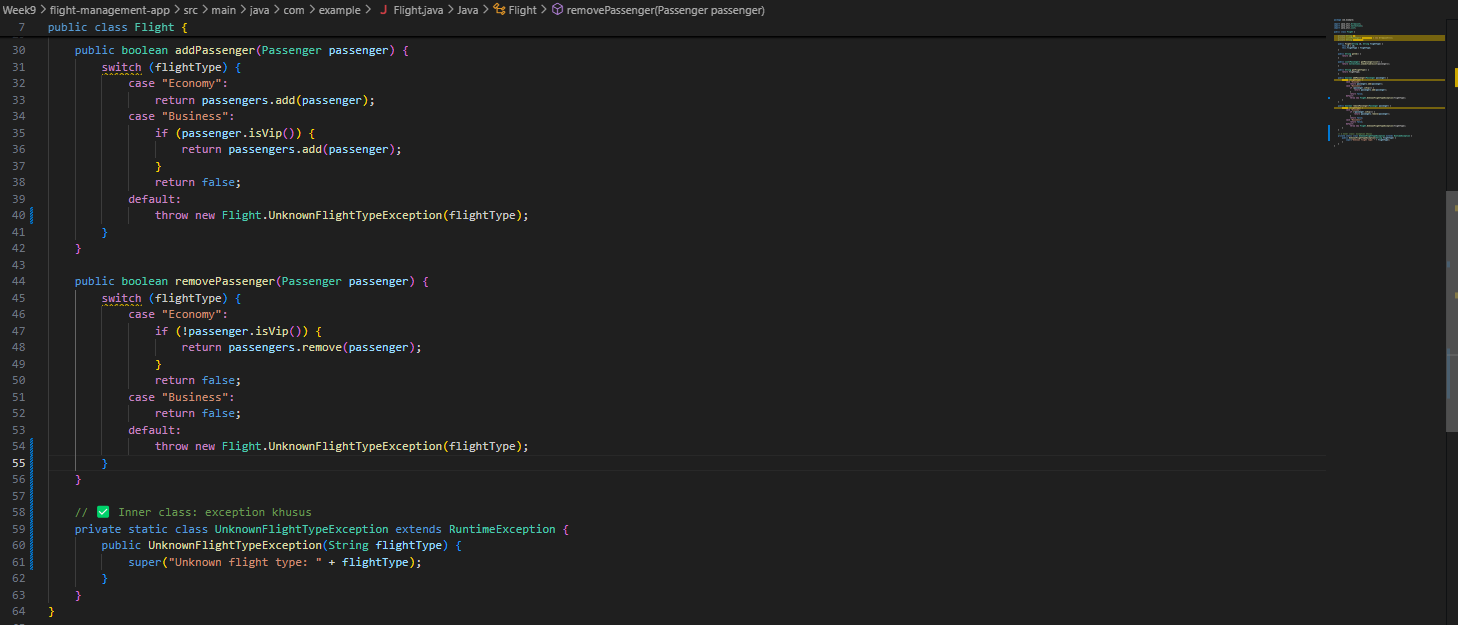
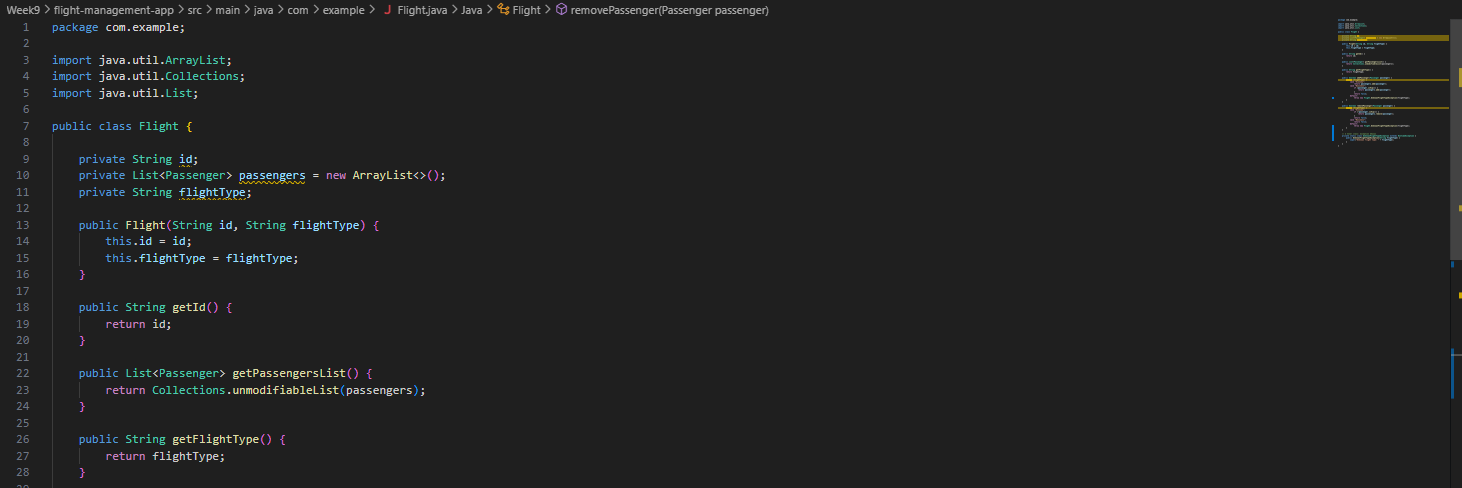
Menggunakan: 

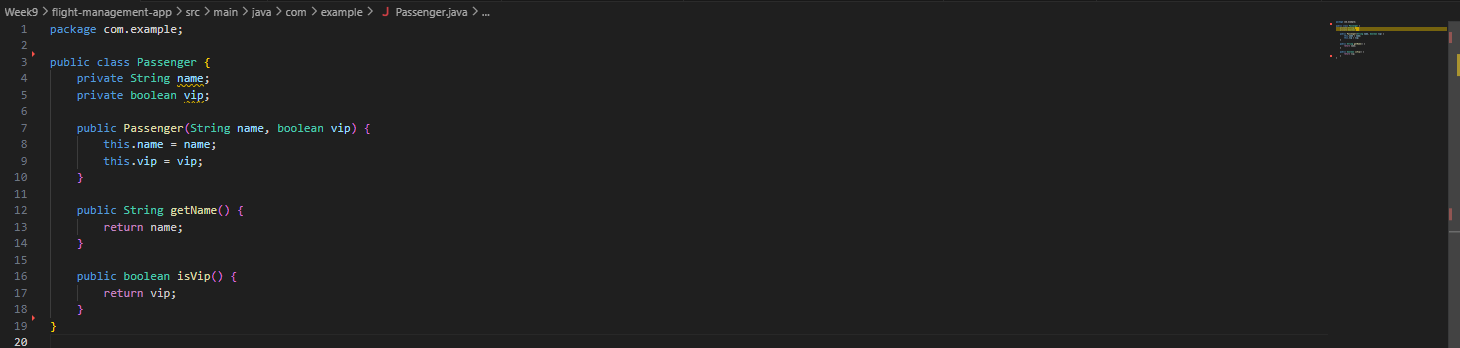
3. Memisahkan kode bisnis dan kode demo



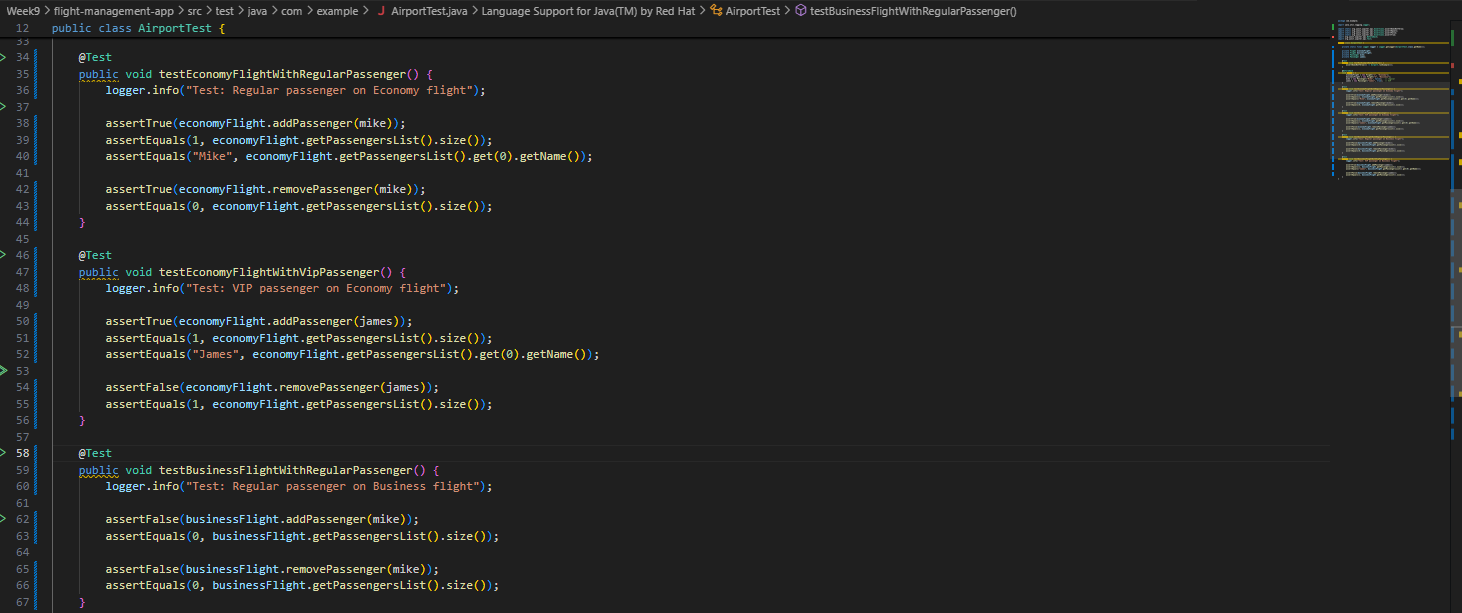
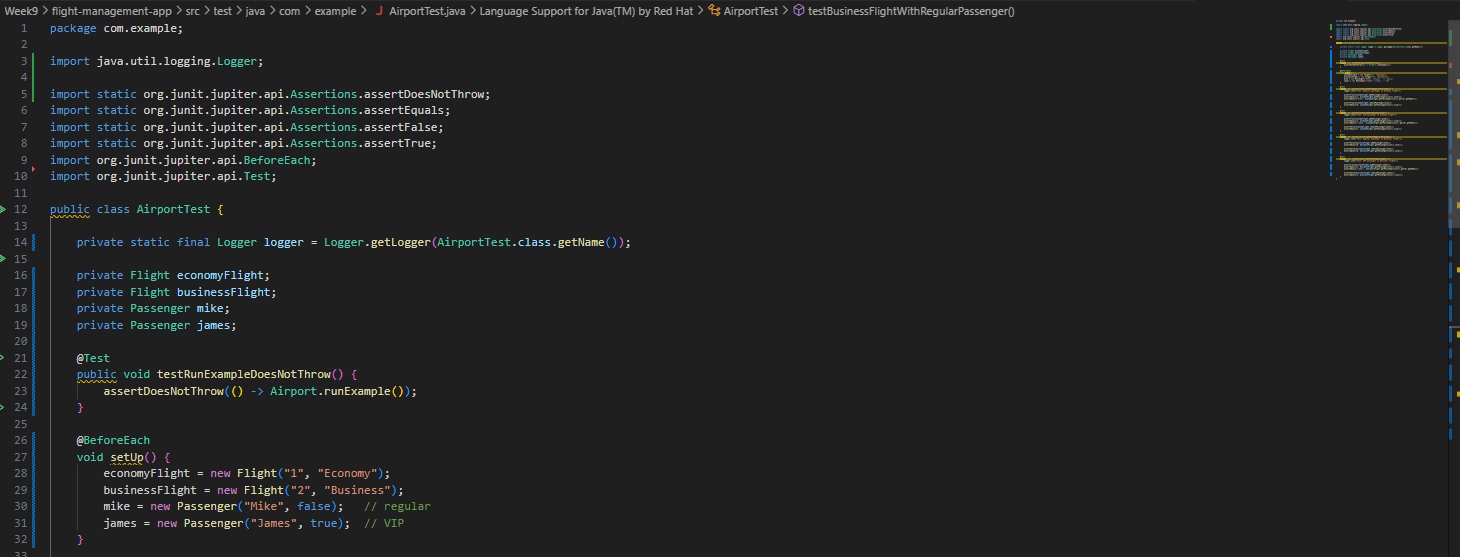
### 3. SETELAH REFACTOR

Airport.java

Flight.java

Passenger.java

AirportTest.java





Di Sonarqube setelah refactoring: 