**SOAL FROM MANUAL TO AUTOMATION TESTING**

1. Jelaskan jenis-jenis dari performance testing (minimal: 3)

Jawab:

<https://hariatylumbantobing.medium.com/performance-testing-pengertian-fokus-jenis-dan-contoh-7f1aaebbff8f>

1. Load Testing

Merupakan bagian dari teknik pengujian performance dimana tujuan pengujian ini adalah untuk menentukan target load dari sebuah sistem. Pengujian ini diperlukan saat adanya sebuah event musiman yang menyebabkan lalu lintas (traffic) sistem bertambah seiring berjalannya waktu.

1. Stress Testing

Merupakan bagian dari teknik pengujian performance yang dijalankan untuk mengetahui limit atau batas kemampuan dari sebuah sistem dengan cara membanjiri lalu lintas (traffic) sistem sampai sistem tersebut crash. Salah satu alasan utama dilakukannya pengujian ini adalah untuk menghindari terjadinya kegagalan sistem (system down).

1. Endurance Testing

Merupakan bagian dari teknik pengujian performance yang dijalankan untuk menganalisis behavior atau performa sistem dengan load yang normal namun dalam jangka waktu yang panjang. Pengujian ini dilakukan untuk menghindari terjadinya kebocoran memori (memory leak).

1. Sebutkan dan jelaskan tools yang dapat digunakan untuk melakukan performance testing (minimal: 3).

Jawab:

<https://toghr.com/10-tools-testing-terbaik/#:~:text=jMeter%20atau%20The%20Apache%20JMeterTM,bisa%20digunakan%20untuk%20performance%20test>

1. jMeter atau The Apache

JMeterTM merupakan aplikasi open source berbasis Java yang bisa digunakan untuk performance test. jMeter juga bisa digunakan untuk melakukan load/stress testing Web Application, FTP Application dan Database server test.

1. Postman merupakan sebuah aplikasi (berupa plugin) untuk browser chrome, fungsinya adalah sebagai REST Client atau istilahnya adalah aplikasi yang digunakan untuk melakukan uji coba REST API yang telah kita buat.
2. Katalon Studio berfungsi untuk melakukan pengujian API, Web, seluler, dan pengujian aplikasi desktop. Ini juga memiliki serangkaian fitur yang kaya untuk jenis pengujian ini dan mendukung banyak platform termasuk Windows, macOS, dan Linux

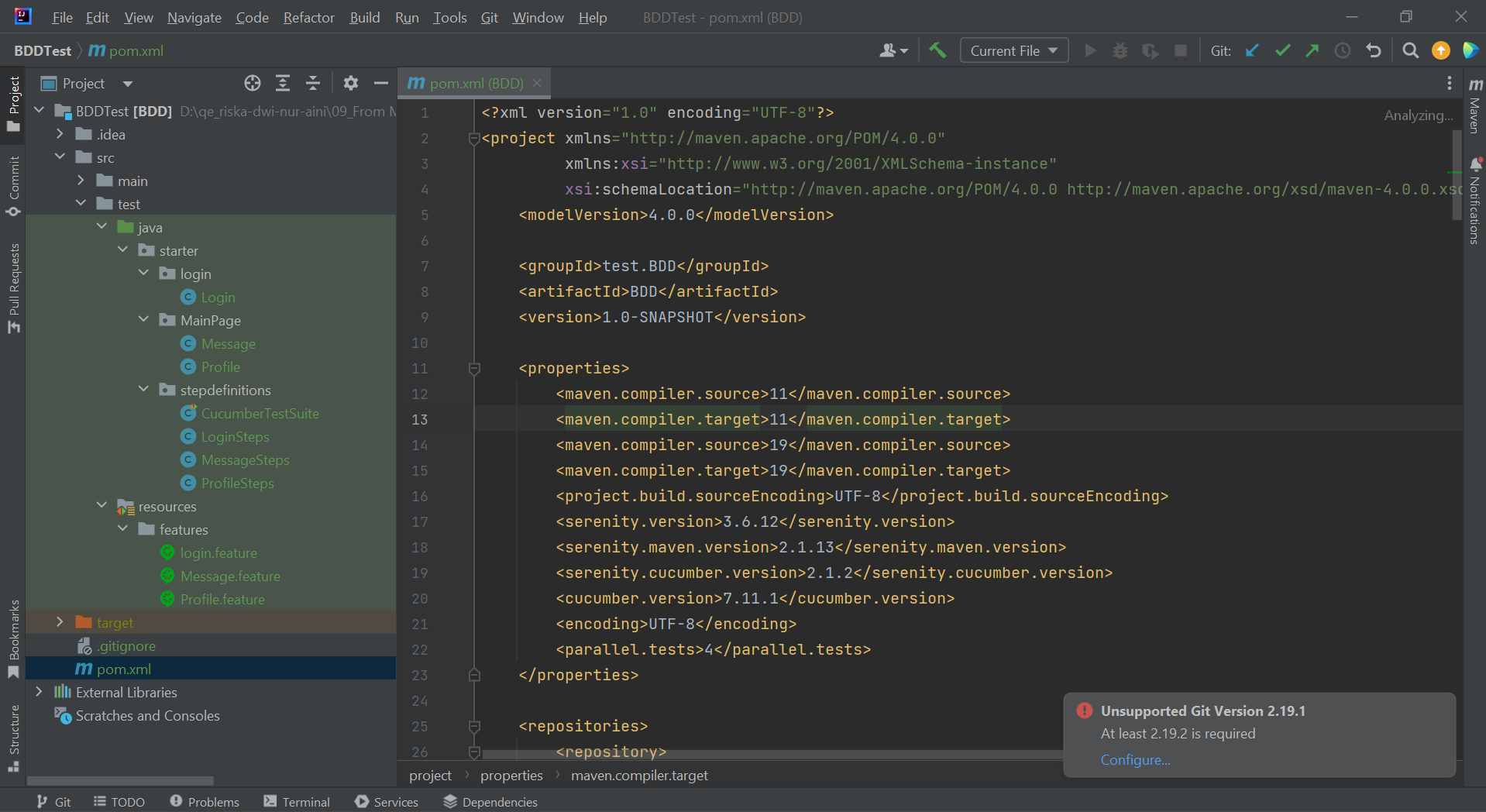
**SOAL BEHAVIOR DRIVEN DEVELOPMENT (BDD)**

**SOAL PRIORITAS 1**

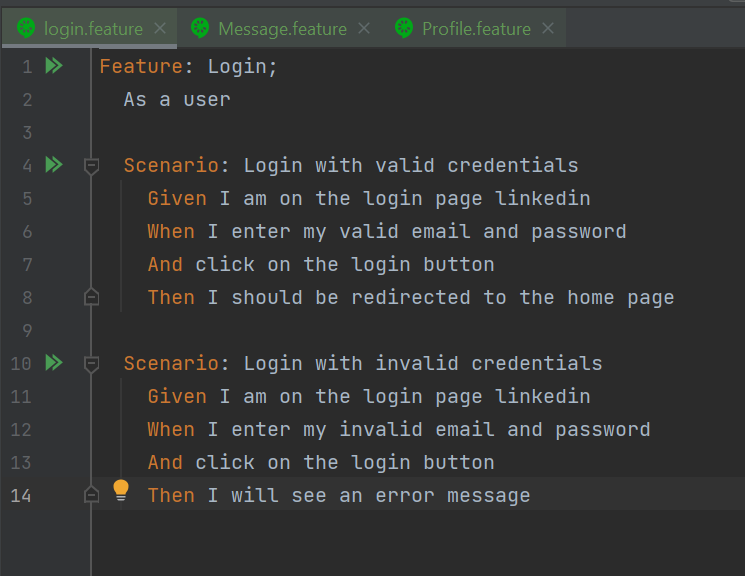
1. Lakukan testing pada aplikasi [https://www.linkedin.com/](https://www.linkedin.com/) dengan langkah-langkah sebagai berikut:
2. Buatlah scenario test sebanyak mungkin menggunakan format BDD pada fitur-fitur halaman utama [https://www.linkedin.com/](https://www.linkedin.com/) .
3. Buatlah project baru menggunakan intelliJ.
4. Buatlah file feature Cucumber
5. Buatlah class runner dan step
6. Buatlah class steps yang berisi method-method yang dijalankan dalam scenario test.
7. Jalankan class runner.
8. Tampilkan report HTML dari Cucumber.

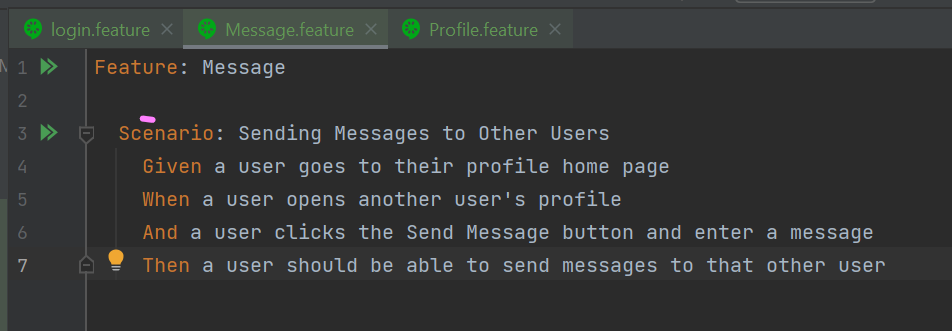
Jawab :

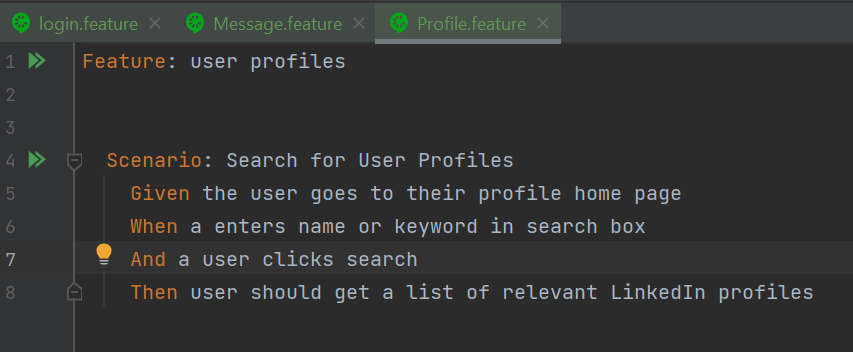
1. Pom.xml (project object model) adalah file konfigurasi pada proyek java yang digunakan untuk mengelola dependensi dan konfigurasi proyek. Pada intellij, POM dapat digunakan untuk proyek BDD (Behavior Driven Development) dengan menggunakan bahasa gherkin.ini adalah contoh gambar code POM pada projek BDD Linkedin.



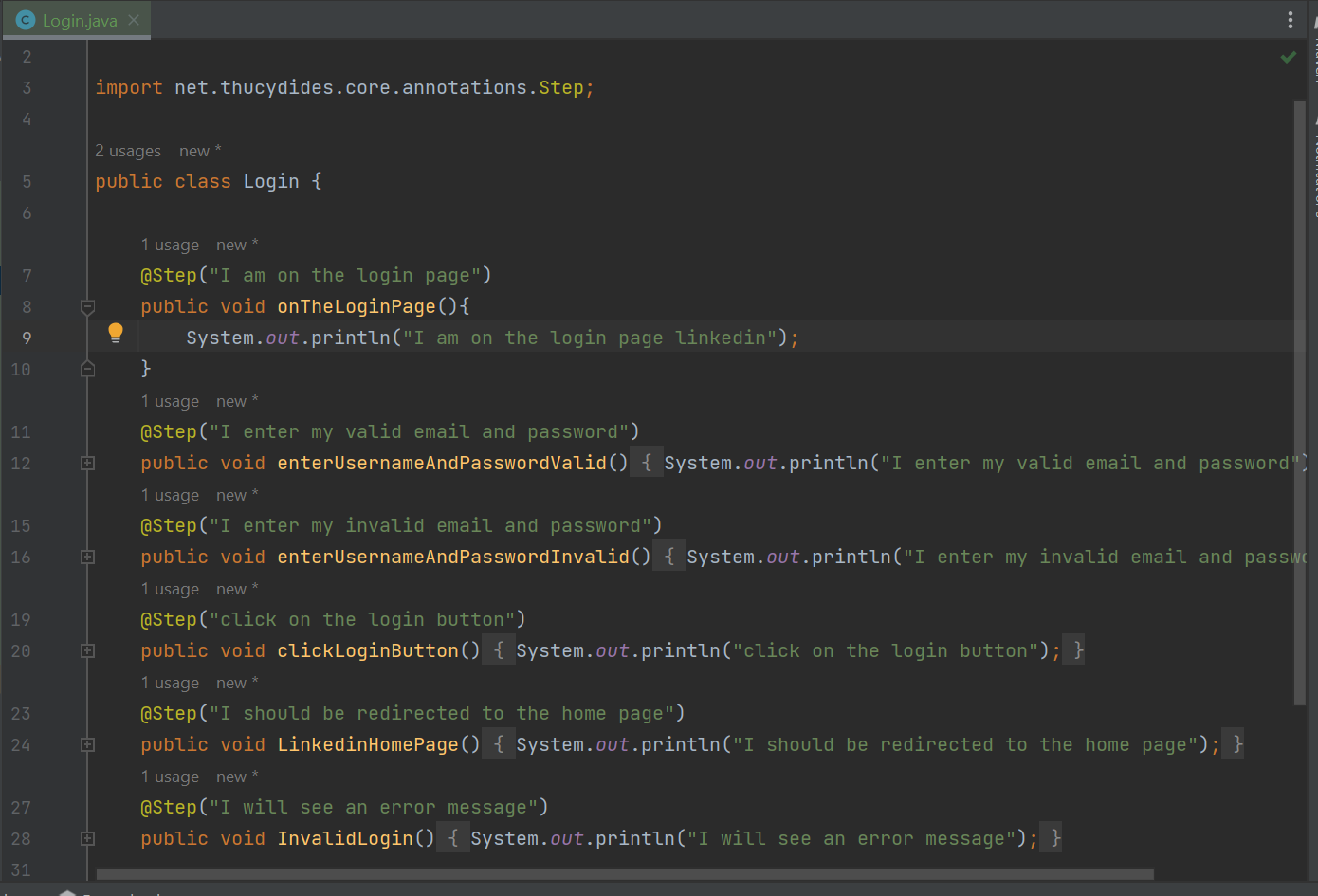
1. Setelah itu kita langsung membuat folder feature yang di dalamnya terdapat file profil.feature dan message.feature. Profil.feature adalah file yang digunakan dalam BDD untuk menulis skenario pengujian menggunakan bahasa gherkin

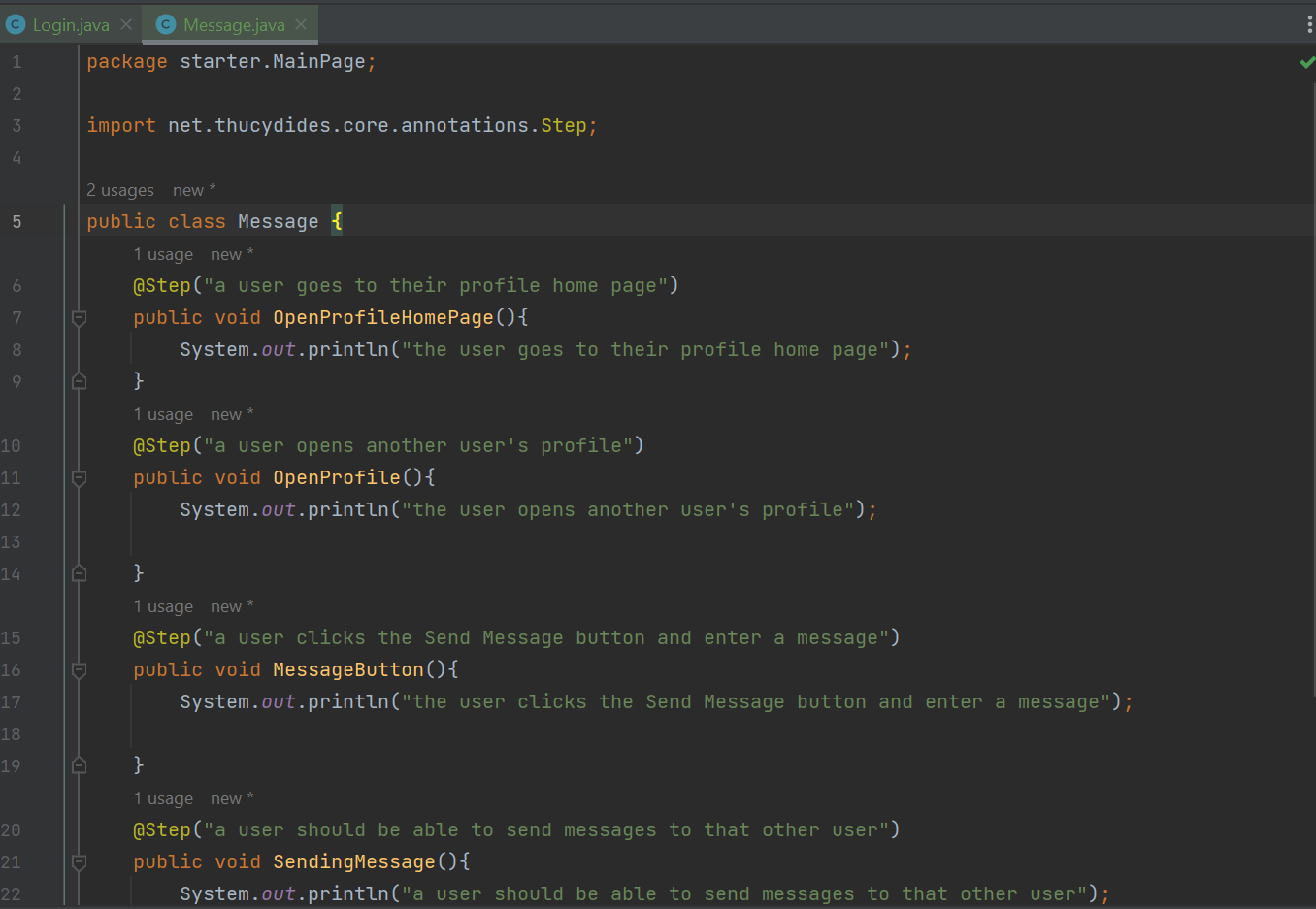


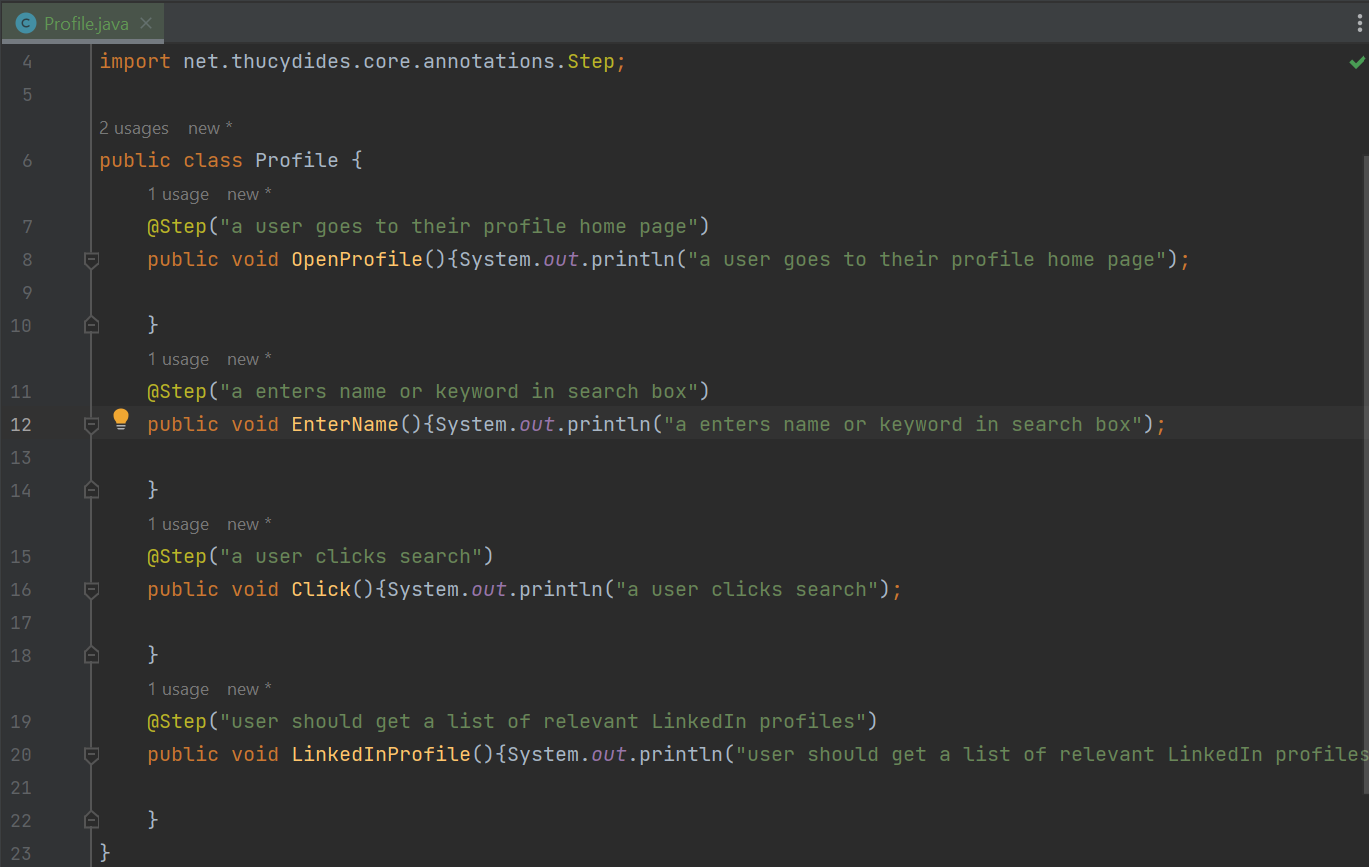




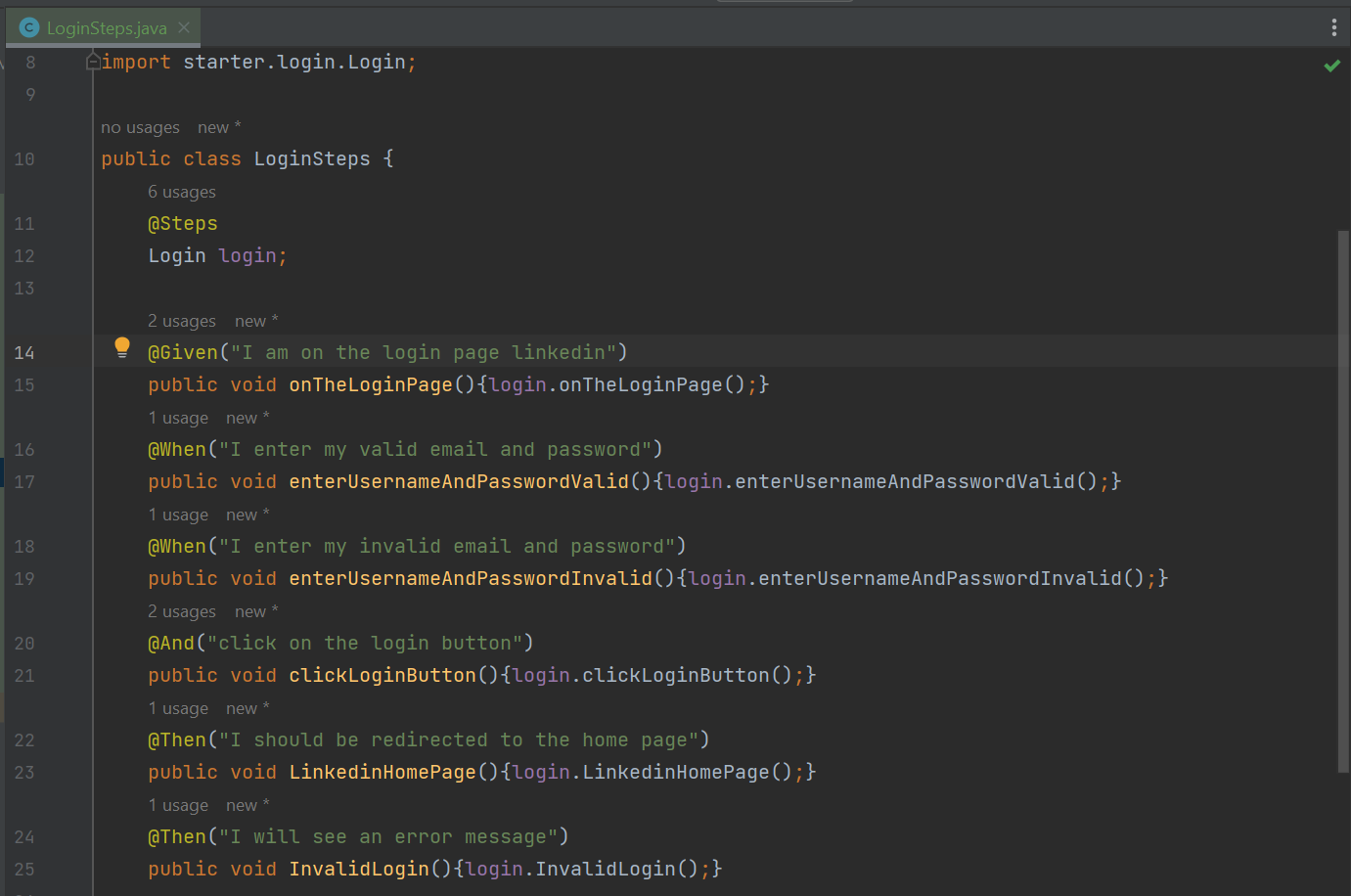
1. Setelah membuat file feature selanjutnya adalah membuat java class profile, login dan message pada package runner halaman utama. folder runner dapat digunakan untuk menyimpan skrip pengujian yang digunakan untuk menjalankan pengujian BDD. Skrip pengujian dapat dijalankan secara otomatis oleh alat pengujian atau oleh pengguna secara manual.

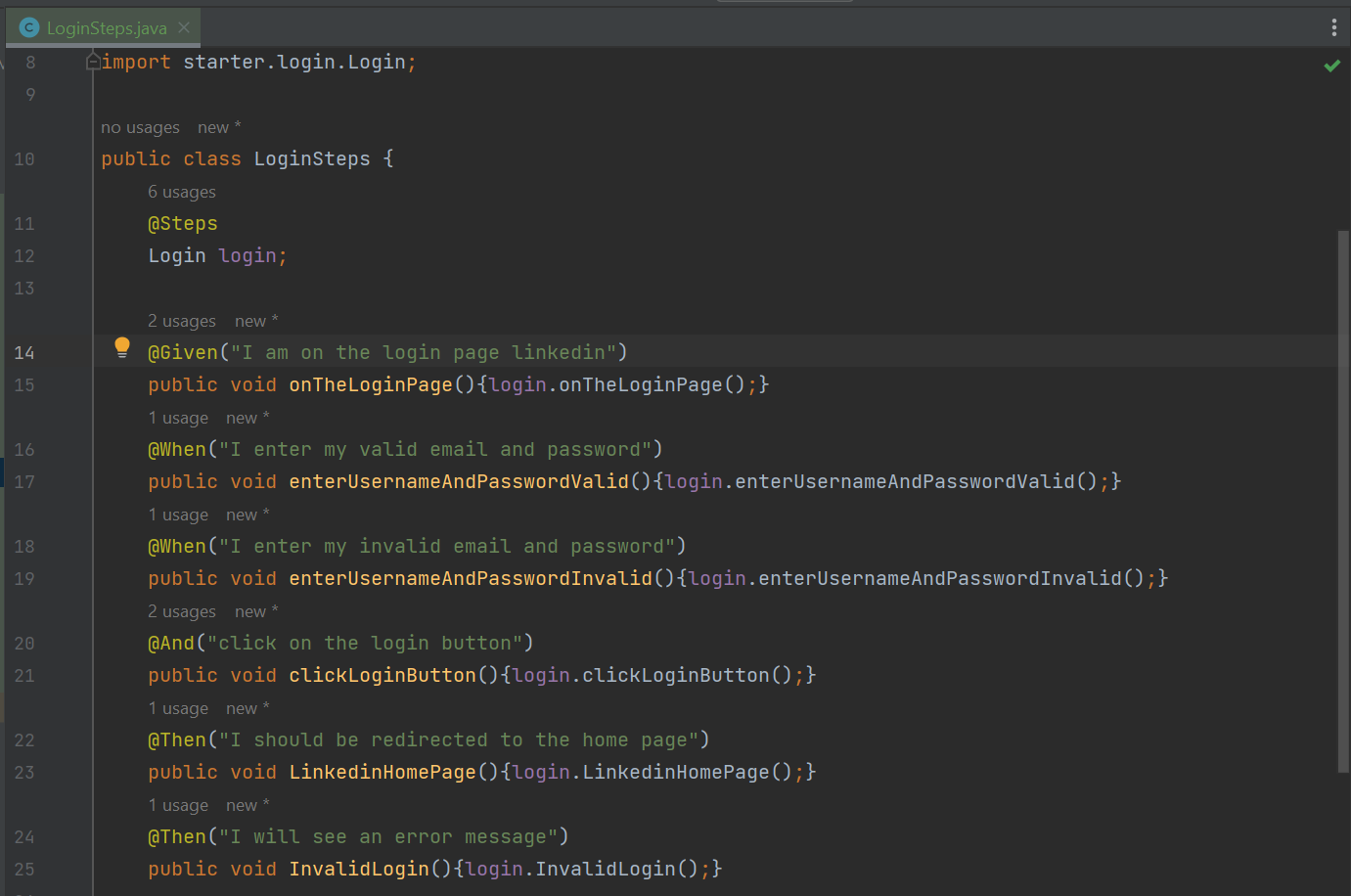


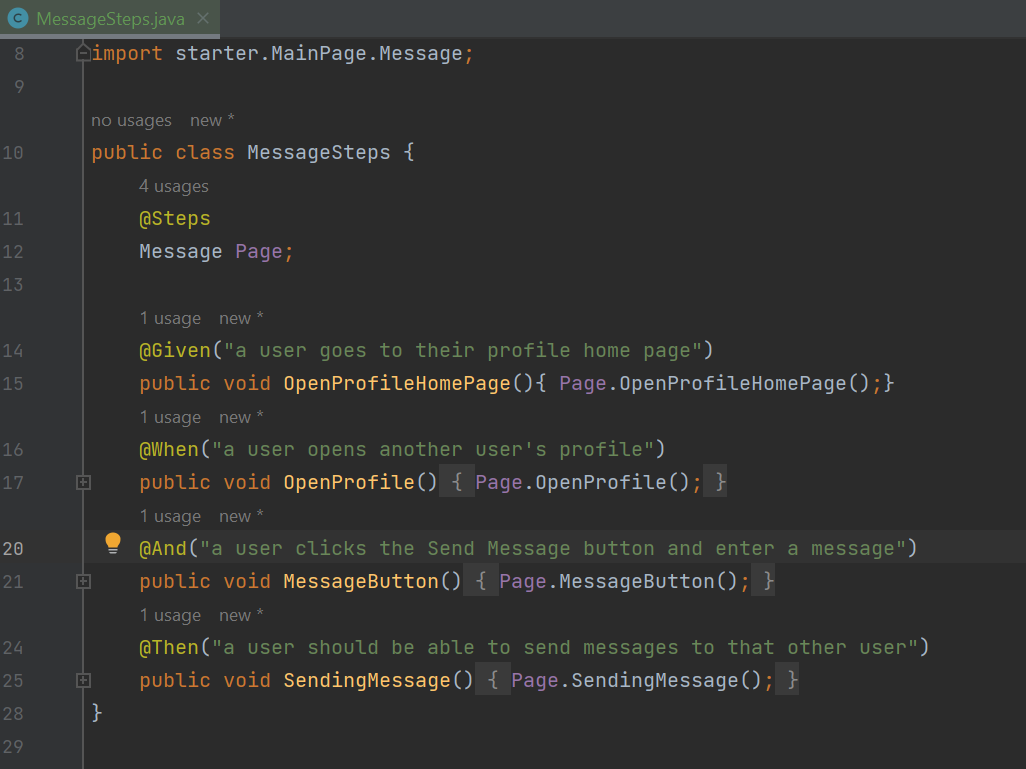




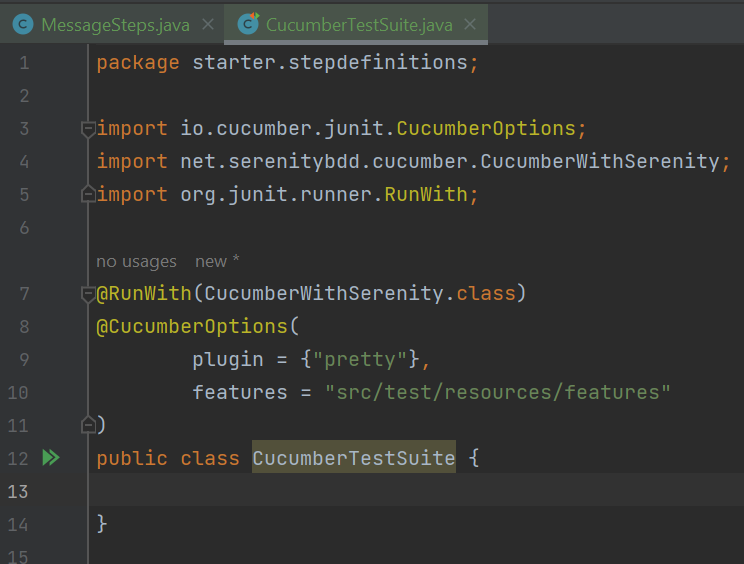
1. Setelah itu membuat folder stepdefinition yaitu untuk tempat scenario yang sama dengan feature dan memanggil file step yang tadi sudah ada pada code di file runner halaman utama.



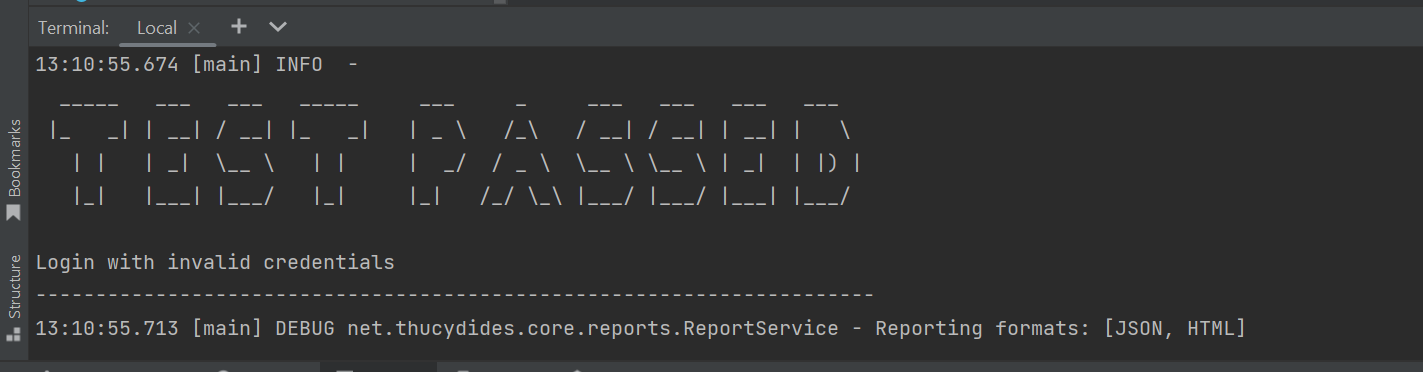




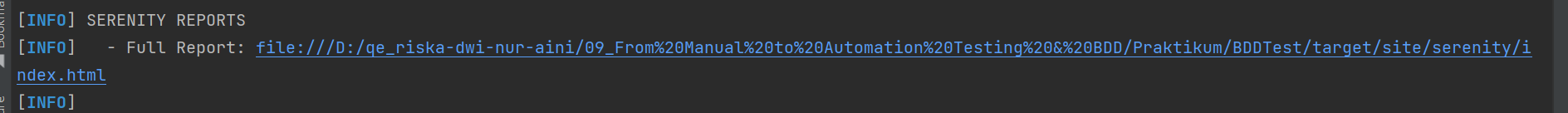
1. Selanjutnya Membuat java class CucumberTestSuit yaitu java class untuk menjalankan featur yang ada pada project.

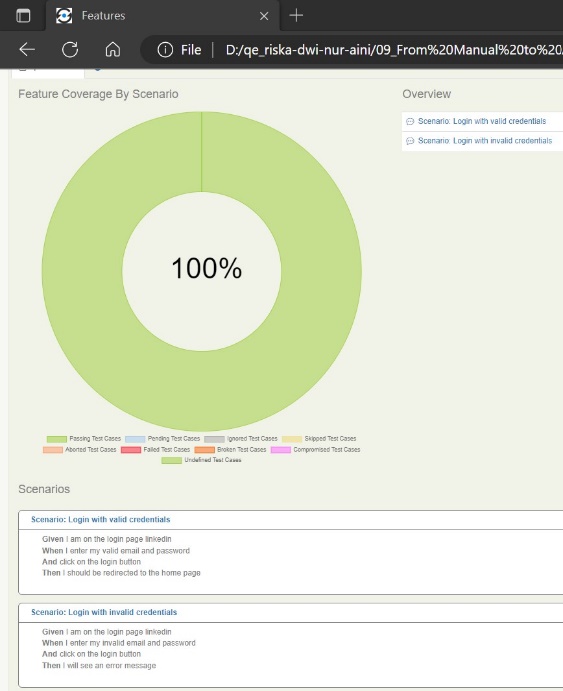


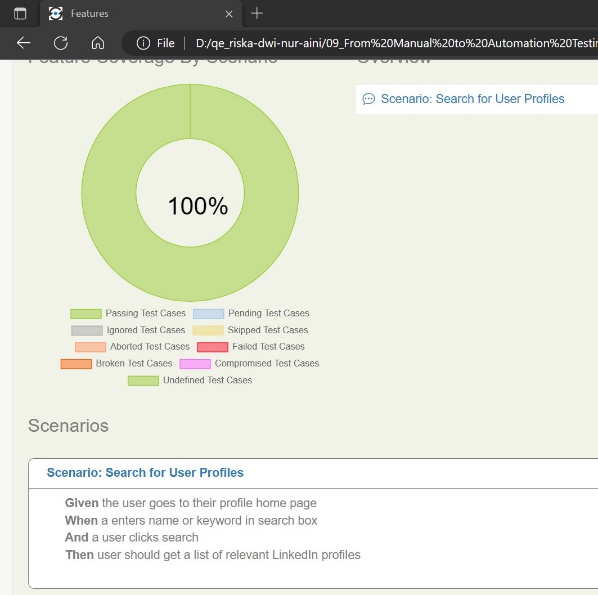
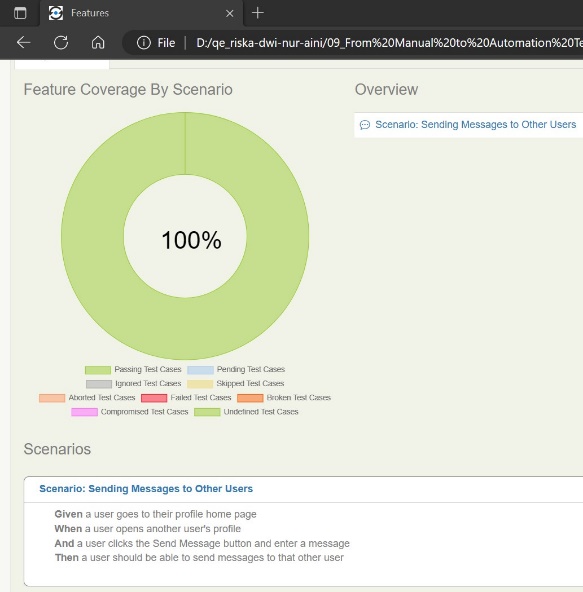
1. Setelah selesai semua tahap scenario BDD selanjutnya yaitu run project menggunakan terminal dengan mengetik mvn clean verify. Setelah itu akan muncul test stared dan tes passed seperti gambar dibawah ini.



1. Setelah itu akan muncul link URL dan Ketika di klik tampilannya seperti ini





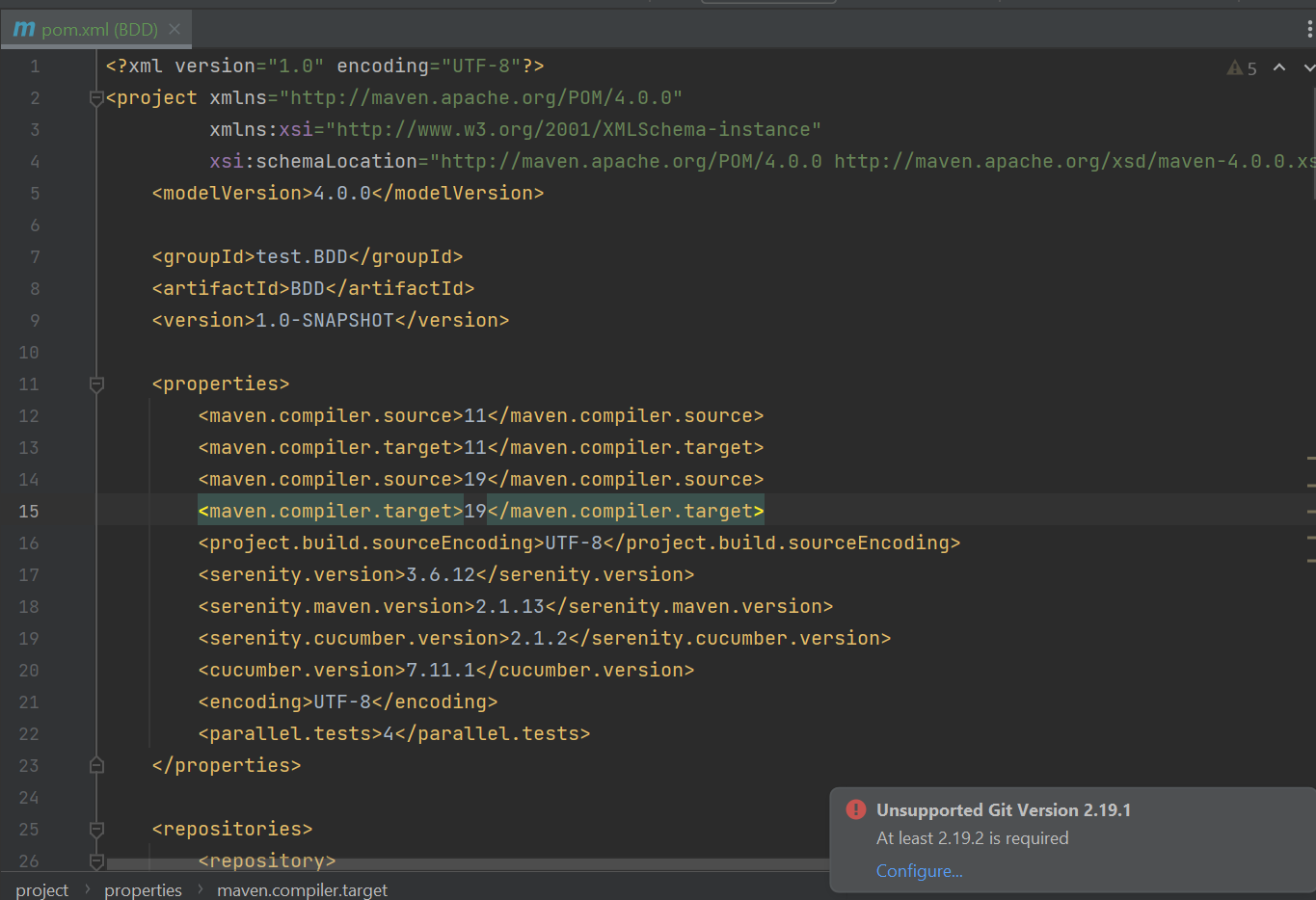


**SOAL PRIORITAS 2 (20)**

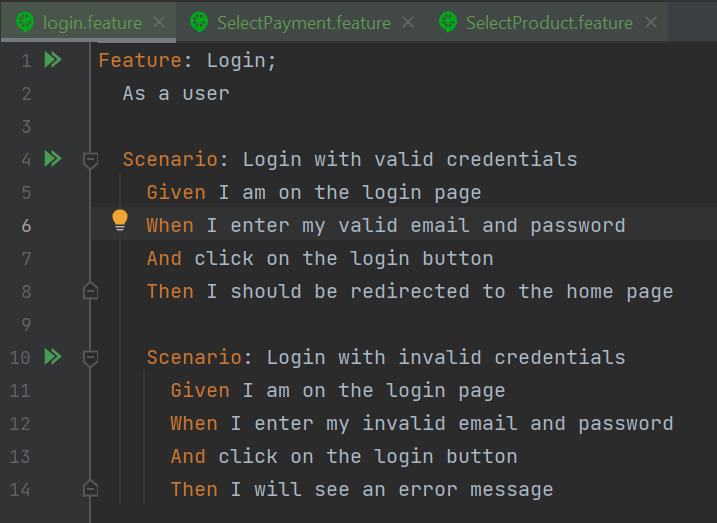
1. Lakukan testing pada aplikasi [https://www.sepulsa.com/](https://www.sepulsa.com/) dengan langkah-langkah sebagai berikut:
2. Buatlah scenario test menggunakan format BDD pada fitur login, pilih produk dan pilih metode pembayaran.
3. Buatlah project baru menggunakan intelliJ.
4. Buatlah file feature Cucumber
5. Buatlah class runner dan step
6. Buatlah class steps yang berisi method-method yang dijalankan dalam scenario test.
7. Jalankan class runner.
8. Tampilkan report HTML dari Cucumber.

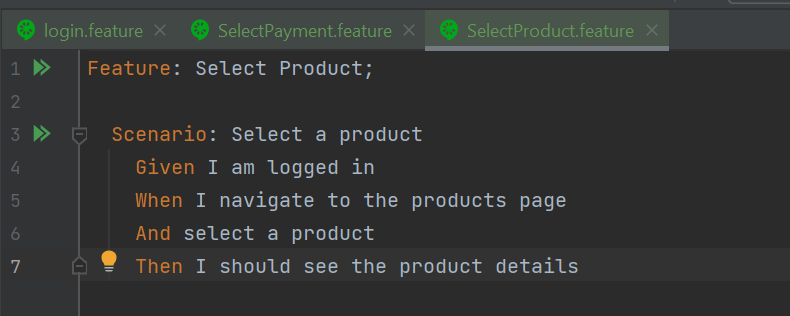
Jawab:

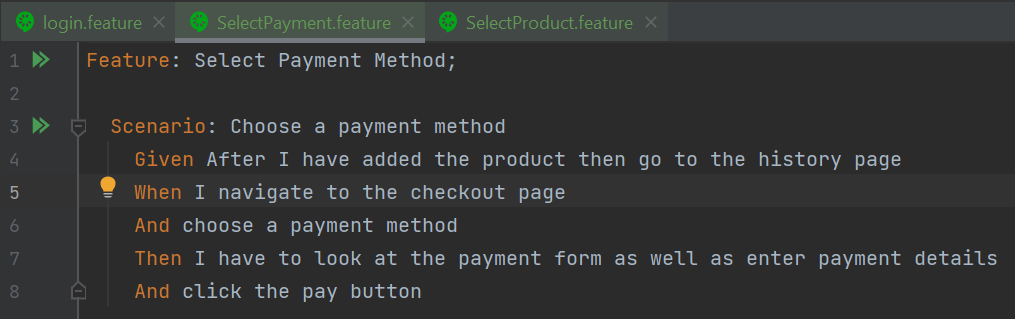
1. Pom.xml (project object model) adalah file konfigurasi pada proyek java yang digunakan untuk mengelola dependensi dan konfigurasi proyek. Pada intellij, POM dapat digunakan untuk proyek BDD (Behavior Driven Development) dengan menggunakan bahasa gherkin.ini adalah contoh gambar code POM pada projek BDD Linkedin.



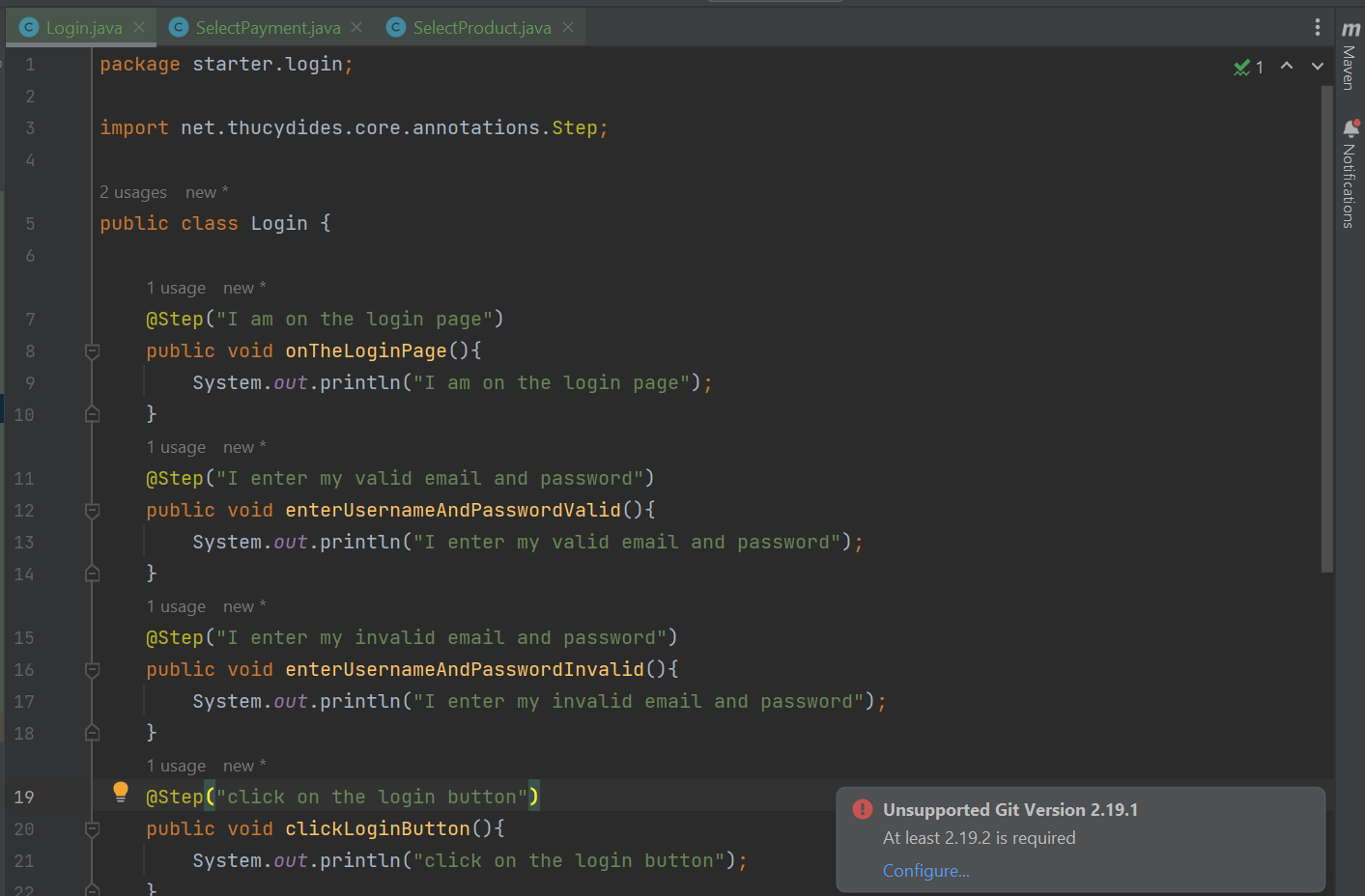
1. Setelah itu kita langsung membuat folder feature yang di dalamnya terdapat file Login.feature dan SelectProduct.feature. SelectPayment.feature adalah file yang digunakan dalam BDD untuk menulis skenario pengujian menggunakan bahasa gherkin

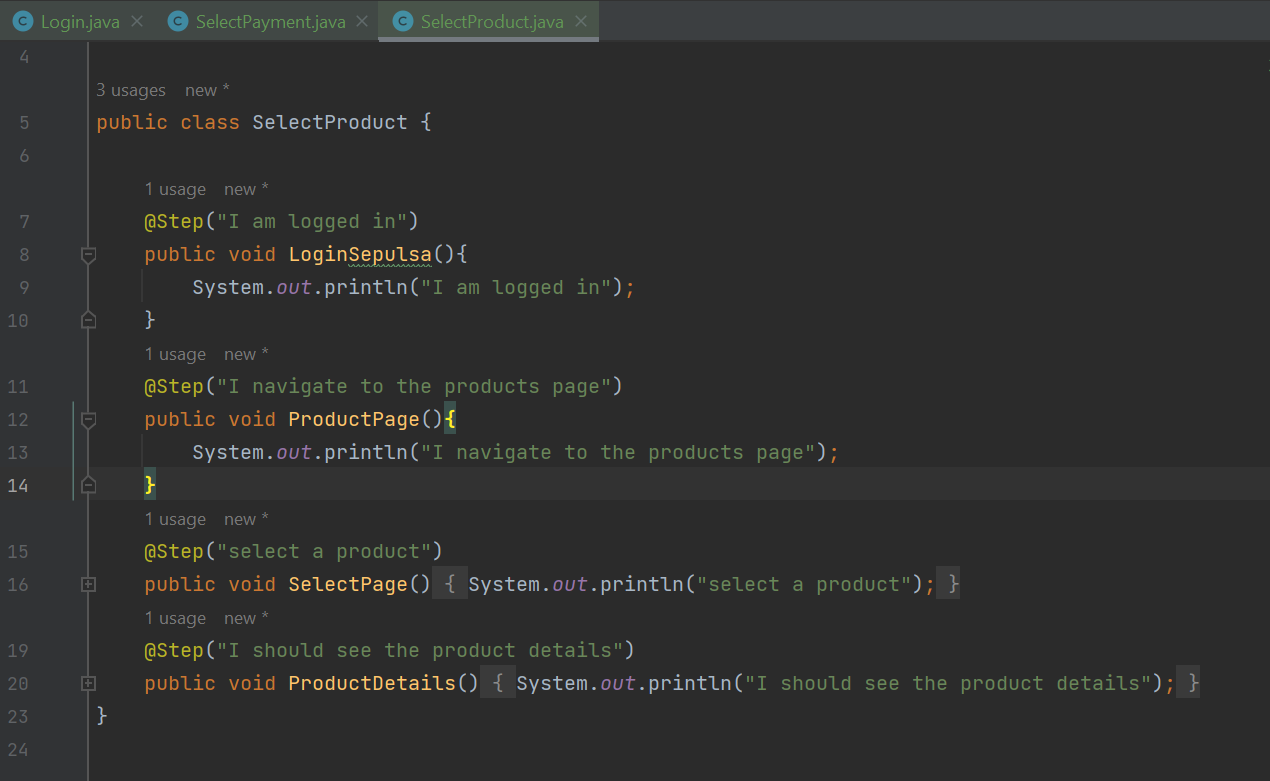


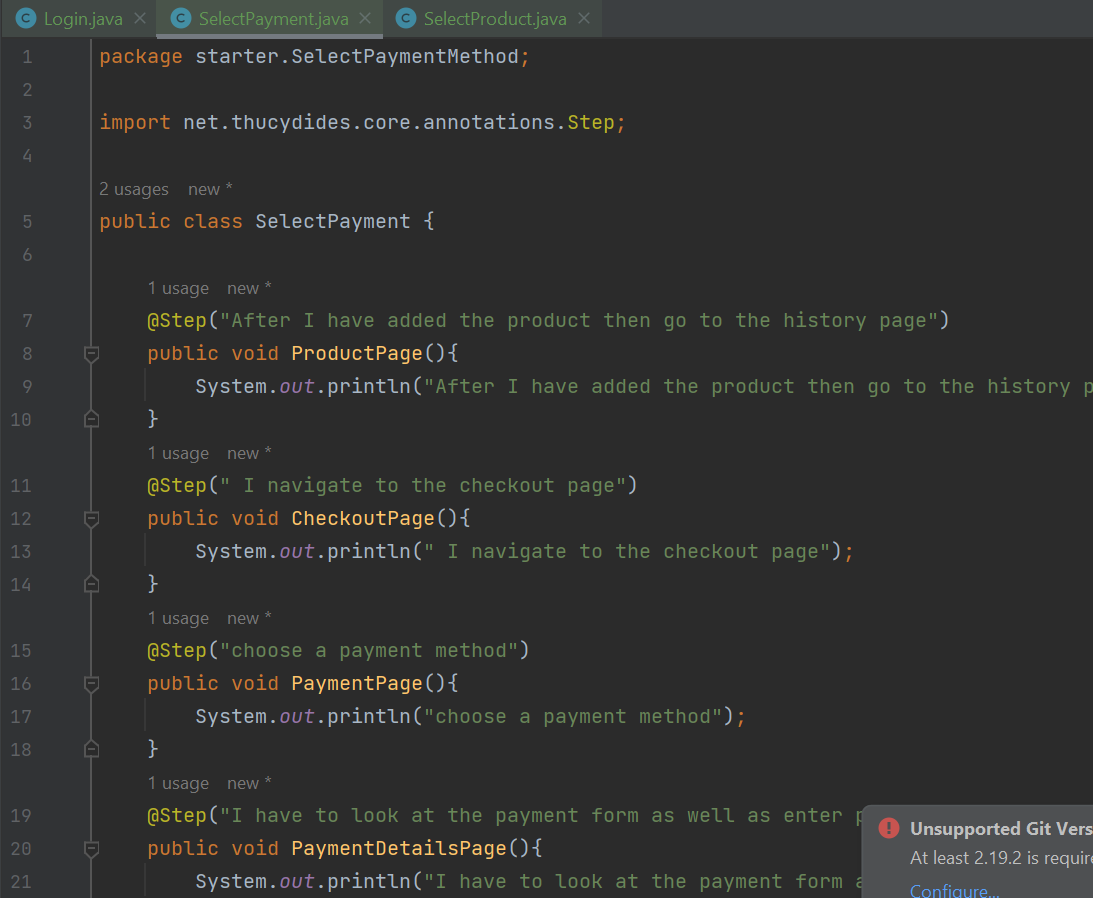




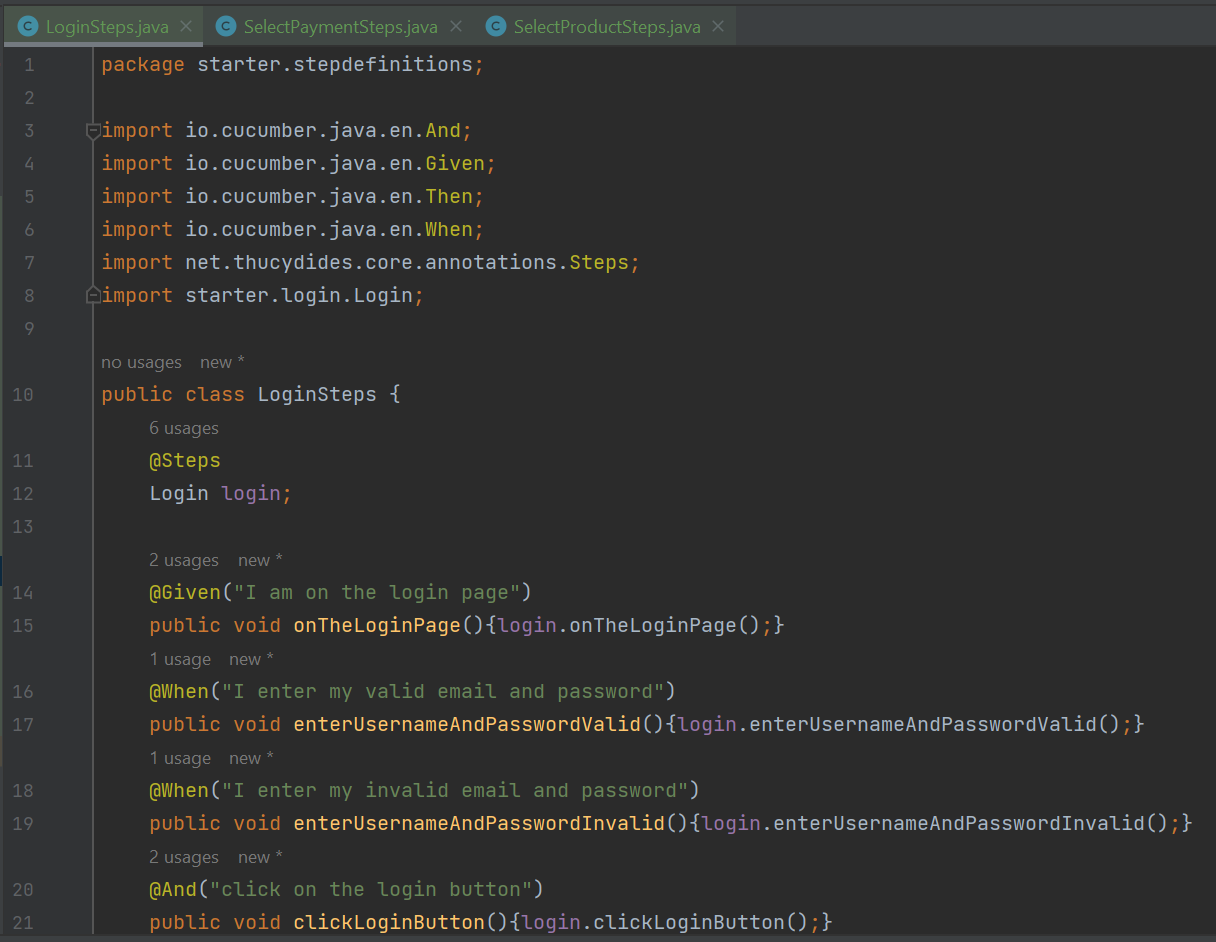
1. Setelah membuat file feature selanjutnya adalah membuat java class login, select product dan select payment pada package runner halaman utama. folder runner dapat digunakan untuk menyimpan skrip pengujian yang digunakan untuk menjalankan pengujian BDD. Skrip pengujian dapat dijalankan secara otomatis oleh alat pengujian atau oleh pengguna secara manual.

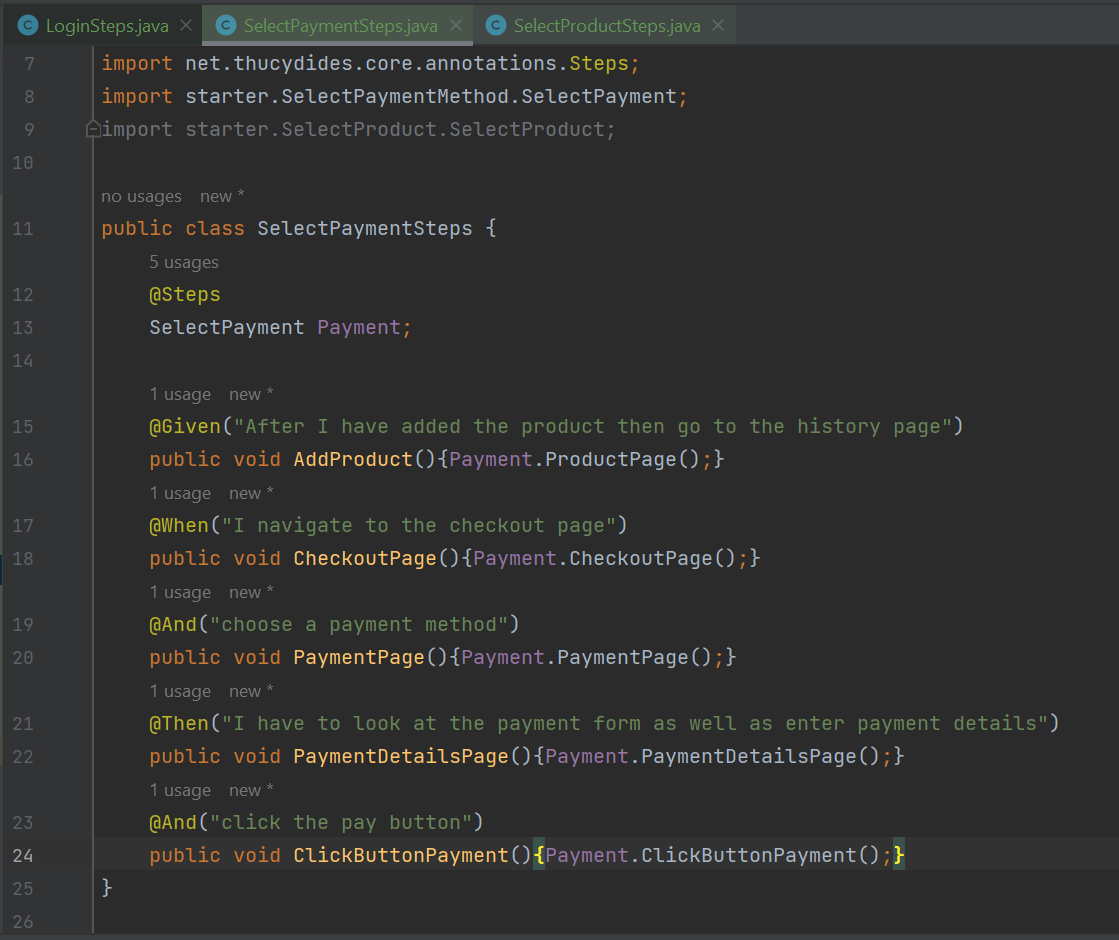


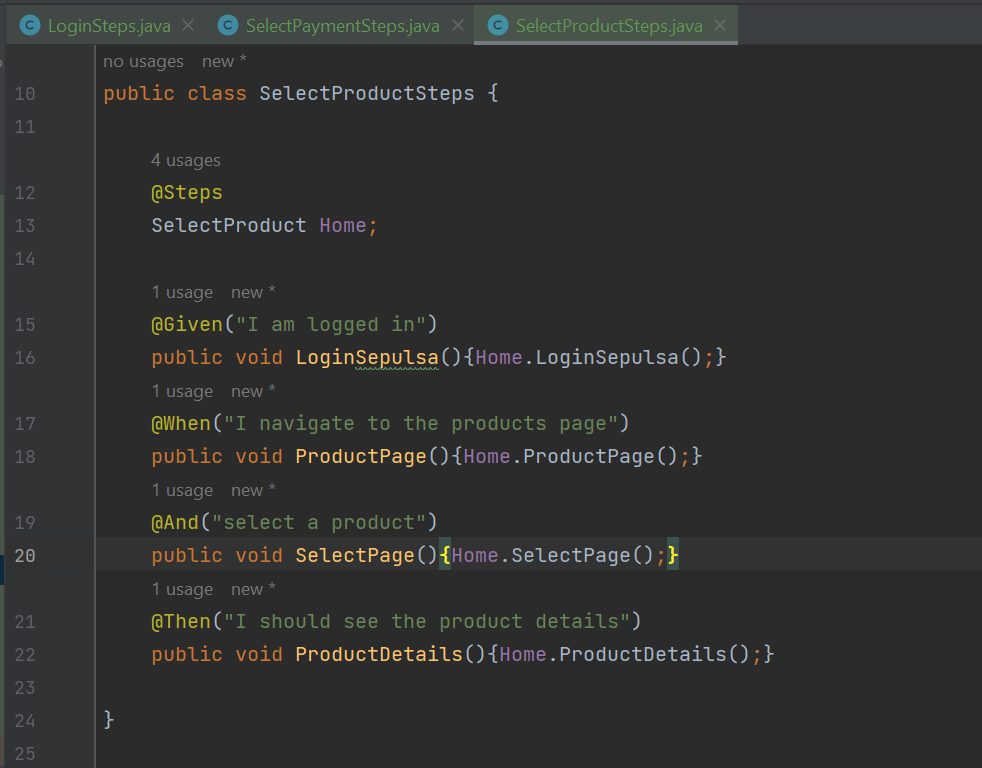




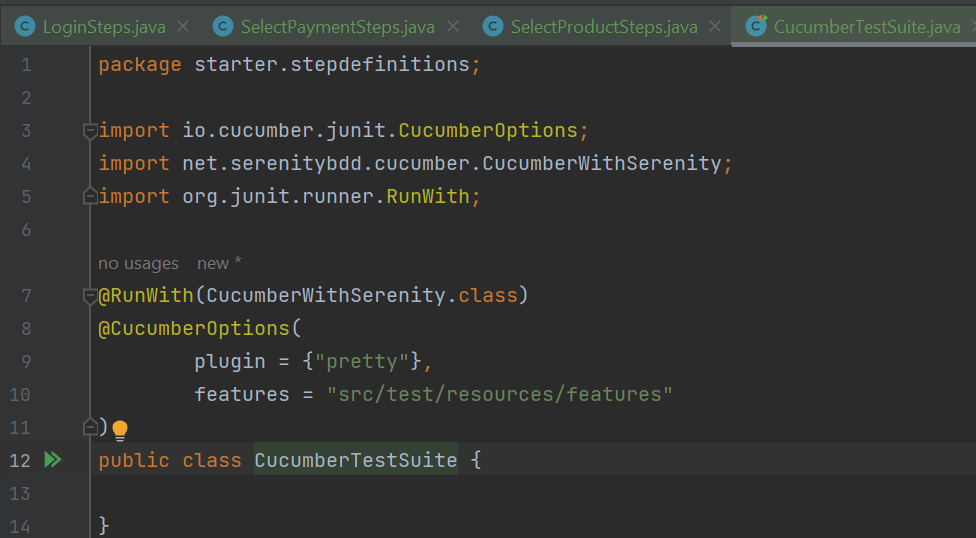
1. Setelah itu membuat folder stepdefinition yaitu untuk tempat scenario yang sama dengan feature dan memanggil file step yang tadi sudah ada pada code di file runner halaman utama.



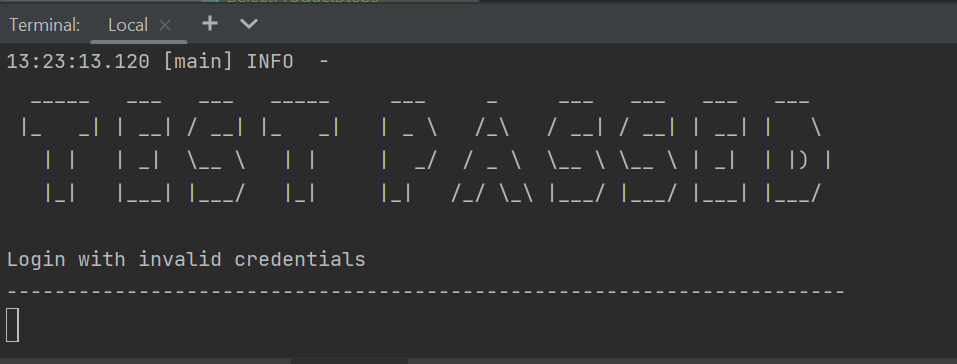




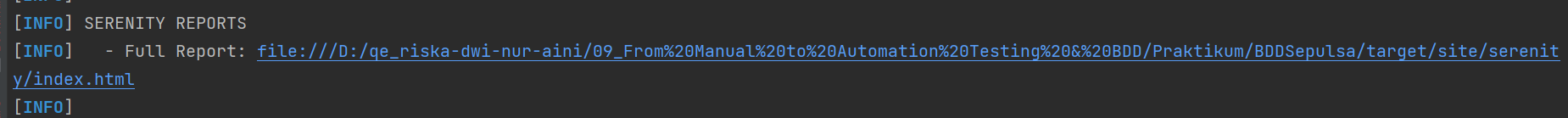
1. Selanjutnya Membuat java class CucumberTestSuit yaitu java class untuk menjalankan featur yang ada pada project.

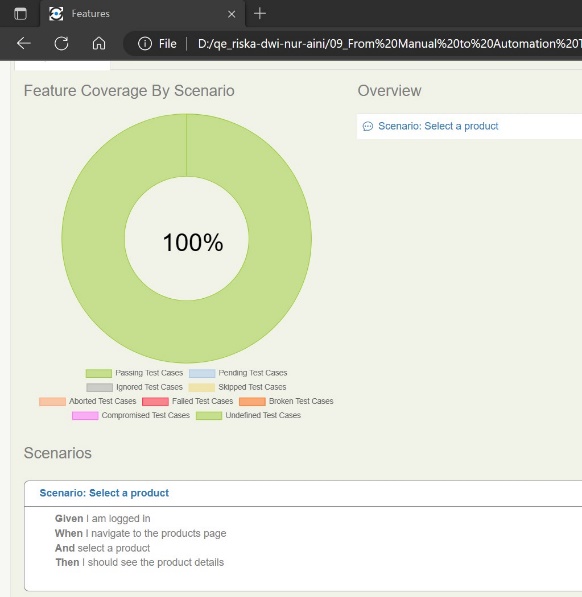
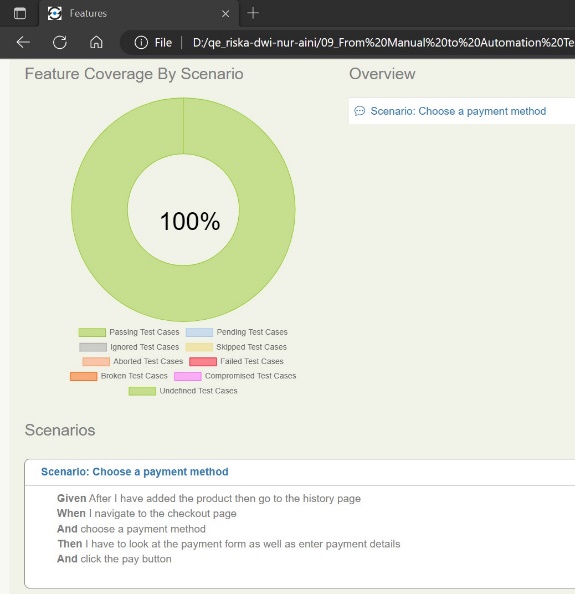
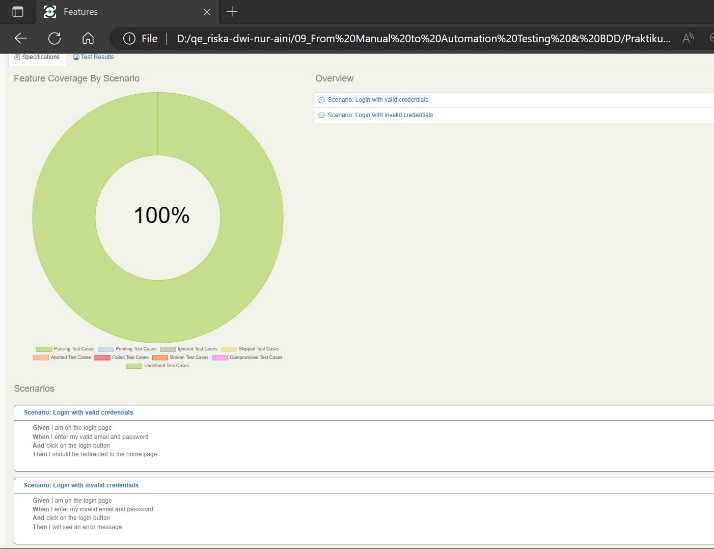


1. Setelah selesai semua tahap scenario BDD selanjutnya yaitu run project menggunakan terminal dengan mengetik mvn clean verify. Setelah itu akan muncul test stared dan tes passed seperti gambar dibawah ini.



1. Setelah itu akan muncul link URL dan tampilannya seperti ini



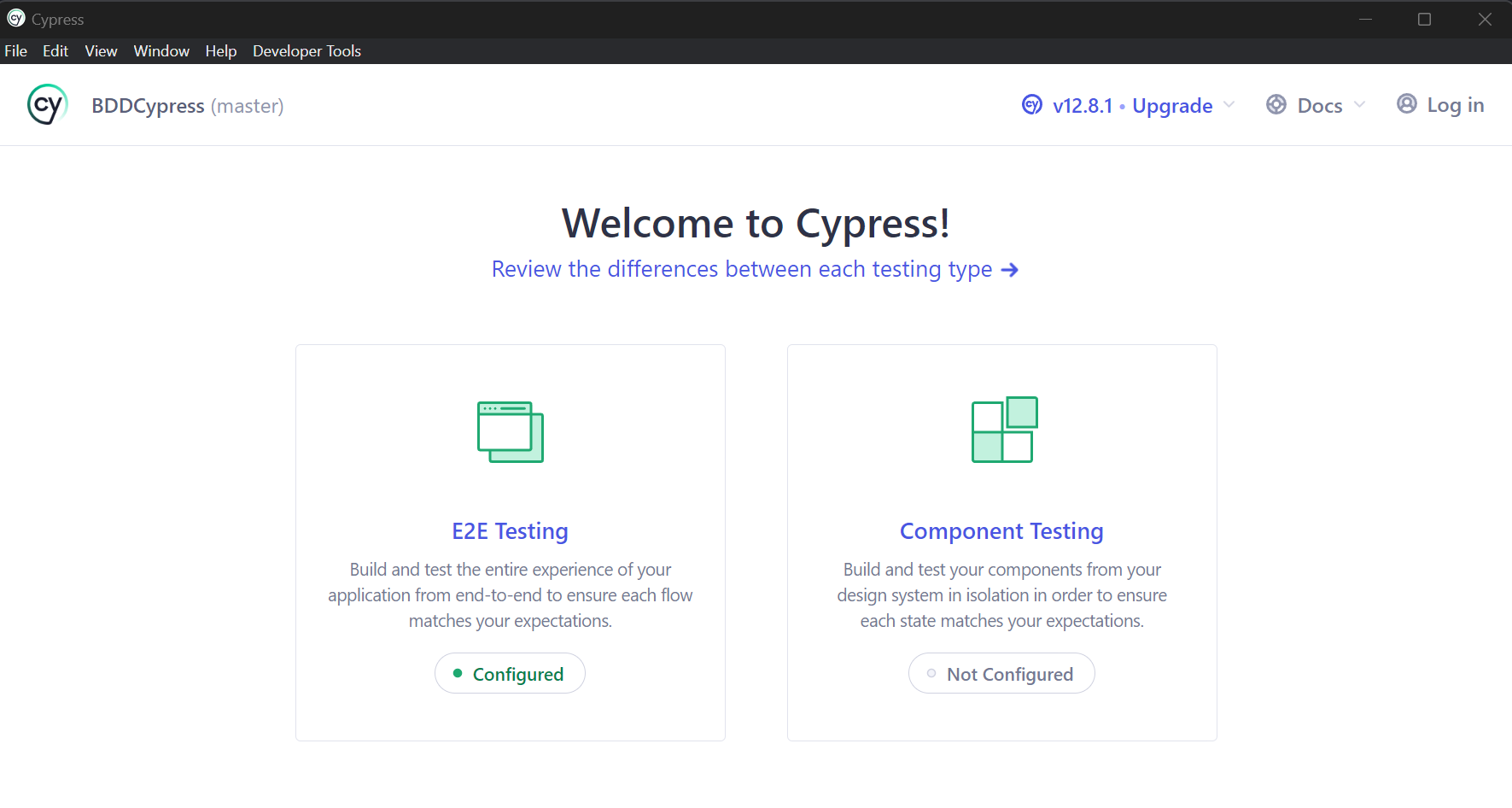


### Soal Eksplorasi (20)

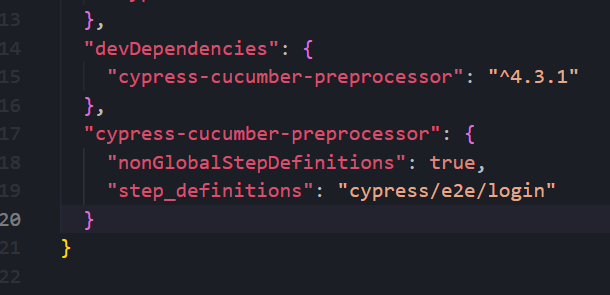
1. Lakukan testing pada aplikasi <https://www.sepulsa.com/> dengan langkah-langkah sebagai berikut:
   1. Menggunakan **cypress** sebagai tools untuk melakukan testing. Referensi penggunaan BDD dengan cypress dapat dilihat [disini](https://www.toolsqa.com/cypress/bdd-automation-framework-in-cypress/).
   2. Buatlah scenario test menggunakan format BDD pada fitur login, pilih produk dan pilih metode pembayaran.
   3. Tampilkan hasil testing dengan menggunakan cypress.

Jawab:

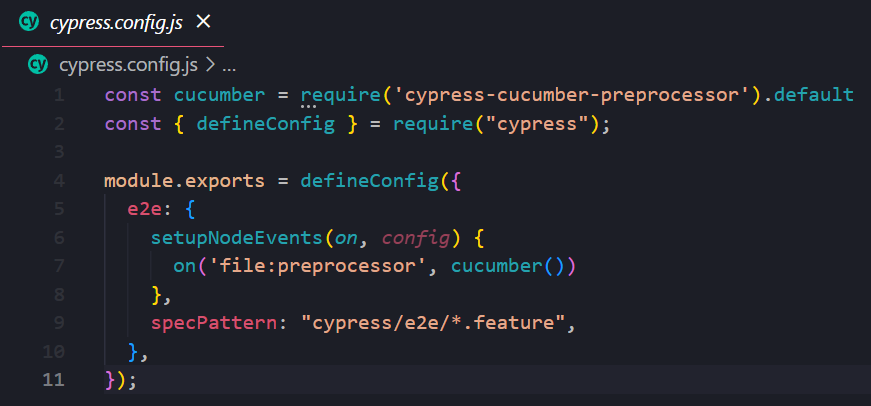
1. Membuat folder baru di file explore setelah itu membuka visual studio code dan open folder. Setelah itu kita lalu ketik di terminal npm init fungsi tersebut adalah perintah yang digunakan untuk membuat proyek Node.js baru dan membuat file package.json.
2. Setelah itu kita mengetik di terminal npm install cypress --save--dev perintah tersebut digunakan untuk Cypress sebagai dependensi pengembangan untuk proyek node.js. Setelah menjalankan perintah tersebut cypres akan di install di direktori nod -module proyek dan informasi baru akan ditambahkan ke dalam file package.json
3. Setelah itu kita mengetik lagi perintah npm install --save-dev cypress-cucumber-preprocessor. Perintah tersebut berfungsi untuk menginstal dan menyimpan cypress cucumber sebagai dependensi pengembangan di proyek.js. Dan perintah npx cypres open adalah untuk membuka cypresnya.
4. Setelah itu akan muncul tampilan cypress dan langsung pilih e2e. End to End (E2E) testing pada cypress bertujuan untuk menguji bagian aplikasi secara keseluruhan dari permulaan sampai akhuir melalui pengujian jalur yang diambil olleh pengguna dan interaksi



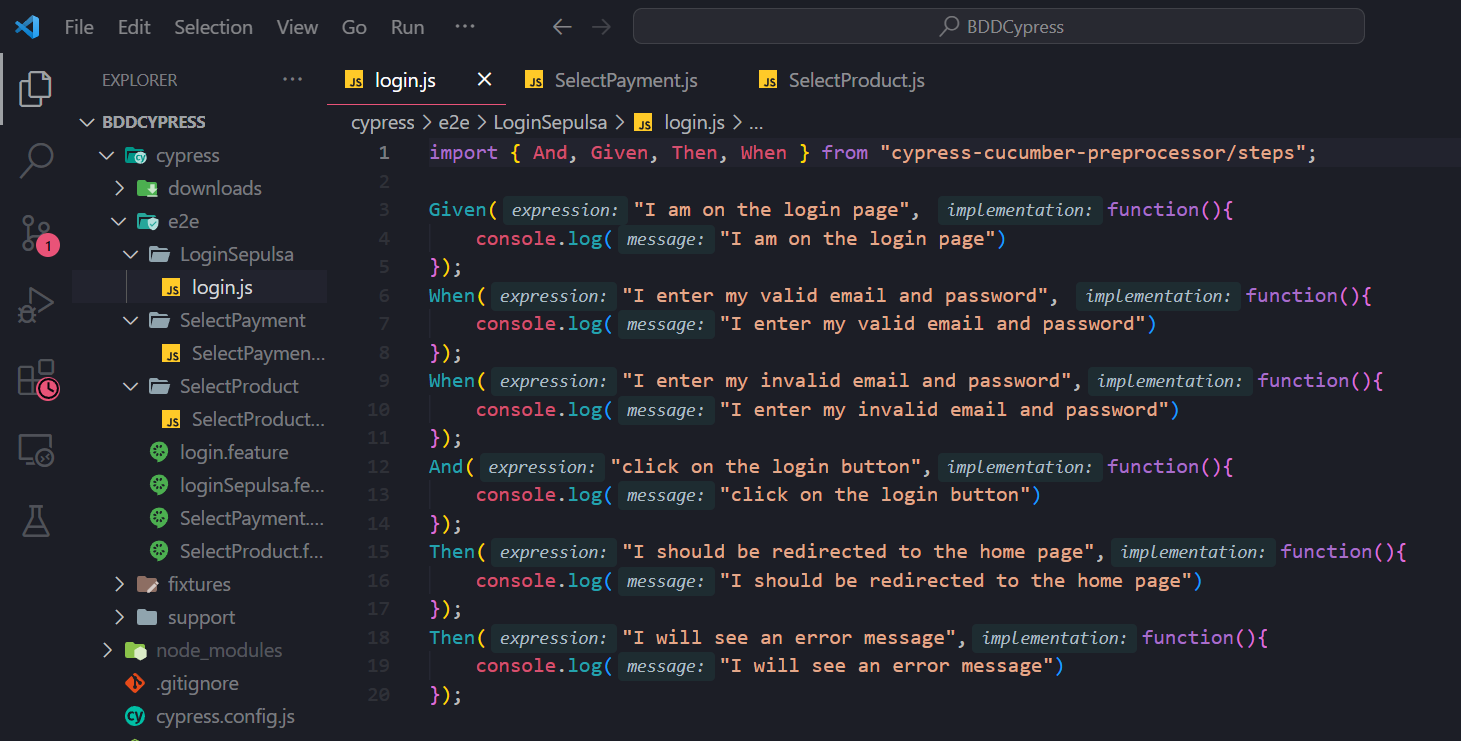
1. Setelah itu kita menambahkan code di file package.json

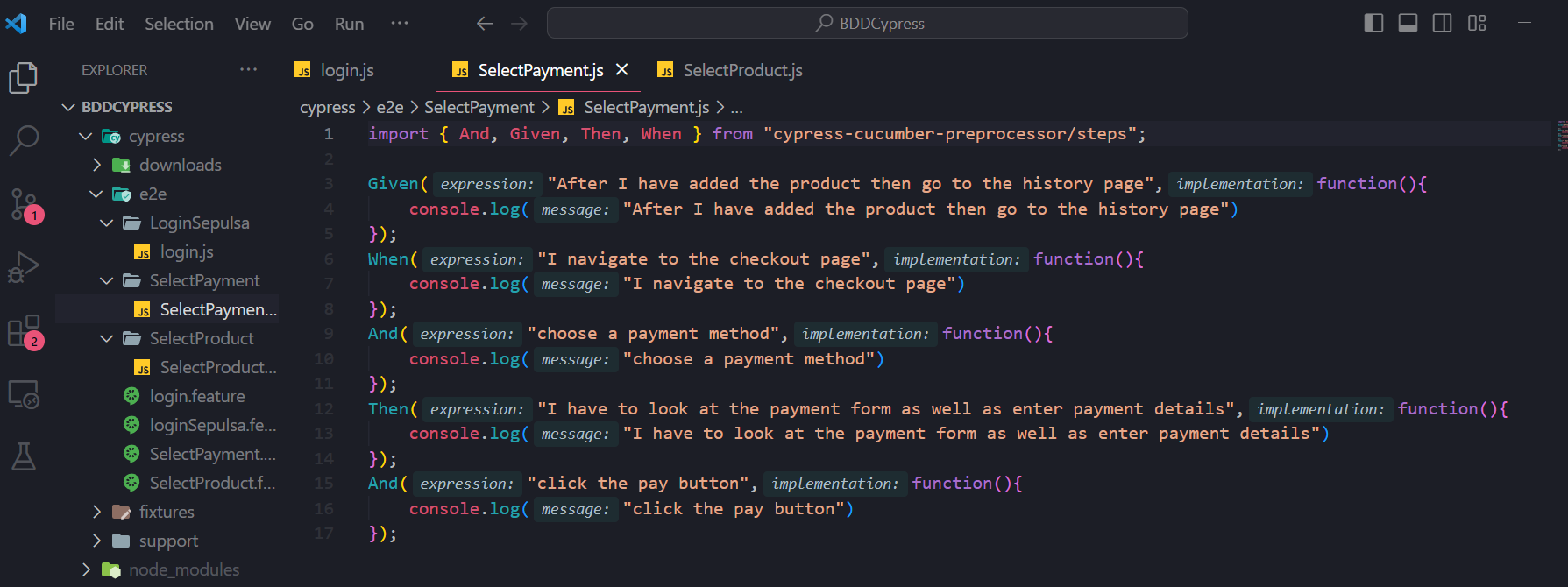


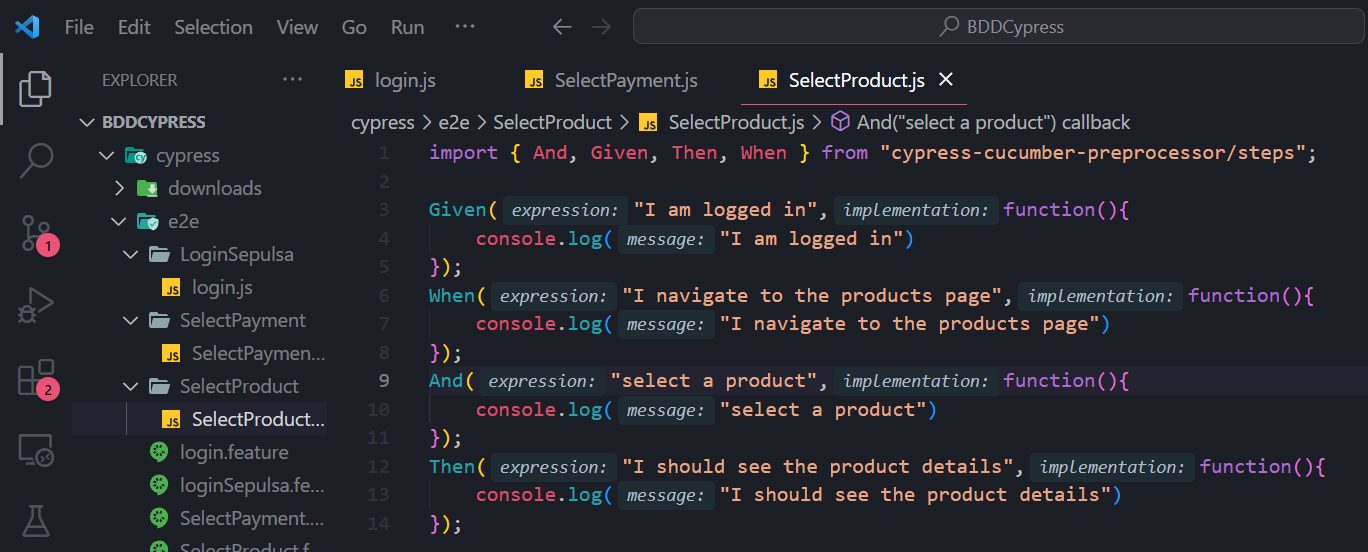
1. Setelah itu menambahkan code pada line 8 sampai 10



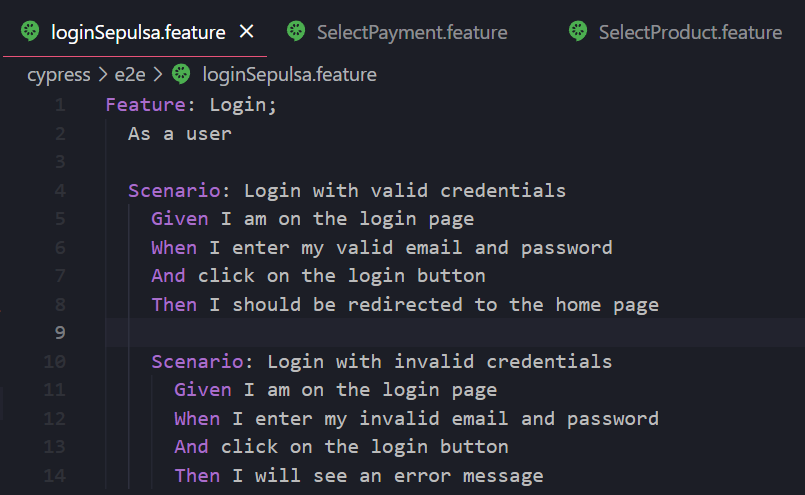
1. Setelah membuat folder pada e2e kemudian ketikkan source code seperti dibawah ini

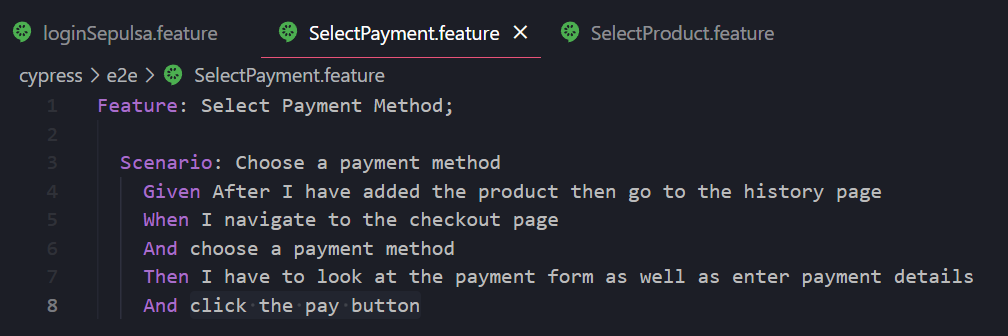


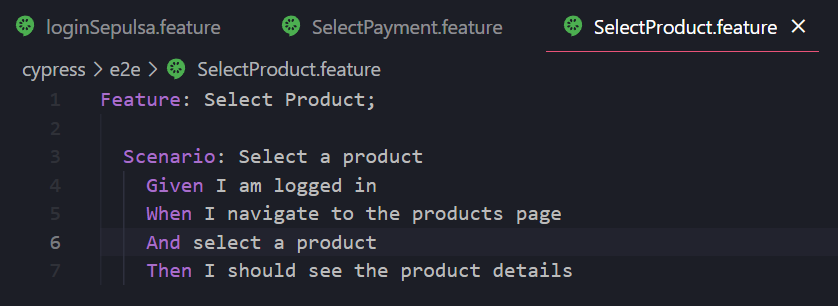




1. Setelah itu membuat file scenario







1. Ketikkan npx cypress open untuk melihat hasil run maka tampilannya akan seperti dibawah ini

