Cypress.Commands.add('loginViaAPI', (

*email* = Cypress.env('userEmail'),

*password* = Cypress.env('userPassword')

 ) => {

   cy.request('POST', `${Cypress.env('apiUrl')}/users/login`, {

     username: *email*,

     password,

   }).then((*response*) => {

     cy.setCookie('sessionId', *response*.body.sessionId)

     cy.setCookie('userId', *response*.body.userId)

     cy.setCookie('userName', *response*.body.userName)

     cy.visit('/#!/main')

   })

})

1.

Cypress.Commands.add('loginViaAPI', (

Kode berfungsi untuk menambahkan fungsi command dengan nama loginViaAPI di file cypress/support/commands.js

2.

*email* = Cypress.env('userEmail'),

*password* = Cypress.env('userPassword')

 ) => {

Kode berikut mempunyai fungsi untuk menambahkan object option dengan nama email dan password

3.

   cy.request('POST', `${Cypress.env('apiUrl')}/users/login`, {

Kode berikut merupakan parent commands yang. Akan menjadi awalan chain code.

* cy.request 🡪 berfungsi untuk membuat HTTP Request
* POST 🡪 merupakan method API yang akan kita gunakan pada postman
* `${Cypress.env('apiUrl')}/users/login 🡪 merupakan value dari url

4.

username: *email*,

     password,

Kode ini merupakan body dari lanjutan cy.request yang berisikan argument yang akan digunakan pada url

5.

}).then((*response*) => {

     cy.setCookie('sessionId', *response*.body.sessionId)

     cy.setCookie('userId', *response*.body.userId)

     cy.setCookie('userName', *response*.body.userName)

     cy.visit('/#!/main')

Kode ini merupakan logic then berupa respon jika request berhasili maka akan melakukan fungsinya yang berupa:  
- cy.setCookie 🡪 merupakan command untuk melakukan aturan pada cookie browser.

-cy.visit 🡪 merupakan command untuk mengunjungi remote url yang akan disebut.