

MODUL

AUTOMATION UI



**Created by Jago QA Institute
2021**

Apa itu Selenium?

Selenium adalah **tools yang digunakan untuk melakukan otomasi pada web browser**. Selenium bersifat open source sehingga penggunaannya gratis.

Otomasi kaya gimana?

Otomasi atau otomatisasi adalah **penggantian tenaga manusia dengan tenaga mesin yang otomatis** melakukan pekerjaan tersebut.

Contoh :

Ketika kamu sebagai QA sedang mengetes fitur Login yang ada di web browser secara manual, kamu akan menggunakan kedua tangan untuk mengecek test case satu persatu. Namun dengan selenium, pekerjaan kita terbantu dengan adanya pengecekan fitur Login yang berjalan otomatis, *web browser akan bergerak sendiri mengikuti perintah dari kodingan automation*.

Bahasa pemrograman yang mendukung Selenium :

Java, Python, C#, Perl, Javascript, Ruby, PHP

Sebelum kita memasuki tutorial penggunaan Selenium, berikut **daftar software** yang harus sudah di install oleh kalian:

- Python (minimal versi 3)
- Install PIP dari file **get-pip.py** yang bertugas sebagai downloader
- Install library selenium -> **pip install selenium**
- Install library webdriver (min. python versi 3.6) -> **pip install webdriver_manager**
- Code Editor (**Visual Studio Code/Sublime/Notepad++** dsb)
- Chromedriver atau mozilla driver

Lanjut ke Praktik :

Sekarang buatlah folder untuk proyek selenium yang akan kita jalankan. Contoh nama folder **belajar_selenium**

```
[mbp@MBPs-MacBook-Pro ~ % mkdir belajar.selenium  
[mbp@MBPs-MacBook-Pro ~ % cd belajar.selenium  
mbp@MBPs-MacBook-Pro belajar.selenium % ]
```

Masuk ke folder **belajar_selenium**. Buatlah file berformat *.py untuk python. Contoh nama file **belajar_selenium.py**

Buka Vs.Code/Sublime/Notepad++. Copy paste script dibawah ini ke file **belajar_selenium.py**

```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
import time

driver = webdriver.Chrome(ChromeDriverManager().install()) #deklarasi library chrome
driver.get('https://barru.pythonanywhere.com/daftar') #url web yang dituju
time.sleep(5) #delay 5 detik
driver.close() #menutup chrome web browser
```

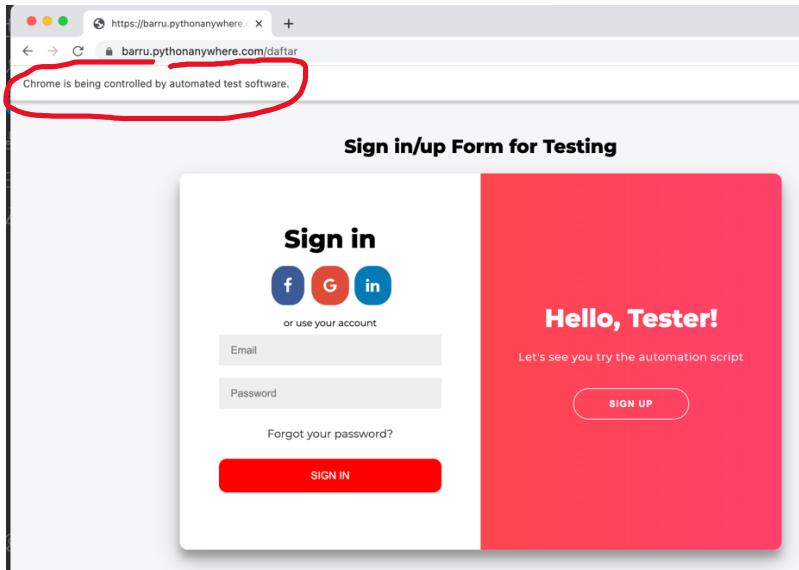
Running file script diatas dengan perintah

python belajar_selenium.py

```
[venv] mbp@MBPs-MacBook-Pro belajar_selenium % python belajar_selenium.py

===== WebDriver manager =====
Current google-chrome version is 96.0.4664
Get LATEST driver version for 96.0.4664
Driver [/Users/mbp/.wdm/drivers/chromedriver/mac64/96.0.4664.45/chromedriver] found in cache
[venv] mbp@MBPs-MacBook-Pro belajar_selenium %
```

Browser akan berjalan secara otomatis. Membuka url *barru.pythonanywhere.com/daftar* dan menutupnya kembali setelah 5 detik.



Selamat.. Anda sudah melakukan dasar automation UI dari website!

Mari ke tahap selanjutnya

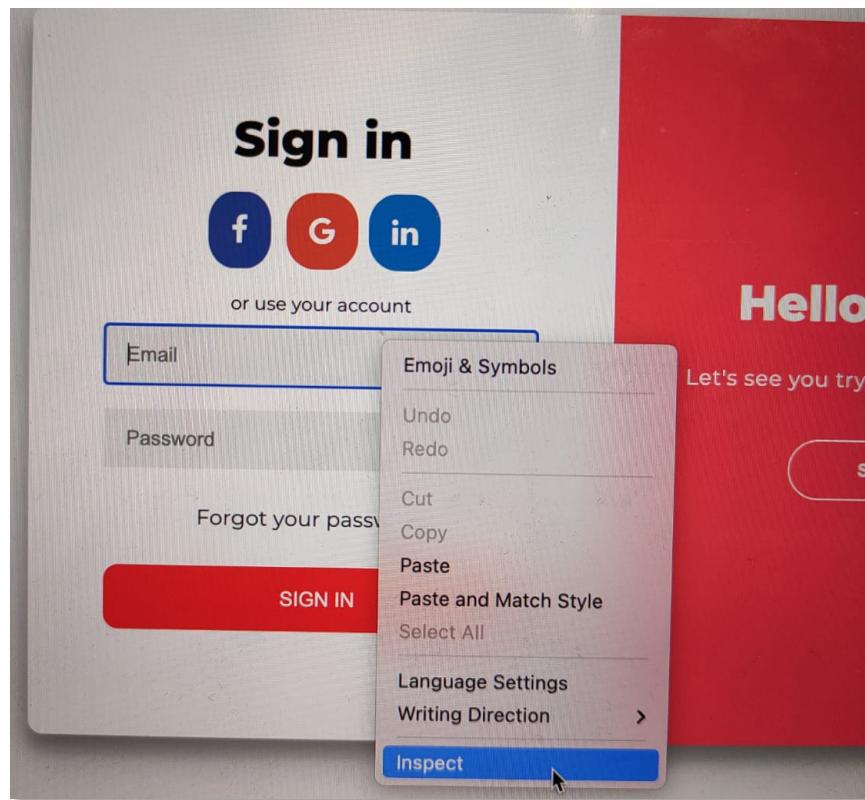
Mencari Elemen Web Target

Tahap selanjutnya adalah memprogram Selenium untuk mencari elemen yang ingin di interaksikan.

Mencari Elemen? Maksudnya?

Maksudnya adalah kita ambil **elemen/identitas** yang ingin kita uji, supaya tepat sasaran. Contoh, Ketika ingin mengisi form email, maka kita akan klik tampilan row bertuliskan email. Kita tahu bahwa row tersebut adalah tempat mengisi email. Karena terdapat petunjuk bertuliskan **Email** di row.

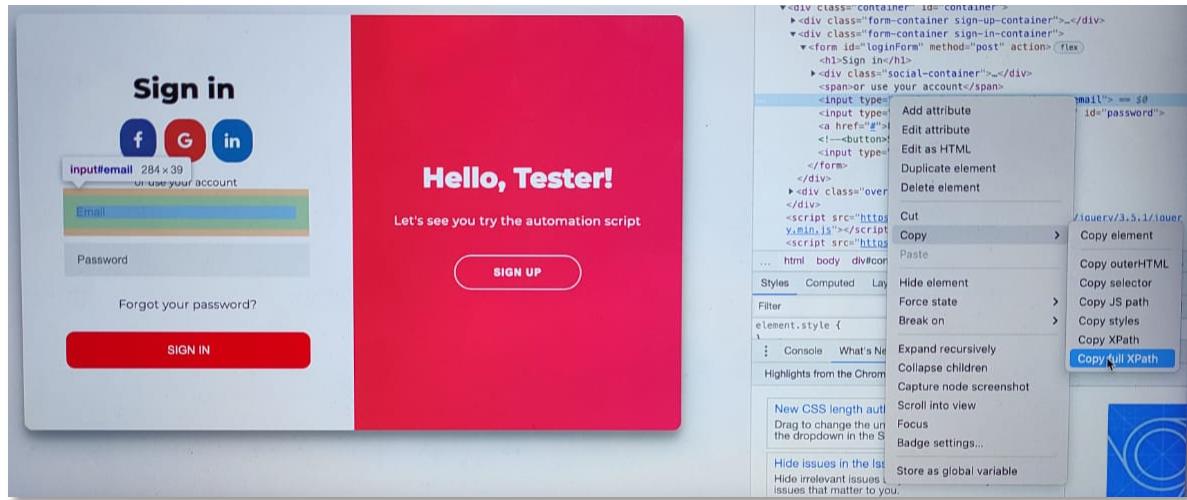
Maka identitas tersebut harus dikenali juga kodingan kita. Caranya yaitu klik kanan row email, pilih **inspect/inspect element**.



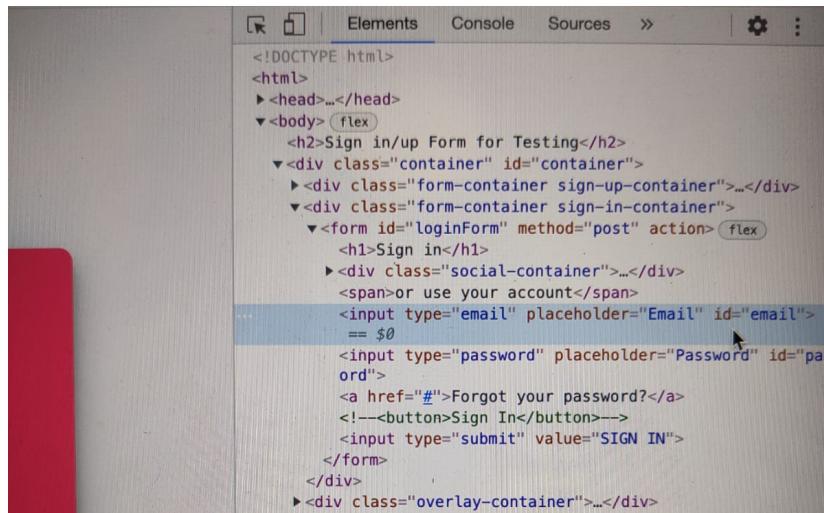
Klik **inspect**, maka akan muncul koding/script disebelah kanan browser yang relate dengan identitas yang kita pilih.

Setelah inspect, kita akan pilih elemen yang akan kita ambil. Berikut 2 elemen yang umum digunakan beserta cara ambil value nya :

1. Cara untuk mendapatkan elemen **Xpath**. Klik kanan script yang relate tersebut. Pilih **Copy Full Xpath**.



2. Cara untuk mendapatkan elemen **id**. Ambil value pada id. Contoh, value nya adalah **email** dari **id** pada gambar dibawah ini.



List elemen yang bisa digunakan :

```
find_element_by_id
find_element_by_name
find_element_by_xpath
find_element_by_link_text
find_element_by_partial_link_text
find_element_by_tag_name
find_element_by_class_name
find_element_by_css_selector
```

Mari kita uji fitur Login. Melakukan automation script untuk UI, maka kita harus tau test step dari test case yang sudah disiapkan. Karena automation script yang disusun adalah otomatisasi dari test step. Contoh :

Test Case :

Verify Success Login with Valid Email and Password

Test Step :

1. Buka browser
2. Klik dan search <https://barru.pythonanywhere.com/daftar> di address bar
3. Klik dan masukkan email di row email
4. Klik dan masukkan password di row password
5. Klik tombol Sign In

Setelah kita ketahui test step dari test case diatas, ikuti langkah nya..

- Buka url <https://barru.pythonanywhere.com/daftar>
- **inspect** tiap row **email, password**, tombol **Sign In**, ambil value **id** atau **copy full xpath**.
- Simpan elemen **xpath** taruh pada query **find_element_by_xpath**. Elemen **id** taruh pada query **find_element_by_id**.

```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
import time

#deklarasi library chrome agar browser berjalan otomatis
driver = webdriver.Chrome(ChromeDriverManager().install())
driver.get('https://barru.pythonanywhere.com/daftar') #url web yang dituju
time.sleep(2)
driver.find_element_by_id("email").send_keys("barru.kurniawan@gmail.com")
time.sleep(2)
driver.find_element_by_id("password").send_keys("sman60jakarta")
time.sleep(2)
driver.find_element_by_xpath("/html/body/div/div[2]/form/input[3]").click()
time.sleep(2)
driver.close() #menutup chrome web browser
```

Lalu running script diatas

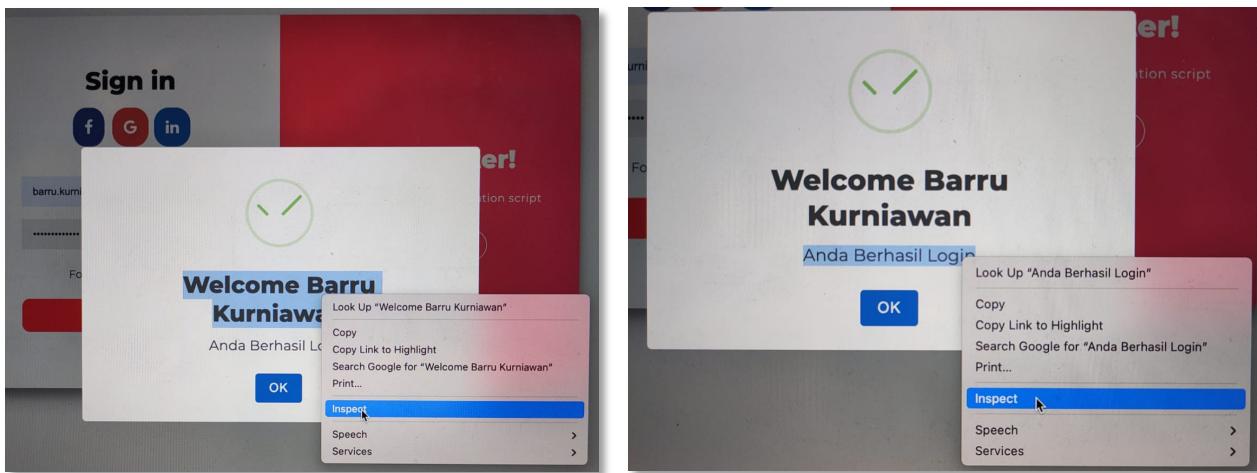
python belajar_selenium.py

```
(venv) mbp@MBPs-MacBook-Pro belajar_selenium % python belajar_selenium.py

===== WebDriver manager =====
Current google-chrome version is 96.0.4664
Get LATEST driver version for 96.0.4664
Driver [/Users/mpb/.wdm/drivers/chromedriver/mac64/96.0.4664.45/chromedriver] found in cache
(venv) mbp@MBPs-MacBook-Pro belajar_selenium %
```

Browser akan berjalan secara otomatis. Membuka url *barru.pythonanywhere.com/daftar*, melakukan login dan menutupnya kembali setelah 2 detik.

Setelah berhasil menjalankan script automation diatas, mari lanjut ke Assertion atau validasi. Sekarang ambil elemen dari hasil respon login. Ambil elemen **xpath** dari Welcome Barru Kurniawan dan Anda Berhasil Login.



```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
import time

#deklarasi library chrome agar browser berjalan otomatis
driver = webdriver.Chrome(ChromeDriverManager().install())
driver.get('https://barru.pythonanywhere.com/daftar') #url web yang dituju
time.sleep(2)
driver.find_element_by_id("email").send_keys("barru.kurniawan@gmail.com")
time.sleep(2)
driver.find_element_by_id("password").send_keys("sman60jakarta")
time.sleep(2)
driver.find_element_by_xpath("//html/body/div/div[2]/form/input[3]").click()
time.sleep(3)

response_message =
driver.find_element_by_xpath("//html/body/div/div[2]/div/div[1]/h2").text
response_data =
driver.find_element_by_xpath("//html/body/div/div[2]/div/div[2]/div[1]").text

assert "Welcome Barru Kurniawan" in response_message
assert "Anda Berhasil Login" in response_data

driver.close() #menutup chrome web browser
```

```
[(venv) mbp@MBPs-MacBook-Pro belajar_selenium % python belajar_selenium.py

===== WebDriver manager =====
Current google-chrome version is 96.0.4664
Get LATEST driver version for 96.0.4664
Driver [/Users/mbp/.wdm/drivers/chromedriver/mac64/96.0.4664.45/chromedriver] found in cache
(venv) mbp@MBPs-MacBook-Pro belajar_selenium %
```

```

import unittest
import time
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager

class TestLogin(unittest.TestCase):

    def setUp(self):
        self.driver = webdriver.Chrome(ChromeDriverManager().install())

    def test_a_success_login(self):
        driver = self.driver

        driver.get("http://barru.pythonanywhere.com/daftar")
        time.sleep(1)
        driver.find_element_by_xpath("/html/body/div/div[2]/form/input[1]").send_keys('jagoqaindonesia@gmail.com')
        time.sleep(1)
        driver.find_element_by_id("password").send_keys('sman60jakarta')
        time.sleep(1)
        driver.find_element_by_xpath('/html/body/div/div[2]/form/input[3]').click()
        time.sleep(2)

        respon_welcome = driver.find_element_by_xpath('/html/body/div[2]/div/div[1]/h2').text
        respon_berhasil = driver.find_element_by_xpath('/html/body/div[2]/div/div[2]/div[1]').text

        self.assertEqual(respon_welcome, 'Welcome Jago QA')
        self.assertEqual(respon_berhasil, 'Anda Berhasil Login')

    def test_b_failed_login_email_not_registered(self):
        driver = self.driver

        driver.get("http://barru.pythonanywhere.com/daftar")
        time.sleep(1)
        driver.find_element_by_xpath("/html/body/div/div[2]/form/input[1]").send_keys('tester.ganteng@jumawa.com')
        time.sleep(1)
        driver.find_element_by_id("password").send_keys('sman60jakarta')
        time.sleep(1)
        driver.find_element_by_xpath('/html/body/div/div[2]/form/input[3]').click()
        time.sleep(2)

        respon_welcome = driver.find_element_by_xpath('/html/body/div[2]/div/div[1]/h2').text
        respon_berhasil = driver.find_element_by_xpath('/html/body/div[2]/div/div[2]/div[1]').text

        self.assertIn('not found',respon_welcome)
        self.assertIn('Salah', respon_berhasil)
        self.assertEqual(respon_welcome, "User's not found")
        self.assertEqual(respon_berhasil, 'Email atau Password Anda Salah')

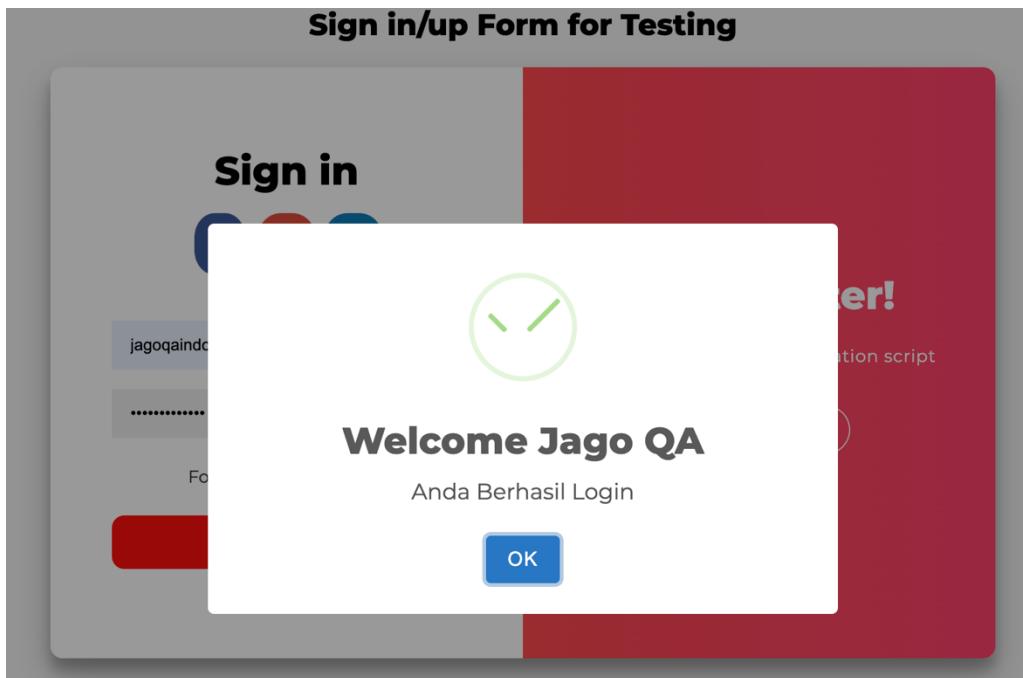
    def tearDown(self):
        self.driver.close()

if __name__ == "__main__":
    unittest.main()

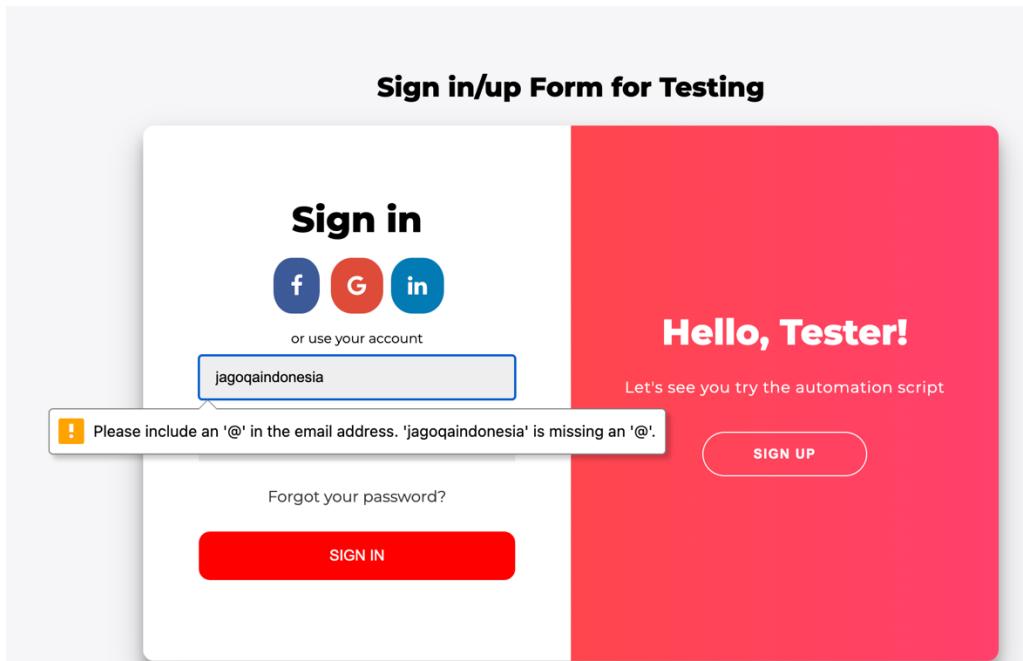
```

DOCUMENTATION

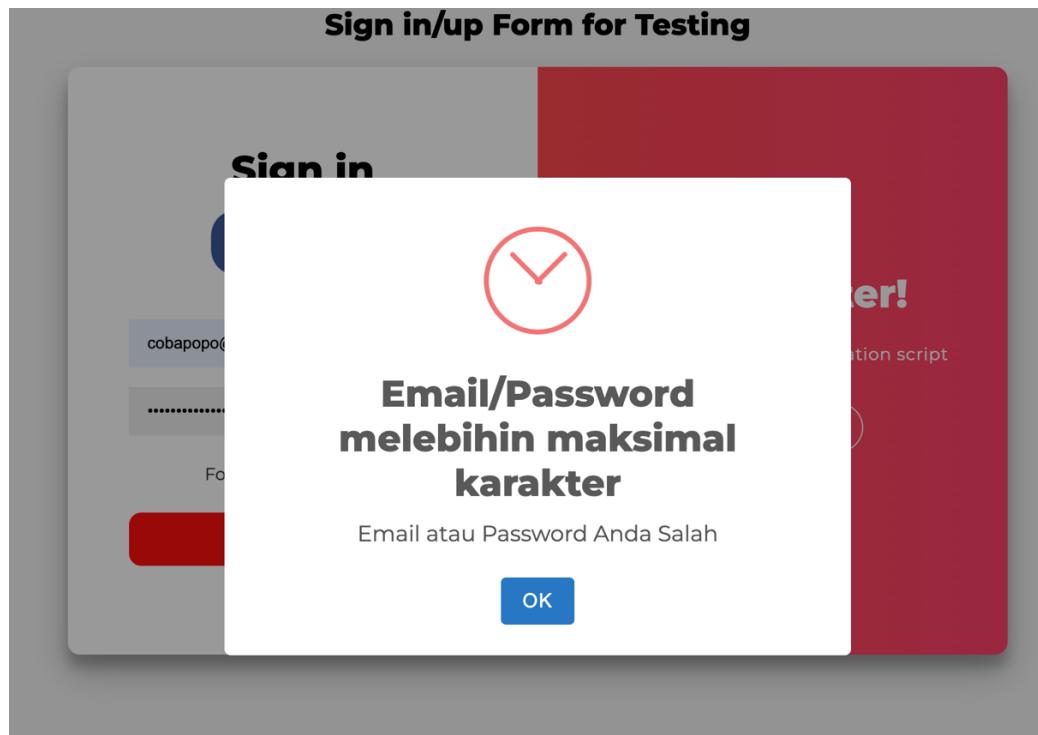
LOGIN SUCCESS



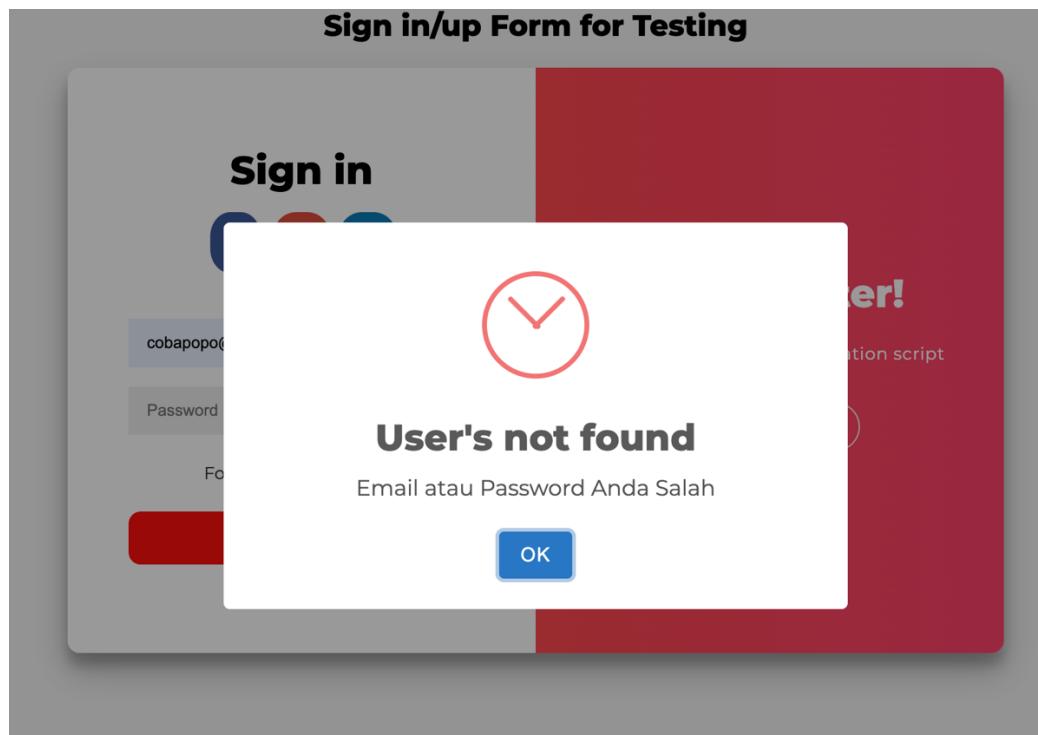
LOGIN FAILED (NOT VALID EMAIL)



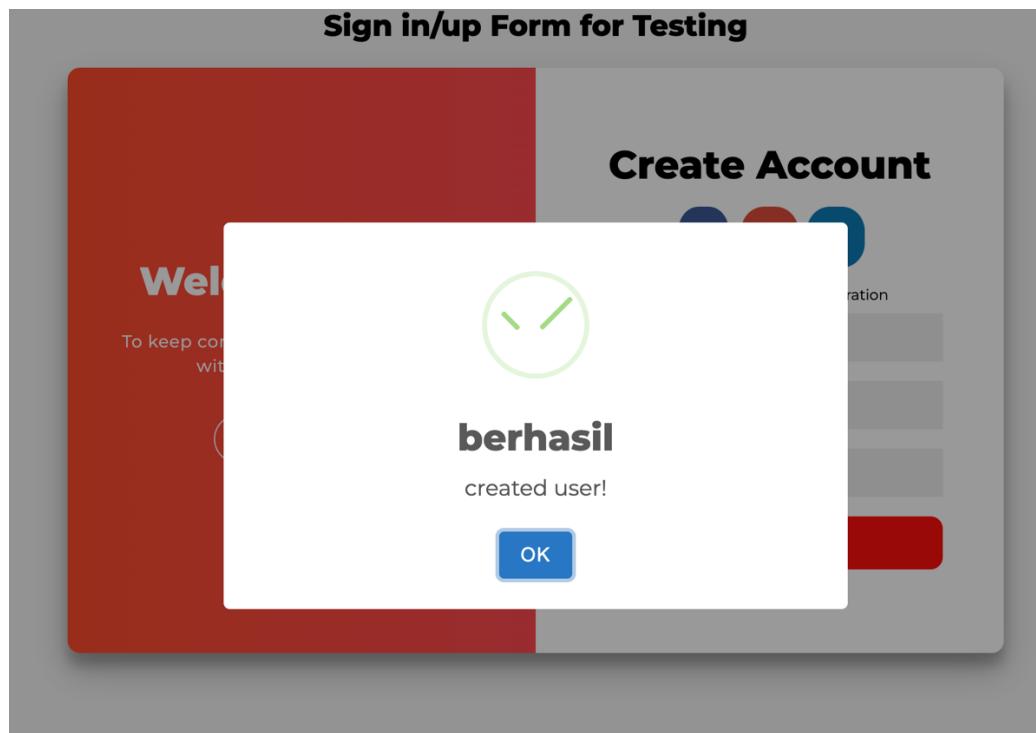
LOGIN FAILED (EMAIL/PASSWORD MAX CHARACTERS)



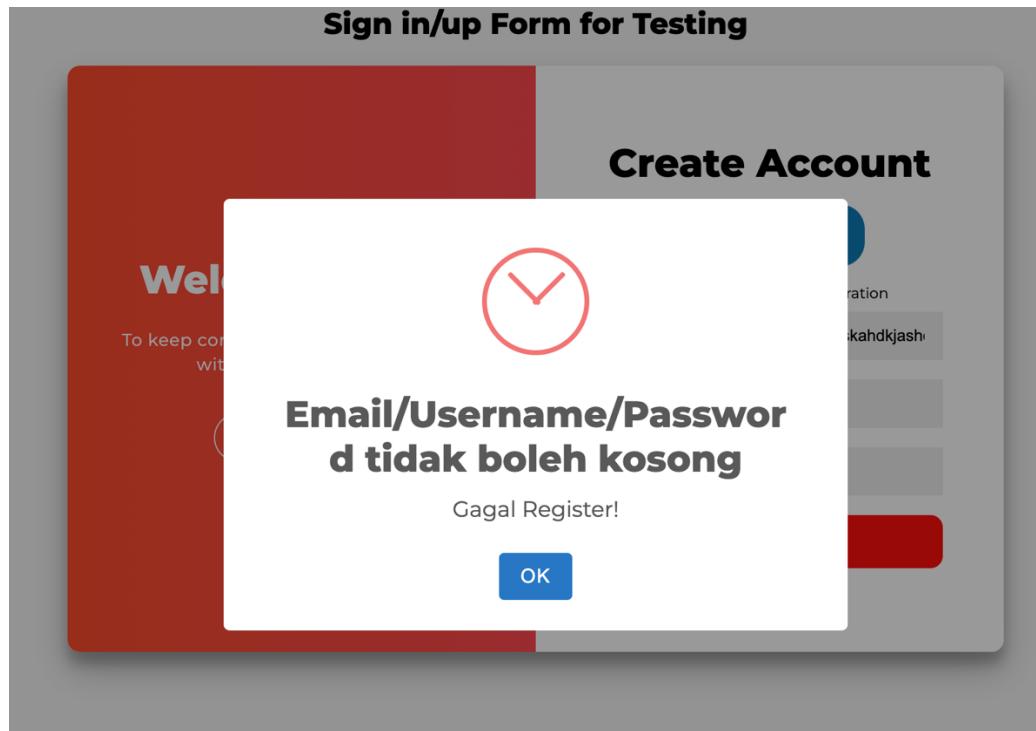
LOGIN FAILED (EMAIL/PASSWORD EMPTY)



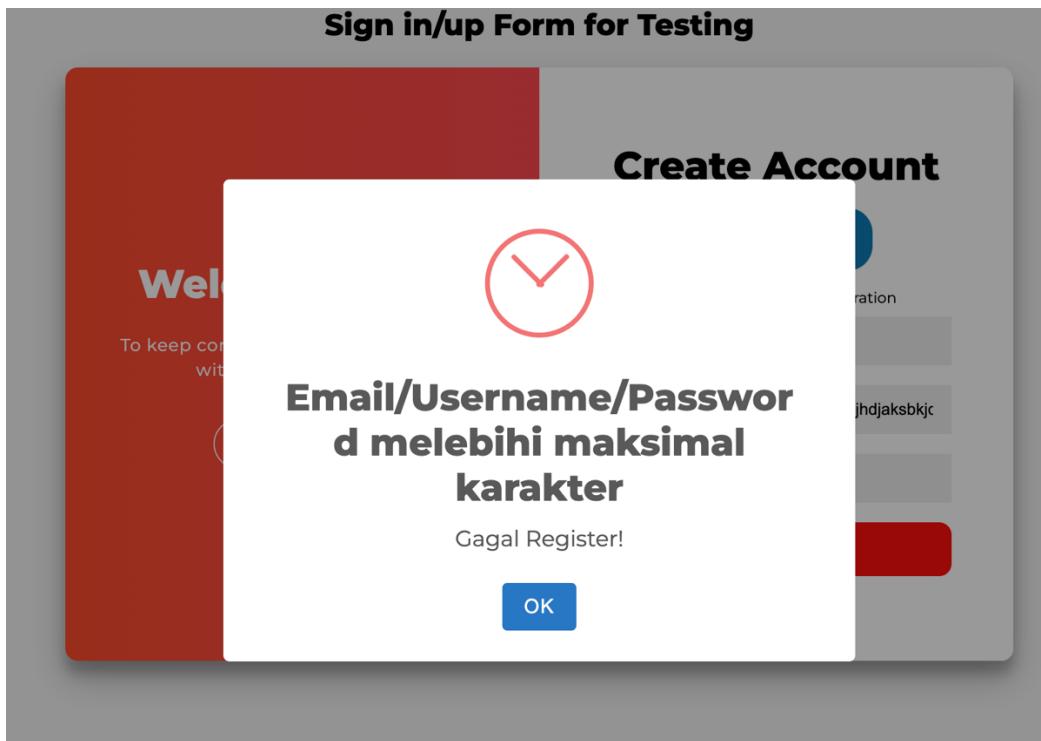
REGISTER SUCCESS



REGISTER FAILED (EMAIL/PASSWORD/NAME EMPTY)



REGISTER FAILED (EMAIL/PASSWORD/NAME MELEBIHI KARAKTER)



REGISTER FAILED (EMAIL/PASSWORD EMPTY)

