**LAPORAN TESTING WEB TWITER**

**MENGGUNAKAN SELERON**

**Dosen : Iwan Lesmana, S.Kom., M.kom.**



Disusun oleh:

NAMA : Handy Nanda Fachrizal

NIM : 20220810002

KELAS : TINFC-2022-03

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS KUNINGAN**

**2023**

**Daftar Isi**

[**Pendahuluan** 3](#_Toc202555910)

[**Deskripsi Aplikasi** 3](#_Toc202555911)

[**Pemilihan Tool Automation** 3](#_Toc202555912)

[**Skenario Pengujian** 4](#_Toc202555913)

[**Implementasi Test (Cuplikan Script)** 4](#_Toc202555914)

[**Hasil Pengujian** 5](#_Toc202555915)

[**Permasalahan & Solusi** 6](#_Toc202555916)

[**Kesimpulan** 6](#_Toc202555917)

[**Link Video YouTube** 6](#_Toc202555918)

# **Pendahuluan**

Pengujian perangkat lunak merupakan tahapan penting dalam proses pengembangan sistem untuk memastikan bahwa aplikasi bekerja sesuai dengan fungsionalitas yang diharapkan dan memiliki tingkat keandalan yang tinggi. Salah satu pendekatan yang digunakan dalam proses ini adalah **pengujian otomatis (automation testing)**, yaitu metode pengujian yang dijalankan secara otomatis dengan memanfaatkan alat bantu tertentu.

Dalam laporan ini, dilakukan pengujian otomatis terhadap beberapa fitur utama pada aplikasi web **Twitter (X.com)** menggunakan tools **Selenium WebDriver**. Selenium dipilih karena kemampuannya dalam mengotomatisasi interaksi dengan elemen-elemen antarmuka pengguna (UI) pada browser secara real-time. Fitur-fitur yang diuji mencakup proses **login**, **pencarian konten**, dan **logout**.

Tujuan dari pengujian ini adalah untuk memastikan bahwa fitur-fitur tersebut dapat dijalankan dengan baik dan sesuai dengan ekspektasi pengguna. Selain itu, melalui automation testing, pengujian dapat dilakukan dengan lebih cepat, konsisten, dan efisien dibandingkan pengujian manual.

# **Deskripsi Aplikasi**

Aplikasi yang diuji adalah Twitter/X versi desktop [("https://x.com/login")](https://x.com/login) Twitter/X merupakan platform media sosial yang memungkinkan pengguna untuk membagikan teks, gambar, video, serta melakukan pencarian dan interaksi sosial. Pengujian ini difokuskan pada:

* Login dengan kredensial benar dan salah
* Autentikasi tambahan (username)
* Pencarian konten ("android developer")
* Logout dari aplikasi

# **Pemilihan Tool Automation**

Tool yang digunakan adalah Selenium WebDriver dengan bahasa pemrograman Python. Alasan pemilihannya:

* Gratis dan open-source
* Mendukung berbagai browser (Chrome, Firefox, dll)
* Cocok untuk pengujian berbasis antarmuka (GUI)
* Kompatibel dengan banyak bahasa pemrograman
* Dapat mengontrol interaksi web secara dinamis dan kompleks

# **Skenario Pengujian**

1. Skenario 1**:** Login gagal dengan password yang salah
2. Skenario 2**:** Login berhasil dengan email dan password benar (termasuk autentikasi username)
3. Skenario2b**:** Melakukan pencarian "android developer" di kolom search
4. Skenario 3**:** Logout dari akun Twitter

# **Implementasi Test (Cuplikan Script)**

        username\_input = WebDriverWait(driver, 20).until(

            EC.presence\_of\_element\_located((By.XPATH, "//input[@name='text']"))

        )

        print(f"INFO: Memasukkan username: {username}")

        username\_input.send\_keys(username)

        next\_button = WebDriverWait(driver, 10).until(

            EC.element\_to\_be\_clickable((By.XPATH, "//span[contains(text(), 'Next')]"))

        )

        next\_button.click()

        print("INFO: Mengklik tombol 'Next'.")

        password\_input = WebDriverWait(driver, 20).until(

            EC.presence\_of\_element\_located((By.XPATH, "//input[@name='password']"))

        )

        print("INFO: Memasukkan password...")

        password\_input.send\_keys(password)

        login\_button = WebDriverWait(driver, 10).until(

            EC.element\_to\_be\_clickable((By.XPATH, "//button[@data-testid='LoginForm\_Login\_Button']"))

        )

        login\_button.click()

        print("INFO: Mengklik tombol 'Log in'.")

        return True

    except TimeoutException as e:

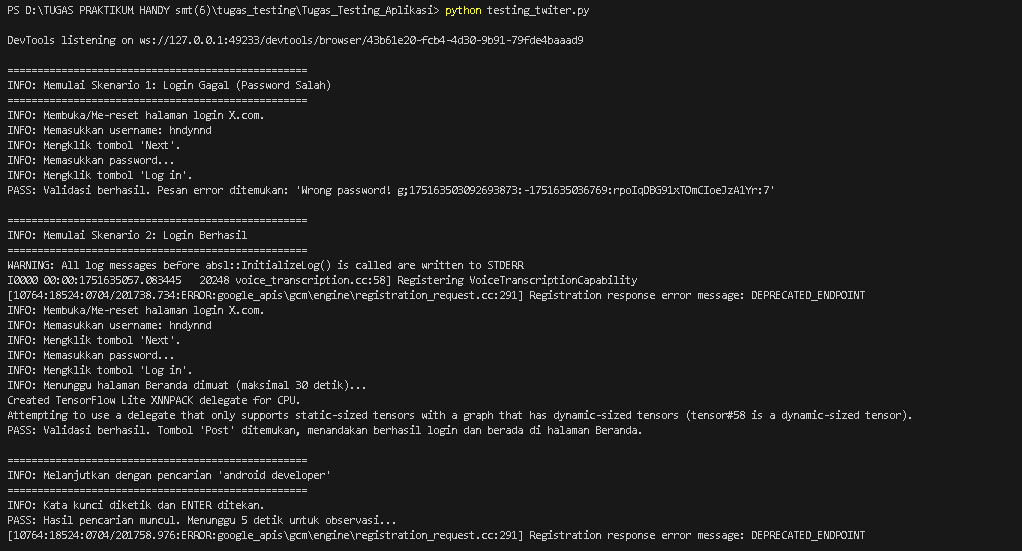
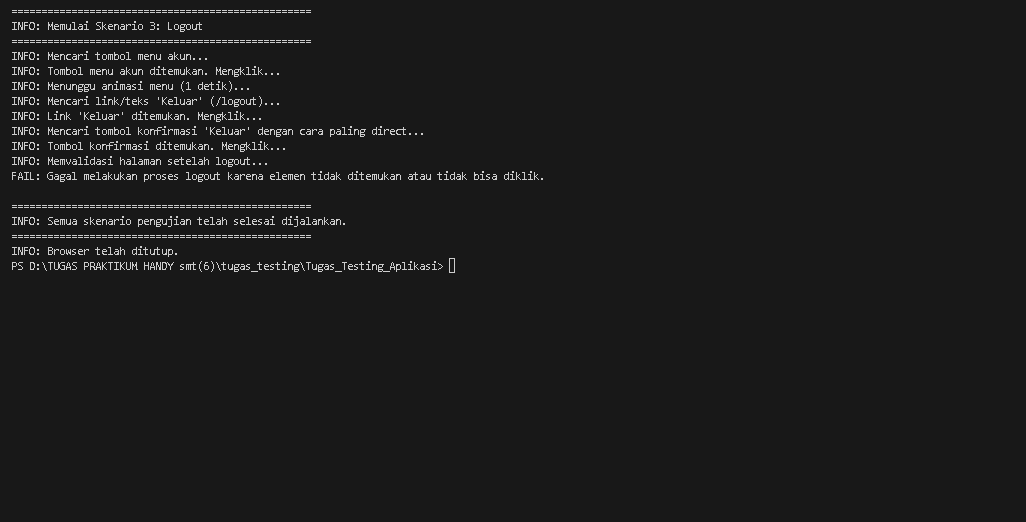
        print(f"FAIL: Elemen login tidak ditemukan atau tidak bisa diklik. Error: {e}")

        return False

# **Hasil Pengujian**

* Skenario 1**:** Gagal login berhasil dideteksi dan menampilkan pesan error
* Skenario 2**:** Login berhasil setelah input email, autentikasi username, dan password
* Skenario 2b**:** Kolom pencarian ditemukan, dan hasil pencarian "android developer" muncul
* Skenario 3**:** Logout berhasil dan pengguna dikembalikan ke halaman login

Log hasil print menunjukkan semua proses berhasil dilalui dengan PASS.



# **Permasalahan & Solusi**

|  |  |
| --- | --- |
| **Permasalahan** | **Solusi** |
| Autentikasi tambahan muncul setelah login | Ditambahkan input username kedua sebelum password |
| Lokasi tombol dan input sering berubah | XPath dinamis digunakan, dan menunggu (WebDriverWait) ditambahkan |
| TimeoutException saat hasil tidak muncul | Timeout ditambah hingga 20–30 detik, dan ditambahkan validasi hasil |

# **Kesimpulan**

Berdasarkan hasil pengujian otomatis menggunakan **Selenium WebDriver** terhadap aplikasi **Twitter (X.com) versi desktop**, dapat disimpulkan bahwa semua skenario yang dirancang—mulai dari login, pencarian konten, hingga logout—berjalan dengan baik. Pengujian ini juga berhasil menangani proses autentikasi tambahan yang muncul setelah pengguna memasukkan alamat email.

Penerapan automation testing terbukti efektif dalam mempercepat proses pengujian, khususnya untuk pengujian regresi yang berulang. Selain meningkatkan efisiensi waktu, metode ini juga mampu meminimalkan kesalahan manusia yang kerap terjadi pada pengujian manual.

Dengan keberhasilan skenario yang diuji, automation testing ini berpotensi untuk diintegrasikan ke dalam **CI/CD pipeline** guna mendukung proses pengembangan dan rilis aplikasi yang lebih cepat dan andal di masa mendatang.

# **Link Video YouTube**

<https://youtu.be/ZvaKmIAlM78>