BAB IV
IMPLEMENTASI DAN UJICOBA

4.1 Implementasi

Bagian ini menjelaskan tentang implementasi rancangan yang telah dideskripsikan pada bab sebelumnya. Bagian ini terdiri dari spesifikasi produk, implementasi database dan implementasi program.

4.1.4 Spesifikasi Produk

Pada bagian spesifikasi produk berisi tentang penjelasan tentang spesifikai produk yang dihasilkan, kebutuhan hardware dan software untuk penerapan sistem pada lingkungan yang sebenarnya.

1. Spesifikasi Produk yang Dihasilkan:

Produk yang dihasilkan adalah sebuah sistem informasi aktivitas laboratorium riset berbasis web yang memungkinkan manajemen proyek riset, kolaborasi antarpeneliti, pengelolaan data eksperimental, dan peminjaman peralatan laboratorium. Fitur utama dari sistem ini meliputi:

1. **Manajemen Proyek Riset:** Kemampuan untuk membuat, mengelola, dan melacak proyek riset.
2. **Kolaborasi:** Fasilitas untuk kolaborasi antarpeneliti dan mahasiswa melalui sistem pesan dan berbagi data.
3. **Pengelolaan Data Eksperimental:** Penyimpanan dan pengelolaan data sampel serta hasil eksperimentasi.

4. **Peminjaman Peralatan:** Sistem peminjaman dan pengembalian peralatan laboratorium.

Dashboard Pengguna: Antarmuka yang memudahkan pengguna dalam mengakses informasi proyek dan kolaborasi.

4.1.5 Implementasi Database

Pada bagian Implementasi database menjelaskan tentang implementasi rancangan database yang telah di gambarkan pada bab sebelumnya ke dalam DBMS yang dipilih serta hasil implementasi, seperti penamaan database dan tabel yang terbentuk.

1. Pemilihan DBMS:

DBMS yang dipilih untuk sistem ini adalah MongoDB, sebuah sistem basis data NoSQL yang fleksibel dan skalabel, cocok untuk menyimpan data dengan struktur yang kompleks dan dinamis.

2. Penamaan Database:

Database untuk sistem informasi aktivitas laboratorium riset diberi nama lab_research_system.

3. Struktur Tabel yang Terbentuk:

Implementasi database di MongoDB menghasilkan beberapa koleksi (collections) sebagai berikut:

1. **borrowed_equipment**: Menyimpan informasi tentang peminjaman peralatan.
2. **borrowed_requests**: Mencatat permintaan peminjaman peralatan oleh pengguna.

3. **equipment**: Mencatat informasi tentang peralatan yang tersedia.
4. **messages**: Menyimpan pesan terkait dengan proyek.
5. **notification**: Mencatat notifikasi untuk pengguna.
6. **project**: Menyimpan data tentang proyek riset.
7. **sample**: Mencatat informasi tentang sampel dokumen yang diunggah untuk proyek.
8. **statistik**: Mencatat statistik mengenai jumlah proyek per tahun.
9. **user**: Menyimpan informasi tentang pengguna sistem.

Untuk menghubungkan Python ke MongoDB menggunakan Visual Studio Code (VSCode) tanpa menginstal MongoDB secara lokal, Anda dapat mengikuti langkah-langkah berikut:

1. Buat Project di VSCode

1. Buka VSCode dan buat folder baru untuk project Python Anda.
2. Buka terminal di VSCode (View > Terminal) dan pastikan Anda berada di folder project tersebut.

2. Buat Virtual Environment (Opsional)

Membuat virtual environment akan membantu Anda mengelola dependensi Python secara terpisah untuk project ini:

python -m venv venv

Aktifkan virtual environment:

Windows:

.\venv\Scripts\activate

macOS/Linux:

source venv/bin/activate

3. Instal pymongo

pymongo adalah driver MongoDB untuk Python yang akan Anda gunakan untuk menghubungkan dan berinteraksi dengan database MongoDB.

pip install pymongo

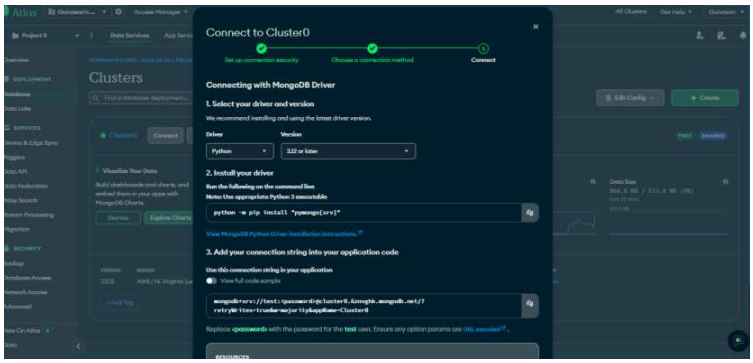
```
(venv) PS C:\Users\Gunawan Krisna\Downloads\Compressed\web_lab-main> pip install pymongo
Collecting pymongo
  Obtaining dependency information for pymongo from https://files.pythonhosted.org/packages/51
/28/577224211f43e2079126bfec53080efba46e59218f47808098f125139558/pymongo-4.8.0-cp311-cp311-win
_amd64.whl.metadata
  Using cached pymongo-4.8.0-cp311-cp311-win_amd64.whl.metadata (22 kB)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in c:\users\gunawan krisna\downloads\comp
ressed\web_lab-main\venv\lib\site-packages (from pymongo) (2.6.1)
Using cached pymongo-4.8.0-cp311-cp311-win_amd64.whl (630 kB)
Installing collected packages: pymongo
Successfully installed pymongo-4.8.0

[notice] A new release of pip is available: 23.2.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Gambar 4. 1 install library pymongo

4. Mendapatkan URI Koneksi MongoDB

1. Masuk ke MongoDB Atlas atau server MongoDB Anda.
2. Pilih Database yang ingin Anda hubungkan.
3. Klik "Connect" di samping nama cluster Anda.
4. Pilih "Drivers" sebagai metode koneksi.
5. Salin URI yang terdapat di bawah "Add your connection string into your application



Gambar 4. 2 Connection Mongodb

5. Membuat Script Python untuk Menghubungkan ke MongoDB

Buat file Python baru, misalnya mongo_connect.py, dan tulis kode berikut untuk menghubungkan ke MongoDB:

```
from pymongo import MongoClient

client = MongoClient(
    "mongodb://test:test@ac-owhiyhy-shard-00-00.6znvghk.mongodb.net:27017,ac-owl
)
db = client["Cluster0"]
users_collection = db["user"]
```

4.1.6 Implementasi Program

Pada bagian implementasi program menjelaskan tentang tahap-tahap yang harus dilakukan dalam membuat program disertai dengan layout program (tanpa inputan) sesuai dengan rancangan pada bab sebelumnya dan potongan program/script yang dibutuhkan (segmen program). Segmen program merupakan sebagian dari program yang merupakan fungsi utama sesuai yang dijelaskan pada masing-masing bagian.

1. Instal Library yang Diperlukan

```
pip install Flask
pip install pymongo
pip install bcrypt
pip install bson
pip install python-dotenv
pip install pytz
pip install sendinblue-api-v3-sdk
```

Segmen Program 4. 1 instal library yang diperlukan

2. Implementasi Rute-rute Utama

Rute /login

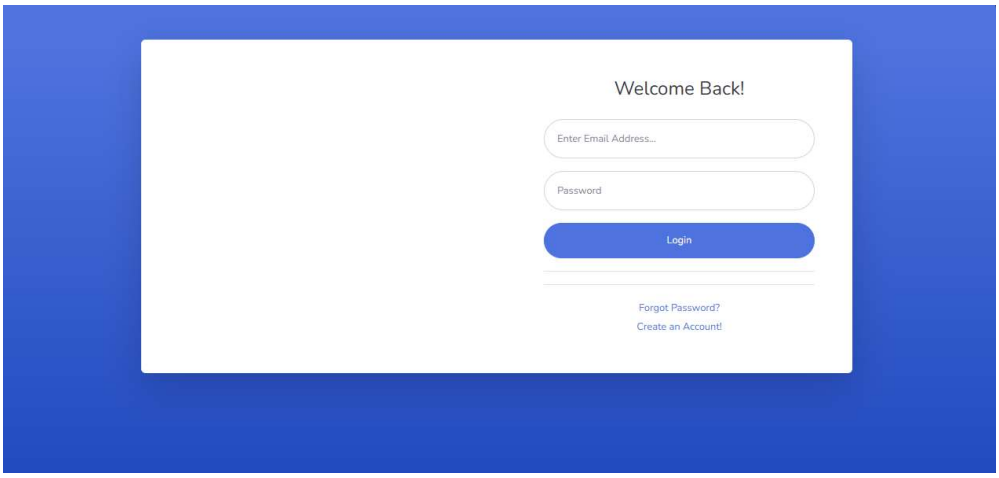
```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password'].encode('utf-8')

        user_data = users_collection.find_one({'email': email})

        if user_data and bcrypt.checkpw(password,
user_data['password']):
            session['user'] = {
                '_id': str(user_data['_id']),
                'name': user_data['name'],
                'email': user_data['email'],
                'role': user_data['role']
            }
            return redirect(url_for('home'))
        else:
            return render_template('login.html', error='Invalid
email or password')

    return render_template('login.html')
```

Segmen Program 4. 2 Rute Login



Gambar 4. 3 Hasil Halaman Login

Rute /register

```python
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        password = request.form['password'].encode('utf-8')
        role = request.form['role']
        photo = request.files['photo'] if 'photo' in request.files else None
        default_bio = "This is a default bio."
        nidn = request.form['nidn']
        nrp = request.form['nrp']

        existing_user = users_collection.find_one(
            {'$or': [{'name': name}, {'email': email}]})

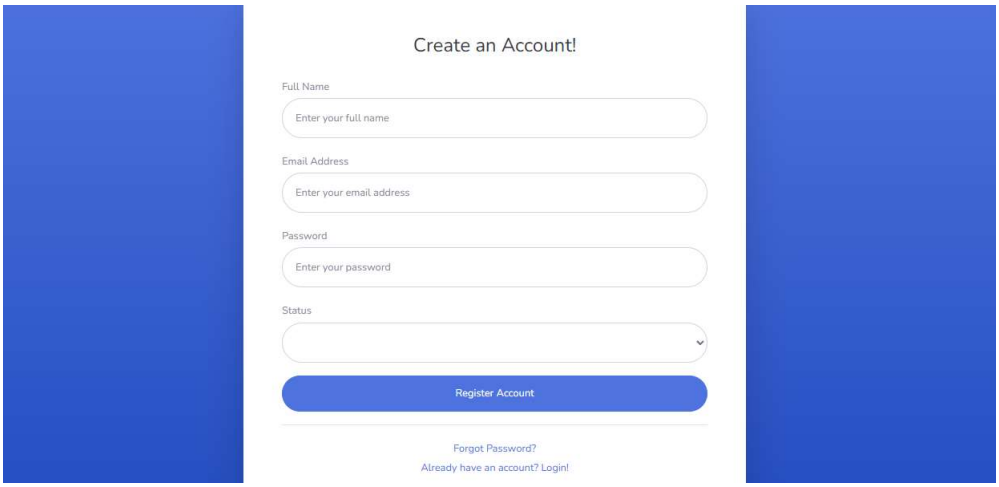
        if existing_user is None:
            hashed_password = bcrypt.hashpw(password, bcrypt.gensalt())

            # untu foto tapi masih tidak bisa
            photo_data = None
            if photo:
                photo_data = photo.read()

            # untuk insert user
            users_collection.insert_one({
                'name': name,
                'email': email,
                'password': hashed_password,
                'role': role,
                'photo': photo_data,  # masih tidak bisa
                'bio': default_bio,
                'nidn': nidn
            })

            if role == 'mahasiswa':
                # untuk insert mahasiswa
                mahasiswa_collection.insert_one({
                    'name': name,
                    'email': email,
                    'password': hashed_password,
                    'role': role,
                    'photo': photo_data,  # masih tidak bisa
```

Segmen Program 4. 3 Rute Register



Gambar 4. 4 Hasil Halaman Register

Rute /newproject

```
@app.route('/newproject', methods=['GET', 'POST'])
def newproject():
    if 'user' in session:
        user_data = session['user']
        user_name = user_data['name']
        user_email = user_data['email']

        if request.method == 'POST':
            project_name = request.form['project_name']
            project_description = request.form['project_description']
            tanggal_mulai = request.form['tanggal_mulai']
            tanggal_selesai = request.form['tanggal_selesai']
            user_email = request.form['user_email']

            existing_project = projects_collection.find_one(
                {'project_name': project_name})

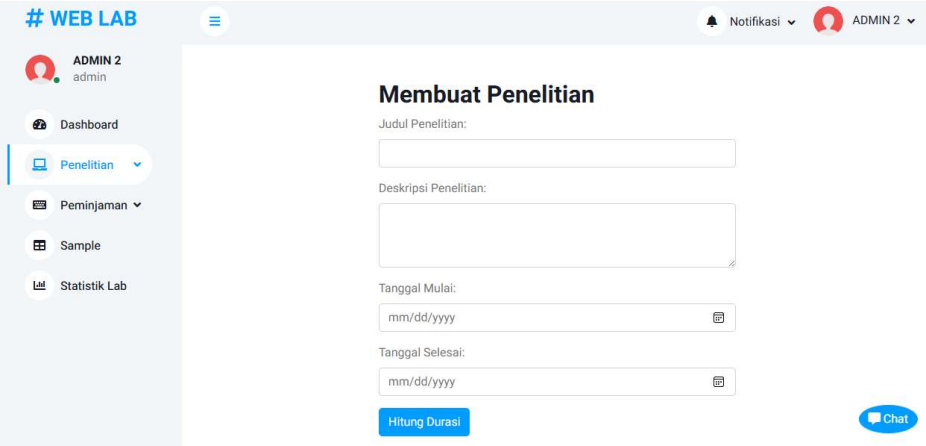
            if existing_project:
                flash(
                    'Project with this name already exists. Please choose a
different name.', 'danger')
            else:
                project_description = request.form['project_description']

            member_data = {
                'name': user_name,
                'email': user_email,
                'role': 'Kepala Peneliti'
            }

            # Insert project into the database
            project_id = projects_collection.insert_one({
                'project_name': project_name,
                'project_description': project_description,
                'member': [member_data],
                'project_creator': user_name,
                'email_creator': user_email,
                'tanggal_mulai': tanggal_mulai,
                'tanggal_selesai': tanggal_selesai
            }).inserted_id

            # Update user's projects
```

Segmen Program 4. 4 Rute New Project



Gambar 4. 5 Hasil Halaman New Project

Rute /borrow_equipment

```
@app.route('/borrow_equipment', methods=['GET', 'POST'])
def borrow_equipment_page():
    # Fetch equipment data
    equipment_data = list(equipment_collection.find({}, {'_id': 0}))

    # Fetch semua borrowed equipment data
    borrowed_equipment_data = list(borrowed_equipment_collection.find())

    if request.method == 'POST':
        # Form untuk pinjam alat
        equipment_name = request.form['equipment']
        requested_quantity = int(request.form['quantity'])
        user_id = ObjectId(session['user']['_id'])
        user_data = users_collection.find_one({'_id': user_id})
        user_name = user_data['name']

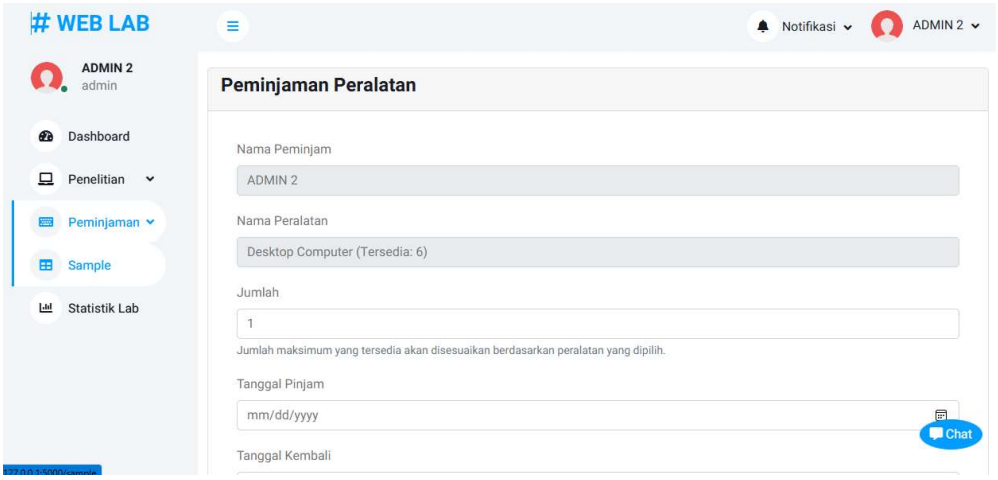
        # Fetch ketersediaan quantity dari collection equipment
        available_quantity_data = equipment_collection.find_one(
            {'name': equipment_name}, {'_id': 0, 'quantity': 1})
        available_quantity = available_quantity_data['quantity'] if
available_quantity_data else 0

        # mengecek apa quantity masih ada
        if requested_quantity > available_quantity:
            flash('Not enough quantity available for borrowing.', 'danger')
        else:
            # Update data quantity di equipment collection agar dapat turun
untuk quantity
            equipment_collection.update_one({'name': equipment_name}, {
                                            '$inc': {'quantity': -
requested_quantity}})

            # menyimpan peralatan yang dipinjam ke database
            borrowed_equipment_collection.insert_one({
                'user_id': user_id,
                'user_name': user_name,
                'equipment_name': equipment_name,
                'quantity': requested_quantity
            })

            flash('Borrow successful!', 'success')
```

Segmen Program 4. 5 Rute borrow_equipment



Gambar 4. 6 Hasil Halaman Peminjaman Peralatan

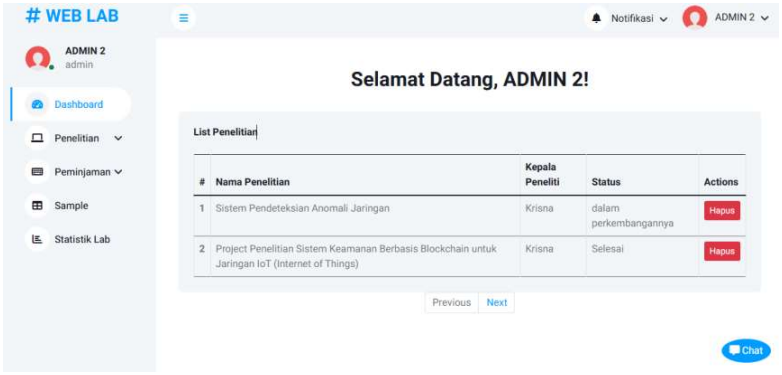
Rute /home

```
@app.route('/home')
def home():
    if 'user' in session:
        user_id = ObjectId(session['user']['_id'])
        user_data = users_collection.find_one({'_id': user_id})

        current_user_projects = session['user'].get('projects', [])
        # Fetch projects untuk user saat ini
        projects_data = list(projects_collection.find({}))

        projects_data = list(projects_collection.find(
            {}, {'_id': 1, 'project_name': 1, 'project_creator': 1,
'project_status': 1, 'email_creator': 1}))
        return render_template('home.html', user_data=user_data,
projects_data=projects_data,)
    else:
        return redirect(url_for('login'))
```

Segmen Program 4. 6 Rute Home

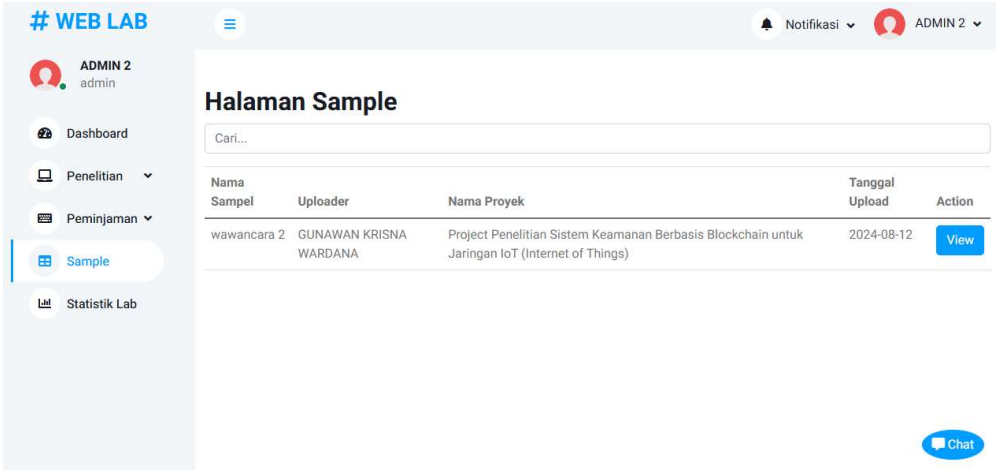


Gambar 4. 7 Hasil Halaman Home

Rute /sample

```
@app.route('/sample')
def sample():
    return render_template('sample.html', data_sample=data_sample)
```

Segmen Program 4. 7 Rute Sample

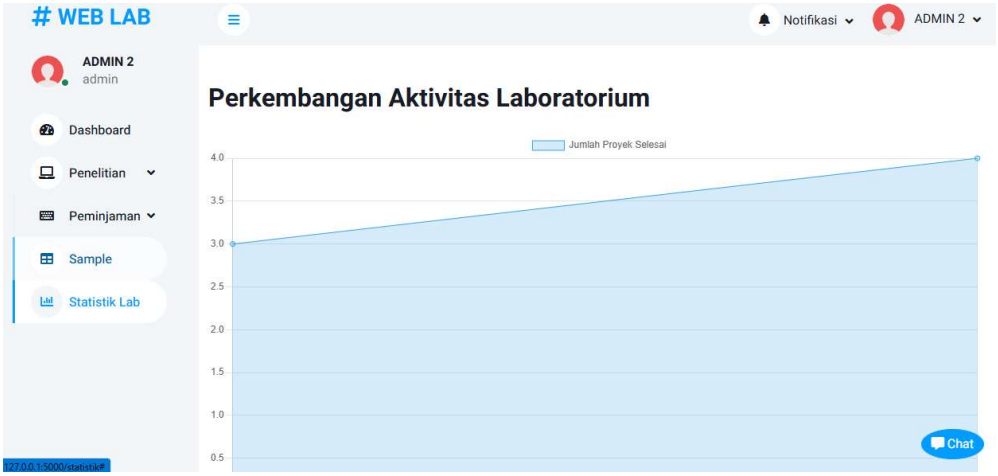


Gambar 4. 8 Hasil Halaman Sample

Rute /statistic

```
@app.route('/statistik')
def statistik():
    return render_template('statistik.html')
```

Segmen Program 4. 8 Rute Statistic



Gambar 4. 9 Hasil Halaman Satistik

4.2 Uji Coba

Bagian ini menjelaskan tentang ujicoba yang dilakukan terhadap produk yang dihasilkan. Ujicoba software dapat dilakukan pada lingkungan yang sebenarnya maupun dengan menggunakan metode black box.

4.2.1 Tahap Pengujian

Pada bagian ini, akan dijelaskan tahapan yang dilakukan untuk pelaksanaan pengujian. Pengujian akan dilakukan dengan menggunakan metode black box.

Target Pengujian:

1. Memastikan setiap fitur pada aplikasi berfungsi sesuai dengan yang diharapkan.
2. Menemukan bug atau kesalahan pada aplikasi sebelum digunakan oleh pengguna akhir.

Instrumen Pengujian:

1. Browser (Google Chrome, Mozilla Firefox, dll.)
2. Postman (untuk pengujian API)
3. Dokumen skenario pengujian
4. Tools pengujian otomatis seperti Katalon Studio

Urutan Langkah Pengujian:

1. **Pengujian Rute /login:**

    Langkah:

    1. Akses halaman login melalui browser.
    2. Masukkan email dan password yang valid.
    3. Klik tombol "Login".
    4. Periksa apakah pengguna diarahkan ke halaman home.

    Skenario Pengujian:

    1. Email dan password valid: Pengguna berhasil login.
    2. Email atau password tidak valid: Muncul pesan error.
    3. Kolom email atau password kosong: Muncul pesan peringatan.

Pengujian Rute /register:

Langkah:

1. Akses halaman registrasi melalui browser.

2. Isi formulir registrasi dengan data yang valid.
3. Klik tombol "Register".
4. Periksa apakah pengguna diarahkan ke halaman login.

Skenario Pengujian:

1. Semua data valid: Pengguna berhasil registrasi.
2. Email sudah terdaftar: Muncul pesan error.
3. Kolom yang wajib diisi kosong: Muncul pesan peringatan.

Pengujian Rute /home:

Langkah:

1. Login ke aplikasi.
2. Akses halaman home.
3. Periksa apakah halaman home ditampilkan dengan benar.

Skenario Pengujian:

1. Pengguna terautentikasi: Halaman home tampil dengan informasi pengguna.
2. Pengguna tidak terautentikasi: Pengguna diarahkan ke halaman login.

Pengujian Rute /newproject:

Langkah:

1. Login ke aplikasi.
2. Akses halaman new project.
3. Isi formulir proyek baru dengan data yang valid.
4. Klik tombol "Create".
5. Periksa apakah proyek baru ditambahkan dan ditampilkan di halaman my projects.

Skenario Pengujian:

1. Data valid: Proyek baru berhasil dibuat.
2. Kolom yang wajib diisi kosong: Muncul pesan peringatan.

Pengujian Rute /myprojects:

Langkah:

1. Login ke aplikasi.
2. Akses halaman my projects.
3. Periksa apakah daftar proyek pengguna ditampilkan dengan benar.

Skenario Pengujian:

1. Pengguna terautentikasi: Daftar proyek tampil dengan benar.
2. Pengguna tidak terautentikasi: Pengguna diarahkan ke halaman login.

Pengujian Rute /project/<int:project_id>:

Langkah:

1. Login ke aplikasi.
2. Akses halaman detail proyek dengan ID tertentu.
3. Periksa apakah detail proyek ditampilkan dengan benar.

Skenario Pengujian:

1. Proyek dengan ID yang valid: Detail proyek tampil dengan benar.

Proyek dengan ID yang tidak valid: Muncul pesan error.

Pengujian Rute /borrow_equipment:

Langkah:

1. Login ke aplikasi.
2. Akses halaman borrow equipment.
3. Pilih peralatan yang ingin dipinjam.
4. Klik tombol "Borrow".
5. Periksa apakah status peralatan berubah menjadi tidak tersedia.

Skenario Pengujian:

1. Peralatan tersedia: Peralatan berhasil dipinjam.

2. Peralatan tidak tersedia: Muncul pesan peringatan.

Pengujian Rute /sample:

Langkah:

1. Login ke aplikasi.
2. Akses halaman sample.
3. Periksa apakah data sampel ditampilkan dengan benar.

Skenario Pengujian:

1. Pengguna terautentikasi: Data sampel tampil dengan benar.
2. Pengguna tidak terautentikasi: Pengguna diarahkan ke halaman login.

Pengujian Rute /statistik:

Langkah:

1. Login ke aplikasi.
2. Akses halaman statistik.
3. Periksa apakah statistik aplikasi ditampilkan dengan benar.

Skenario Pengujian:

1. Pengguna terautentikasi: Statistik tampil dengan benar.

Pengguna tidak terautentikasi: Pengguna diarahkan ke halaman login.

4.2.2 Hasil Pengujian

Berikut adalah hasil pengujian yang dilakukan sesuai dengan metode pengujian dan instrumen yang telah ditetapkan.

Tabel Rekap Hasil Pengujian:

Tabel 4. 1 Rekap Hasil Pengujian

No	Nama Fungsi	Deskripsi Kegunaan	Target Output	Hasil Pengujian
1	Login	Melakukan pengecekan apakah user berhak masuk/akses ke sistem atau tidak	Jika berhak akan menampilkan halaman dashboard, Jika tidak berhak akan muncul pesan error atau peringatan	Sesuai
2	Register	Mendaftarkan pengguna baru ke sistem	Jika semua data valid, pengguna berhasil registrasi, Jika email sudah terdaftar atau kolom wajib kosong, akan muncul pesan error atau peringatan	Sesuai
3	Home	Menampilkan halaman utama setelah login	Jika pengguna terautentikasi, halaman home tampil; jika tidak terautentikasi, diarahkan ke login	Sesuai
4	New Project	Membuat proyek baru dalam sistem	Jika data valid, proyek baru berhasil dibuat; jika kolom wajib kosong, muncul pesan peringatan	Sesuai
5	My Projects	Menampilkan daftar proyek milik pengguna	Jika pengguna terautentikasi, daftar proyek tampil; jika tidak terautentikasi, diarahkan ke login	Sesuai
6	Project Details	Menampilkan detail proyek berdasarkan ID	Jika ID valid, detail proyek tampil; jika ID tidak valid, muncul pesan error	Sesuai

No	Nama Fungsi	Deskripsi Kegunaan	Target Output	Hasil Pengujian
7	Borrow Equipment	Meminjam peralatan dari sistem	Jika peralatan tersedia, berhasil dipinjam; jika tidak tersedia, muncul pesan peringatan	Sesuai
8	Sample Data	Menampilkan data sampel terkait proyek	Jika pengguna terautentikasi, data sampel tampil; jika tidak terautentikasi, diarahkan ke login	Sesuai
9	Statistik	Menampilkan statistik terkait proyek atau penggunaan sistem	Jika pengguna terautentikasi, statistik tampil; jika tidak terautentikasi, diarahkan ke login	Sesuai

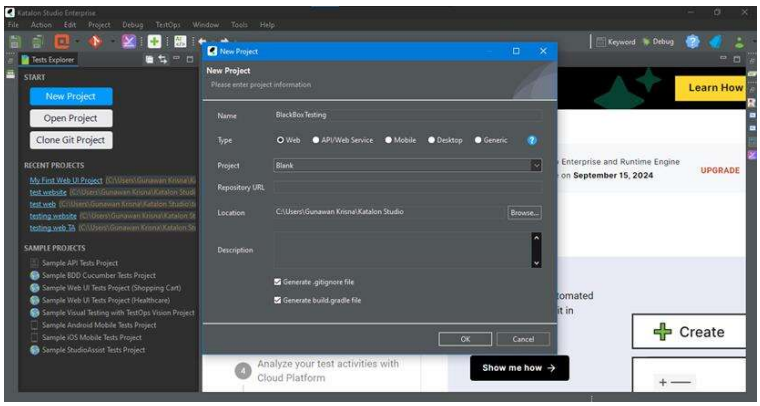
Pengujian dilakukan dengan menggunakan browser dan Postman untuk memastikan setiap rute dan fungsionalitas bekerja dengan baik sesuai dengan skenario pengujian yang telah ditetapkan. Hasil pengujian menunjukkan bahwa semua rute dan fungsionalitas berhasil bekerja dengan baik tanpa ditemukan bug atau kesalahan yang signifikan.

Berikut adalah langkah-langkah untuk menggunakan Katalon Studio dalam melakukan uji coba black box, termasuk pembuatan laporan PDF:

1. Membuat Proyek Baru

1. Buka Katalon Studio.
2. Pilih File > New > Project.
3. Berikan nama proyek Anda, misalnya "BlackBoxTesting".
4. Pilih General > Empty Project, lalu klik Next.

5. Tentukan lokasi penyimpanan proyek dan klik Finish.

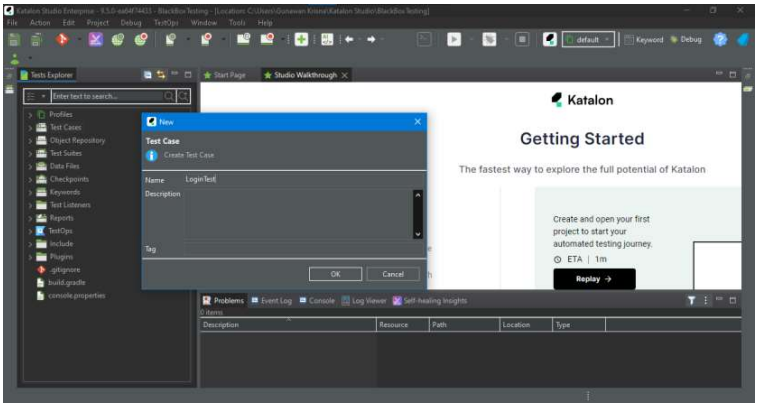


Gambar 4. 10 Buat Project Baru

2. Menggunakan Web Recorder

1. Buka Katalon Studio dan navigasikan ke Test Cases.

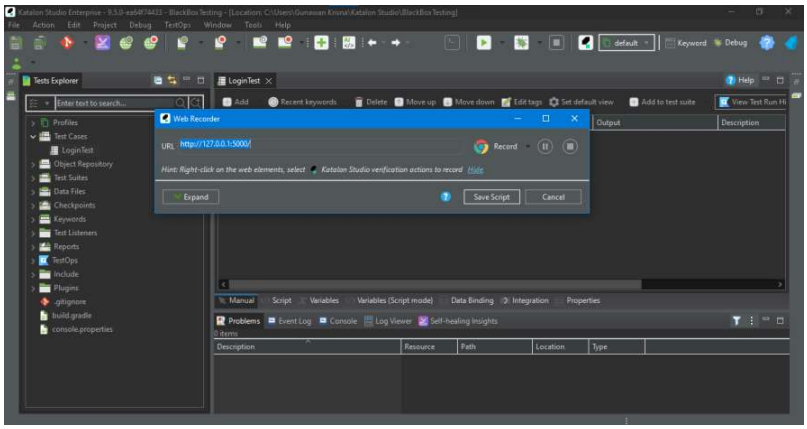
2. Klik kanan pada Test Cases dan pilih New > Test Case. Beri nama test case, misalnya "LoginTest".



Gambar 4. 11 Buat Test Cases

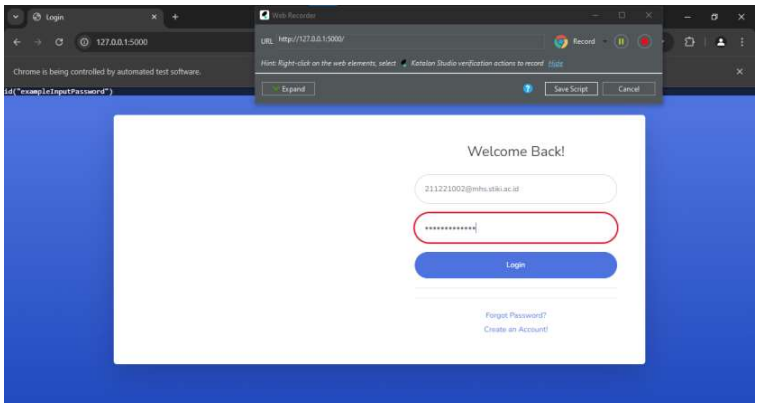
3. Pada toolbar, pilih Record Web untuk memulai perekaman.

4. Masukkan URL dari aplikasi web yang ingin diuji pada kolom URL, lalu pilih browser yang akan digunakan untuk perekaman.



Gambar 4. 12 Toolbar Record Web

5. Klik Record untuk mulai perekaman. Katalon akan membuka browser yang dipilih dan mulai merekam setiap tindakan yang Anda lakukan, seperti mengisi formulir, mengklik tombol, dll.

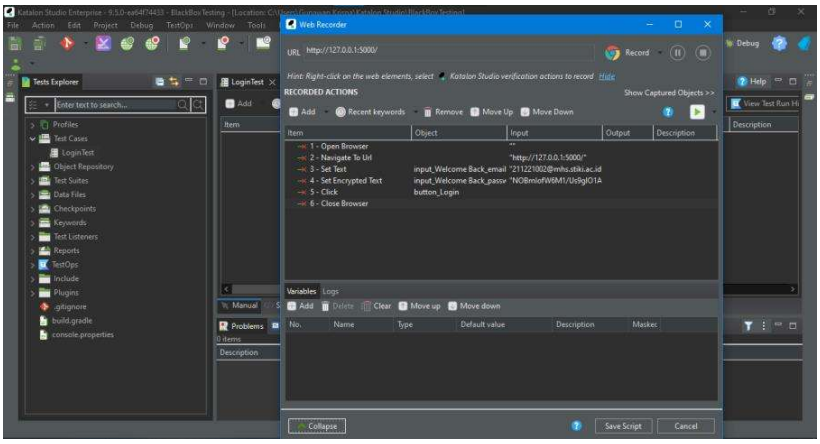


Gambar 4. 13 Perekaman menggunakan Record Web

6. Setelah selesai, klik Stop pada toolbar perekaman.

7. Semua langkah yang direkam akan ditampilkan dalam bentuk skrip di Katalon Studio. Anda dapat memodifikasi skrip ini jika diperlukan.

8. Simpan test case yang telah direkam.



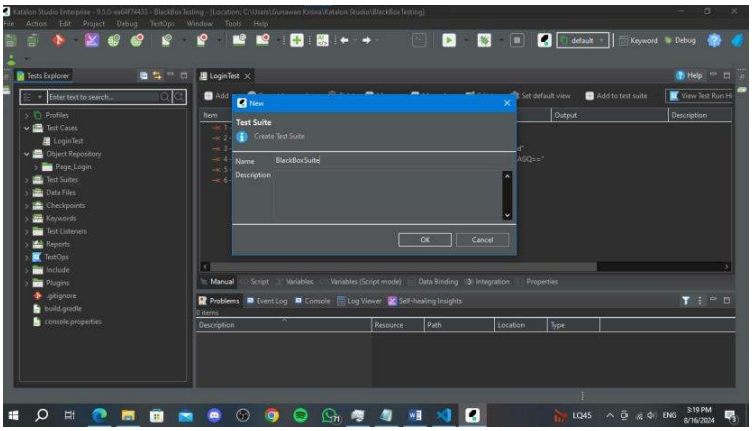
Gambar 4. 14 Hasil Menggunakan Record Web

3. Menyusun Test Suite

1. Klik kanan pada folder Test Suites lalu pilih New > Test Suite.

2. Beri nama Test Suite, misalnya "BlackBoxSuite", lalu klik OK.

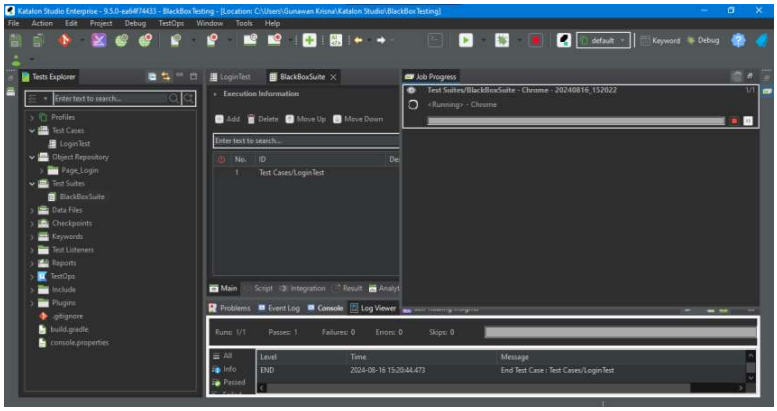
3. Tambahkan test case yang telah direkam ke dalam Test Suite dengan klik ikon Add dan memilih test cases.



Gambar 4. 15 Test Suit

4. Menjalankan Uji Coba

1. Pilih Run pada toolbar dan pilih browser yang diinginkan untuk menjalankan uji coba.
2. Katalon Studio akan menjalankan semua test case dalam Test Suite yang dipilih.



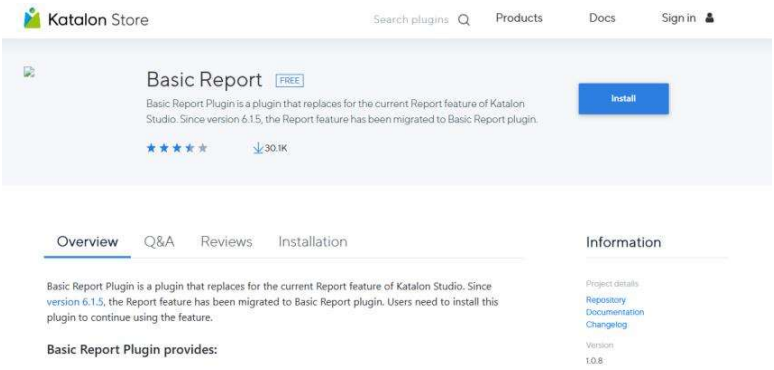
Gambar 4. 16 Run pada Test Suit

5. Menghasilkan Laporan Uji

1. Setelah uji coba selesai, Katalon akan secara otomatis menghasilkan laporan dalam format HTML.
2. Untuk menghasilkan laporan dalam format PDF, Anda perlu menggunakan plugin Katalon Reports.

6. Menginstal Plugin Katalon Reports

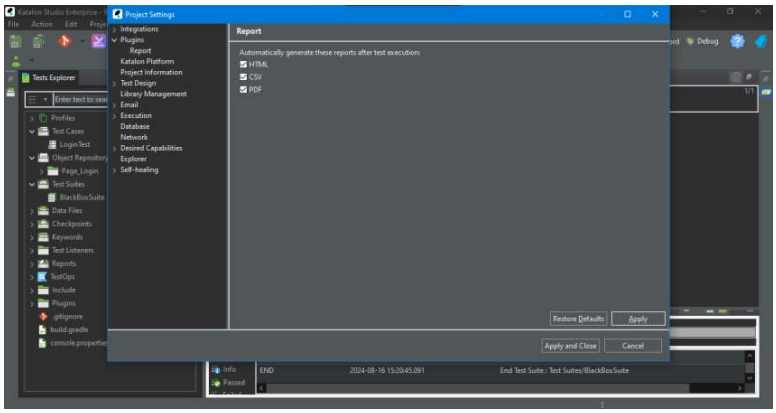
1. Buka Katalon Studio > Katalon Store.
2. Cari Katalon Reports plugin, lalu klik Install.
3. Ikuti petunjuk instalasi hingga selesai.



Gambar 4. 17 Basic Report Plugin

7. Konfigurasi Plugin untuk PDF Report

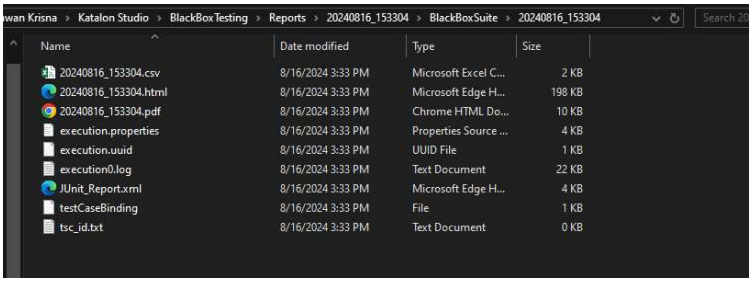
1. Setelah plugin terinstal, buka Project > Settings > Plugins > Report.
2. Centang opsi Enable pada bagian PDF untuk mengaktifkan laporan dalam format PDF.
3. Konfigurasikan opsi lainnya sesuai kebutuhan Anda.



Gambar 4. 18 Setting Plugin

8. Menjalankan Uji Coba dengan Laporan PDF

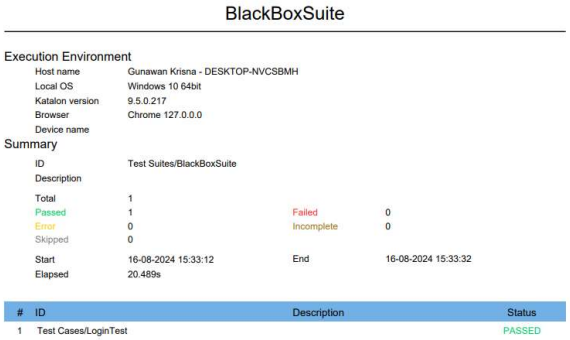
1. Jalankan Test Suite seperti biasa.
2. Setelah uji selesai, Katalon akan menghasilkan laporan PDF secara otomatis di direktori Reports proyek Anda.



Gambar 4. 19 Hasil Laporan

9. Memeriksa dan Mengunduh Laporan PDF

1. Buka folder Reports dalam direktori proyek.
2. Di sana Anda akan menemukan laporan dalam format PDF yang mencakup semua hasil uji.



BlackBoxSuite

Execution Environment

Host name	Gunawan Krisna - DESKTOP-NVCSBMH
Local OS	Windows 10 64bit
Katalon version	9.5.0.217
Browser	Chrome 127.0.0.0
Device name	

Summary

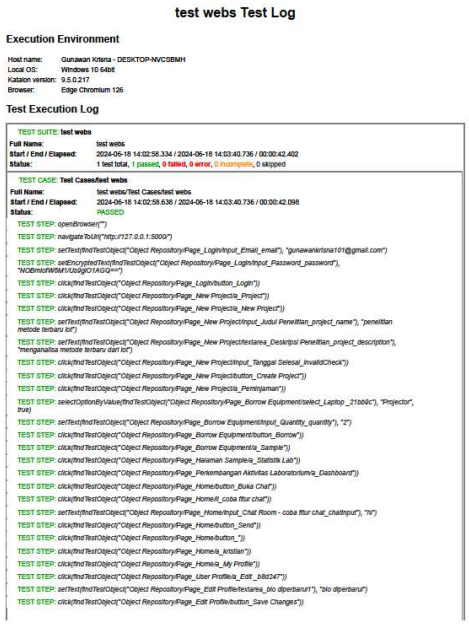
ID	Test Suites/BlackBoxSuite		
Description			
Total	1		
Passed	1	Failed	0
Error	0	Incomplete	0
Skipped	0		
Start	16-08-2024 15:33:12	End	16-08-2024 15:33:32
Elapsed	20.489s		

#	ID	Description	Status
1	Test Cases/LoginTest		PASSED

Gambar 4. 20 Hasil dari Blackboxsuite

Dengan mengikuti langkah-langkah di atas, Dapat melakukan pengujian black box menggunakan Katalon Studio dan menghasilkan laporan dalam format PDF untuk dokumentasi.

Rekap Hasil Pengujian dari katalon studio:



test webs Test Log

Execution Environment

Host name: Gunawan Krisna - DESKTOP-NVCSBMH
Local OS: Windows 10 64bit
Katalon version: 9.5.0.217
Browser: Edge Chromium 126

Test Execution Log

TEST SUITE: test webs
Full Name: test webs
Start / End / Elapsed: 2024-06-18 14:02:58.334 / 2024-06-18 14:03:40.736 / 00:00:42.402
Status: 1 test total, 1 passed, 0 failed, 0 error, 0 incomplete, 0 skipped

TEST CASE: Test Cases/test webs
Full Name: test webs/Test Cases/test webs
Start / End / Elapsed: 2024-06-18 14:02:58.638 / 2024-06-18 14:03:40.736 / 00:00:42.098
Status: PASSED

TEST STEP: openBrowser("")
TEST STEP: navigateToUrl("http://127.0.0.1:8000/")
TEST STEP: setText(findTestObject("Object Repository/Page_Login/input_Email_email"), "gunawankrisna101@gmail.com")
TEST STEP: setEncryptedText(findTestObject("Object Repository/Page_Login/input_Password_password"), "WOBmIorlW9M1/UoGgIO1AGQ==")
TEST STEP: click(findTestObject("Object Repository/Page_Login/button_Login"))
TEST STEP: click(findTestObject("Object Repository/Page_New Project/a_Project"))
TEST STEP: click(findTestObject("Object Repository/Page_New Project/a_New Project"))
TEST STEP: setText(findTestObject("Object Repository/Page_New Project/input_Judul Penelitian_project_name"), "penelitian metode terbaru ibz")
TEST STEP: setText(findTestObject("Object Repository/Page_New Project/textarea_Deskripsi Penelitian_project_description"), "menganalisa metode terbaru dari ibz")
TEST STEP: click(findTestObject("Object Repository/Page_New Project/input_Tanggal Selesai_invalidCheck"))
TEST STEP: click(findTestObject("Object Repository/Page_New Project/button_Create Project"))
TEST STEP: click(findTestObject("Object Repository/Page_New Project/a_Peminjaman"))
TEST STEP: selectOptionByValue(findTestObject("Object Repository/Page_Borrow Equipment/select_Laptop _21bb9c"), "Projector", true)
TEST STEP: setText(findTestObject("Object Repository/Page_Borrow Equipment/input_Quantity_quantity"), "2")
TEST STEP: click(findTestObject("Object Repository/Page_Borrow Equipment/button_Borrow"))
TEST STEP: click(findTestObject("Object Repository/Page_Borrow Equipment/a_Sample")
TEST STEP: click(findTestObject("Object Repository/Page_Halaman Sample/a_Statistik Lab"))
TEST STEP: click(findTestObject("Object Repository/Page_Perkembangan Aktivitas Laboratorium/a_Dashboard"))
TEST STEP: click(findTestObject("Object Repository/Page_Home/button_Buka Chat"))
TEST STEP: click(findTestObject("Object Repository/Page_Home/fl_coba fitur chat"))
TEST STEP: setText(findTestObject("Object Repository/Page_Home/input_Chat Room - coba fitur chat_chatInput"), "hi")
TEST STEP: click(findTestObject("Object Repository/Page_Home/button_Send"))
TEST STEP: click(findTestObject("Object Repository/Page_Home/button_"))
TEST STEP: click(findTestObject("Object Repository/Page_Home/a_kristian"))
TEST STEP: click(findTestObject("Object Repository/Page_Home/a_My Profile"))
TEST STEP: click(findTestObject("Object Repository/Page_User Profile/a_Edit _b8d247"))
TEST STEP: setText(findTestObject("Object Repository/Page_Edit Profile/textarea_bio diperbarui1"), "bio diperbarui")
TEST STEP: click(findTestObject("Object Repository/Page_Edit Profile/button_Save Changes"))

Gambar 4. 21 Rekap Hasil Katalon Studio