

BAB V

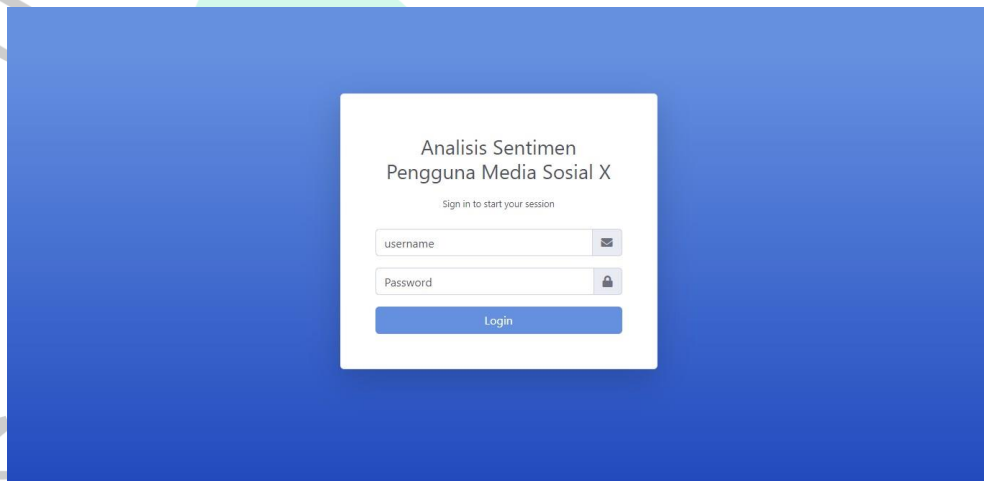
HASIL DAN PEMBAHASAN

Bab 5 fokus pada hasil dan analisis sistem yang telah dirancang dalam penelitian ini. Tahap ini juga melibatkan pengujian sistem dengan menggunakan pendekatan *blackbox testing* dan *whitebox testing*.

5.1 Hasil Implementasi dari Perancangan Antarmuka

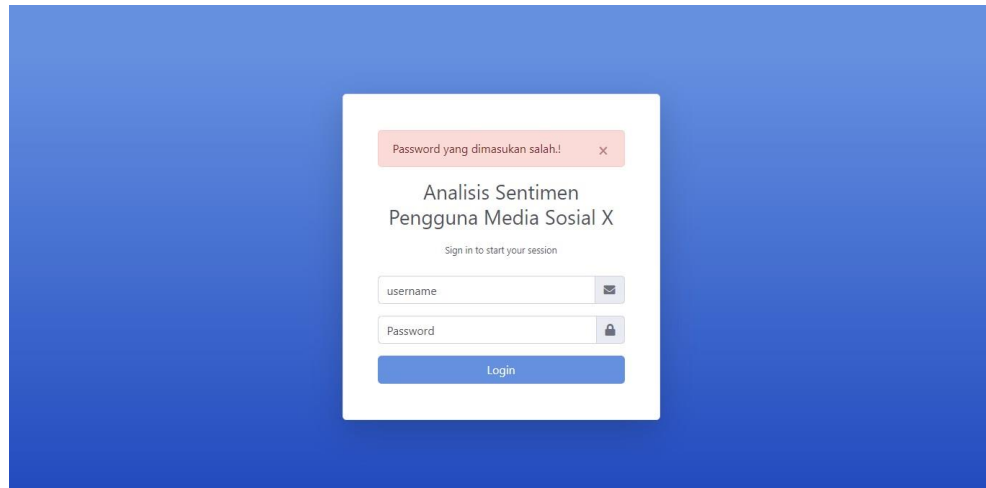
Perancangan aplikasi *website* untuk melakukan analisis sentimen mobil listrik yang berhasil di implementasikan sesuai dengan rancangan yang telah dibuat sebelumnya. Berikut merupakan hasil implementasinya antara lain:

1. Tampilan Halaman Login



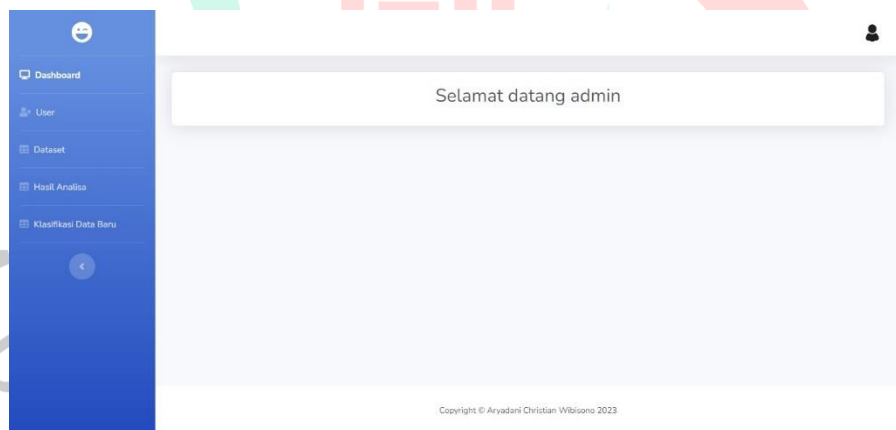
Gambar 5.1 Tampilan Halaman Login

Gambar 5.1 menampilkan tampilan login ketika *user* akan masuk ke dalam website. Halaman ini juga terdapat form login yang berisikan *username* dan *password*. Dimana pengguna dapat memasukkan *username* dan *password* yang sudah terdaftar di sistem sebelumnya.



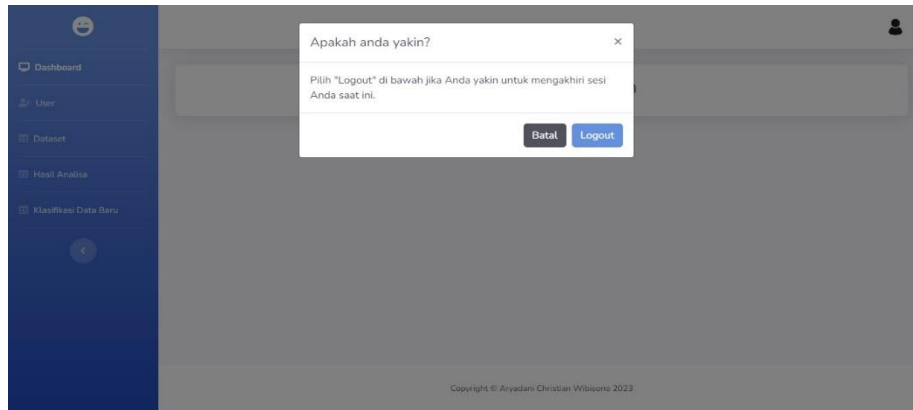
Gambar 5.2 Tampilan Halaman Gagal Login

2. Jika *username* dan *password* yang dimasukkan benar, pengguna langsung diarahkan ke Halaman utama. Sedang *username* dan *password* yang dimasukkan salah, sistem akan memberikan warning, seperti pada Gambar 5.3.



Gambar 5.3 Tampilan Halaman Dashboard Admin

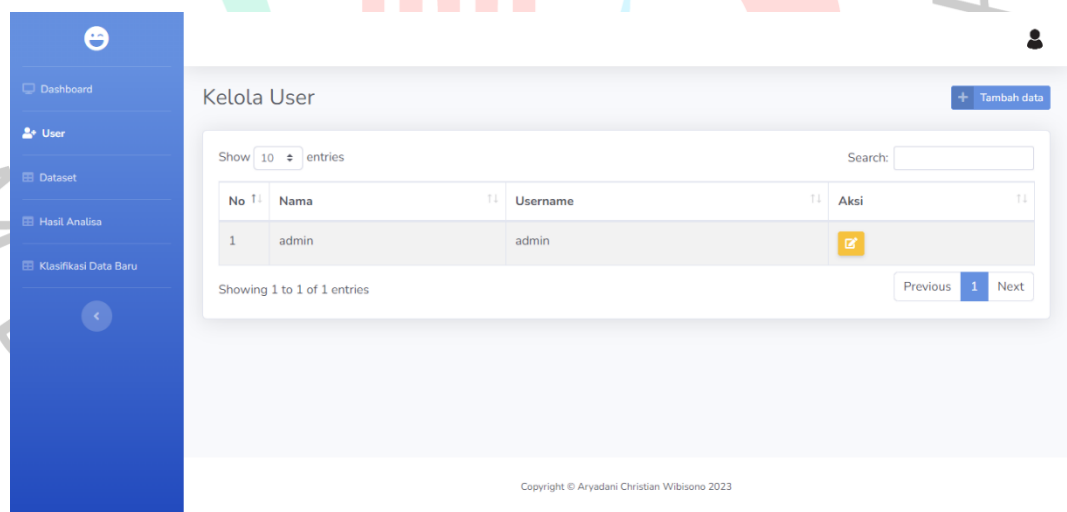
Gambar 5.3 menunjukkan Halaman *dashboard* admin, dimana setelah admin berhasil melakukan login. Halaman ini juga memberikan kemudahan admin untuk memberikan informasi kepada pengguna.



Gambar 5.4 Tampilan Halaman Logout

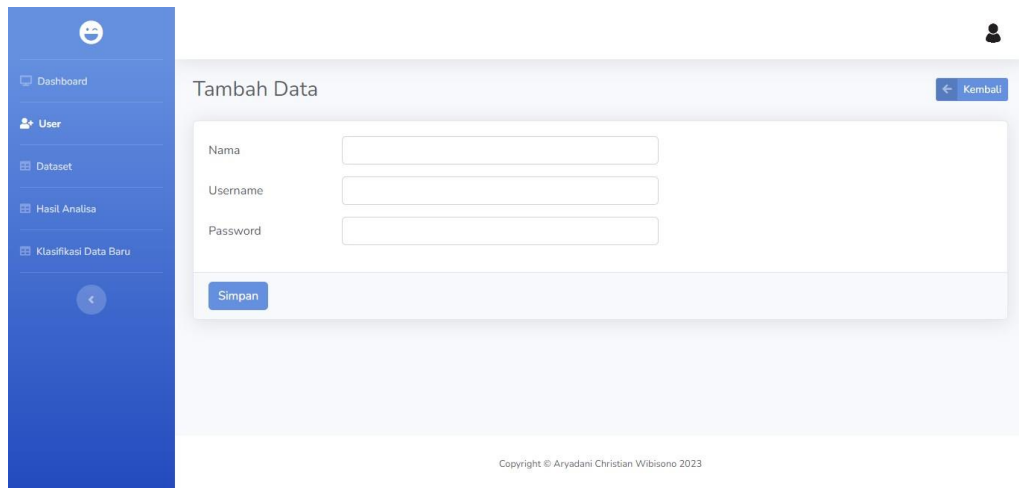
Pada Gambar 5.4 merupakan tampilan *logout*, jika admin ingin keluar dari *website* dapat melakukan klik logout pada simbol admin, setelah itu sistem akan menampilkan popup “apakah anda yakin untuk mengakhiri sesi anda saat ini”.

3. Tampilan Halaman Kelola *User*



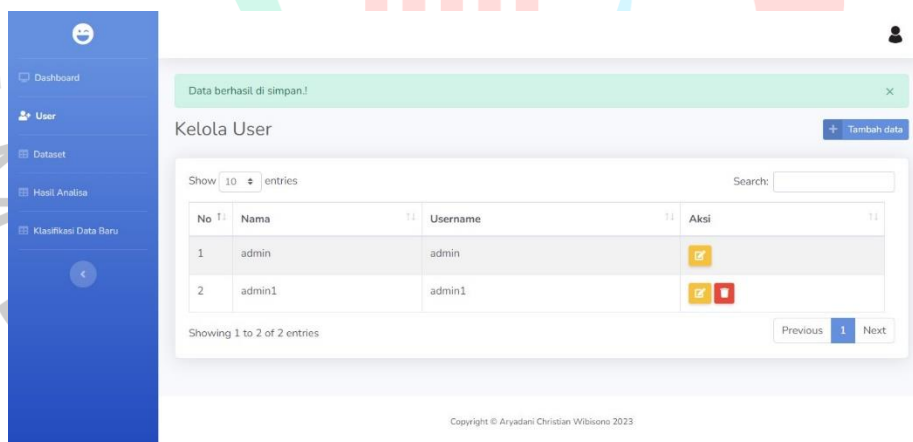
Gambar 5.5 Tampilan Halaman Kelola User

Pada Gambar 5.5 merupakan tampilan Kelola user yang dapat diakses oleh admin. Halaman ini berisikan informasi *user* seperti nama, *username*. Admin dapat melakukan menambahkan, mengedit, bahkan menghapus data *user* sesuai dengan kebutuhan data.



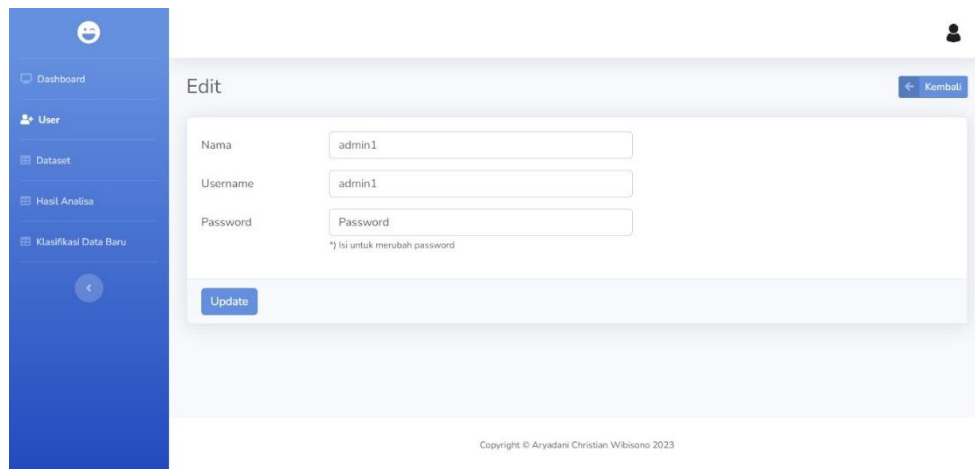
Gambar 5.6 Tampilan Halaman Tambah Data

Gambar 5.6 merupakan halaman tambah data yang dapat diakses oleh admin dengan menambahkan data user sesuai dengan form tambah data, seperti nama, *username*, dan *password*, kemudian admin klik simpan. lalu sistem akan menyimpannya ke database.



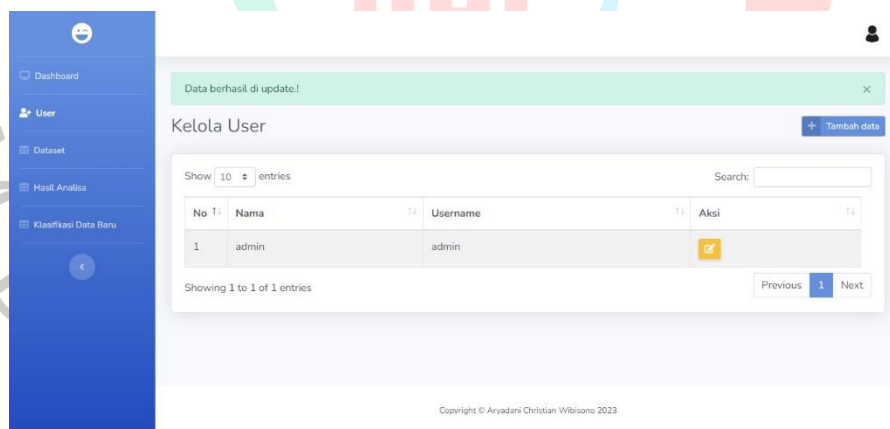
Gambar 5.7 Tampilan Data Berhasil Di Simpan

Gambar 5.7 merupakan tampilan jika data yang dimasukkan benar, maka sistem akan menampilkan pop up “Data berhasil di simpan”.



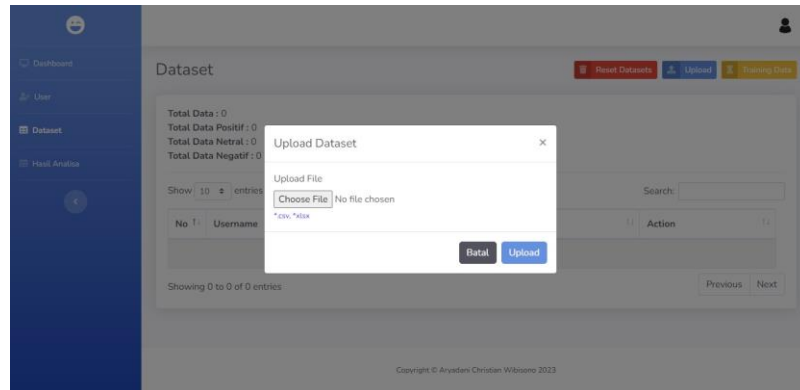
Gambar 5. 8 Tampilan Halaman Edit

Gambar 5.8 merupakan halaman tambah edit data yang dilakukan oleh admin. Dimana admin dapat mengubah data *user* sesuai dengan data yang ingin diubah, setelah itu klik “update”. Lalu sistem akan mengupdate perubahan yang dilakukan oleh admin.



Gambar 5. 9 Tampilan Data Berhasil Di Simpan

Gambar 5.9 merupakan tampilan jika data yang dimasukkan benar, maka sistem akan menampilkan popup “Data berhasil di simpan”.

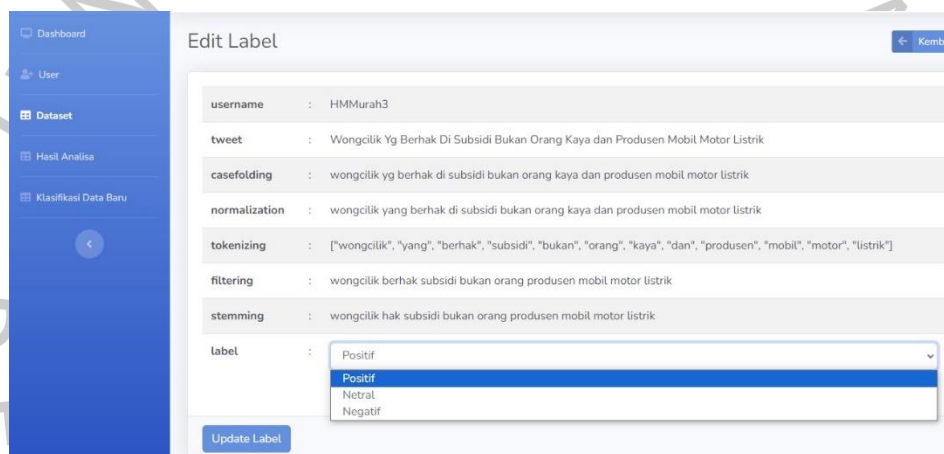


Gambar 5.10 Tampilan Upload Dataset

Gambar 5.10 merupakan tampilan upload dataset yang memungkinkan pengguna, dalam hal ini admin, untuk mengunggah dataset atau kumpulan data ke dalam sistem. Desain halaman ini bertujuan untuk memudahkan proses pengunggahan dan memastikan bahwa data yang dimasukkan sesuai dengan format.

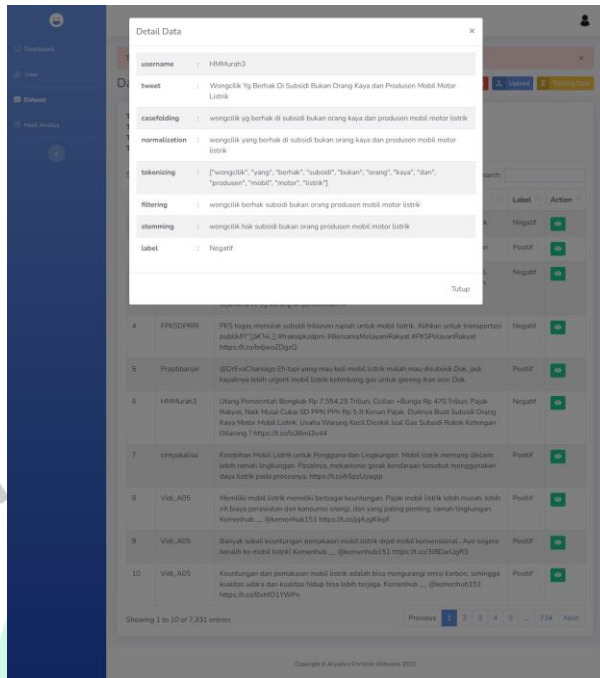
Gambar 5.11 Tampilan Dataset

Gambar 5.11 merupakan tampilan dataset yang memperlihatkan informasi dan struktur dataset secara terorganisir. Halaman ini memberikan pengguna, termasuk administrator atau analis data, wawasan tentang isi dataset dengan cara yang mudah dipahami. Halaman dataset menampilkan tabel yang menyajikan kolom-kolom dataset bersama dengan data yang terkandung di dalamnya. Setiap kolom memiliki label yang jelas untuk mengidentifikasi jenis data yang diwakilinya, dan baris-baris tabel menunjukkan entri data yang sesuai. Pada tampilan ini, pengguna dapat dengan cepat memahami struktur dataset dan nilai-nilai yang ada.



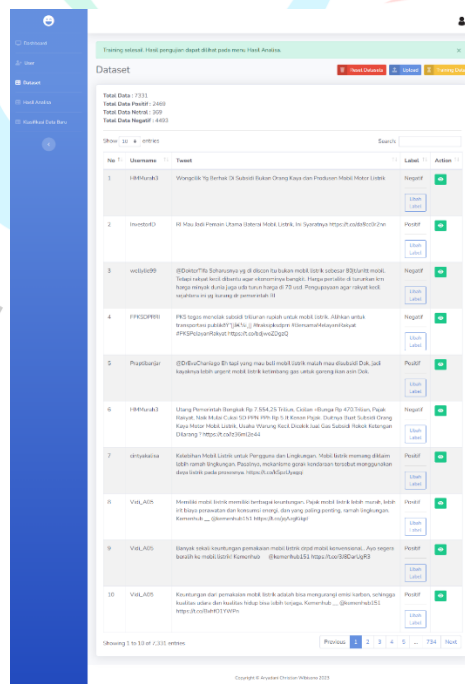
Gambar 5.12 Tampilan Edit Label

Gambar 5.12 menampilkan antarmuka untuk pembaruan label yang dapat diakses oleh pengguna, memungkinkan mereka untuk mengubah label menjadi negatif, positif, atau netral sesuai kebutuhan.



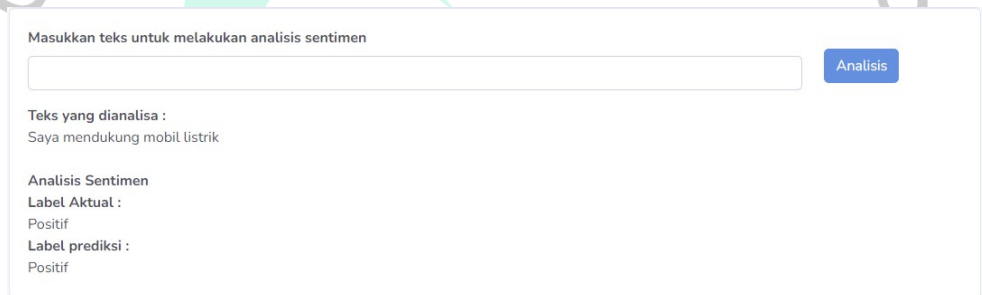
Gambar 5.13 Tampilan Detail Data

Gambar 5.13 merupakan tampilan detail data merupakan tampilan secara detail dari dataset yang sudah di upload sebelumnya. Dimana data tersebut berisikan *username*, *tweet*, *casefolding*, *normalization*, *tokenizing*, *filtering*, *stemming*, dan *label*.



Gambar 5.14 Tampilan Training Data

Gambar 5.14 merupakan tampilan *training* data yang memperlihatkan data yang digunakan untuk melatih suatu model atau algoritma machine learning. Halaman ini memberikan pandangan komprehensif terhadap dataset yang digunakan dalam proses pelatihan, memungkinkan pengguna untuk memahami karakteristik dan distribusi data yang memberikan dasar pembelajaran pada model. Halaman training data menampilkan tabel yang memuat kolom-kolom dataset yang digunakan untuk melatih model. Setiap kolom mewakili fitur atau variabel tertentu, dan baris-baris dataset menunjukkan entri data yang digunakan sebagai sampel latihan. Label-label di setiap kolom memberikan informasi tentang jenis data yang diwakilinya, membantu pengguna memahami makna setiap variabel.



Masukkan teks untuk melakukan analisis sentimen

Teks yang dianalisa :
Saya mendukung mobil listrik

Analisis Sentimen
Label Aktual :
Positif
Label prediksi :
Positif

Gambar 5.15 Tampilan Hasil Klasifikasi Data Baru

Gambar 5.15 merupakan Proses memasukkan teks untuk melakukan analisis sentimen dimulai dengan pengguna menyediakan teks atau paragraf yang ingin dianalisis. Ini bisa menjadi ulasan produk, tanggapan pelanggan, atau teks apa pun yang mengandung opini atau sentimen yang ingin diukur. Pengguna biasanya melakukan langkah-langkah berikut: Pertama, pengguna memasukkan teks ke dalam formulir atau area input yang disediakan oleh platform atau aplikasi analisis sentimen. Teks ini dapat berupa satu kalimat, beberapa kalimat, atau paragraf panjang tergantung pada kebutuhan analisis. Setelah memasukkan teks, pengguna kemudian memicu proses analisis sentimen dengan menekan tombol "Submit" atau tombol serupa. Sistem analisis sentimen, yang dapat mencakup algoritma pemrosesan bahasa alami dan model machine learning, kemudian mulai memproses teks untuk mengekstrak makna sentimen yang terkandung di dalamnya. Algoritma

analisis sentimen akan menilai teks berdasarkan kata-kata kunci, struktur kalimat, dan konteks secara menyeluruh. Dengan menggunakan model yang telah dilatih sebelumnya, sistem akan mengidentifikasi apakah sentimen teks tersebut bersifat positif, negatif, atau netral. Setelah proses analisis selesai, pengguna melihat hasilnya. Hasil ini bisa berupa label sentimen, seperti "positif," "negatif," atau "netral," atau skor numerik yang merepresentasikan tingkat sentimen. Informasi ini memberikan gambaran tentang bagaimana teks tersebut diartikan dari sudut pandang sentimen.

The screenshot displays a web application interface for sentiment analysis. It is divided into three main sections: Training Data, Testing Data, and Hasil Analisis Sentimen Menggunakan SVM.

Training Data: Shows 10 training entries with their original text and predicted labels. For example, entry 1 is "hariku sadar tidak terbayang sudah diujung jalan capes anda bisa subadi kendara listrik untuk transportasi umum..." with a predicted label of "Negatif".

Testing Data: Shows 10 testing entries with their original text and predicted labels. For example, entry 1 is "saya bisa akan napa saya belajar dengan siapa pembali program bangun sendiri jalan kali semesta maka satu adalah..." with a predicted label of "Positif".

Hasil Analisis Sentimen Menggunakan SVM: This section includes a confusion matrix and performance metrics. The confusion matrix shows the relationship between actual and predicted labels. The metrics are: Accuracy = 84.05%, Precision = 81.05%, Recall = 84.05%, and F-Score = 82.42%.

	Actual Positif	Actual Negatif
Predicted Positif	10	0
Predicted Negatif	0	10

Accuracy = 84.05 %
Precision = 81.05 %
Recall = 84.05 %
F-Score = 82.42 %

Gambar 5.16 Tampilan Hasil Analisa

Gambar 5.16 merupakan tampilan hasil analisa sentimen menunjukkan bahwa dari total 1698 data testing yang dianalisis, sebanyak 600 data memiliki sentimen positif, 1009 data memiliki sentimen negatif, dan 89 data bersifat netral.

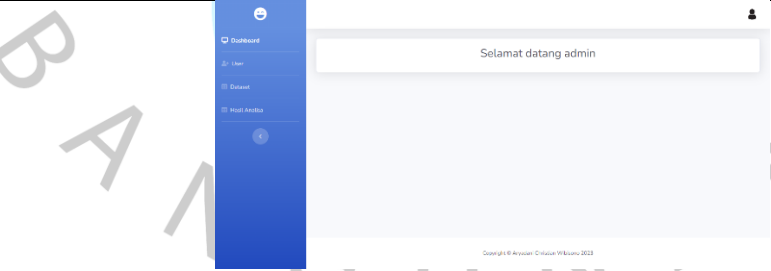
5.2 Pembahasan

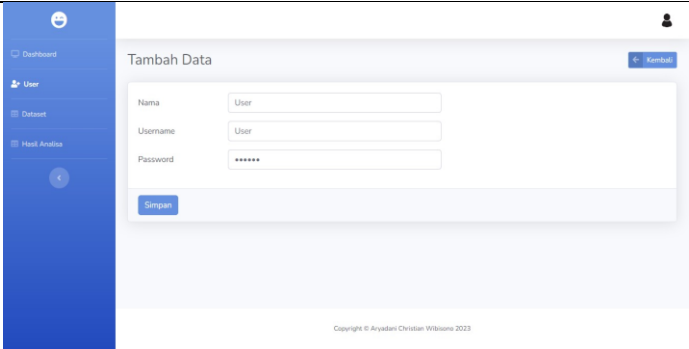
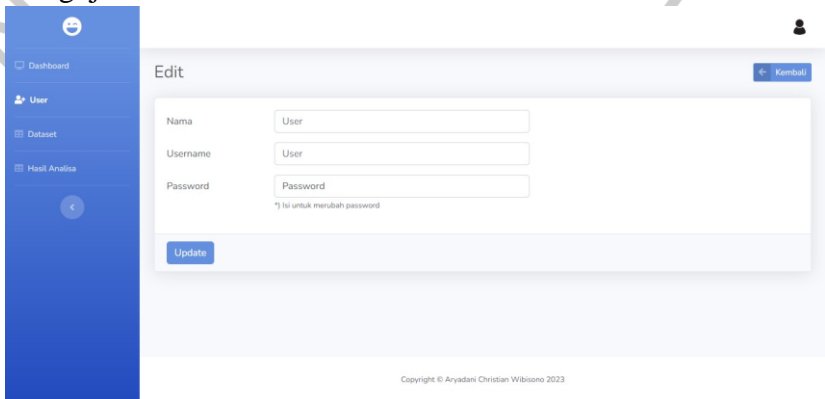
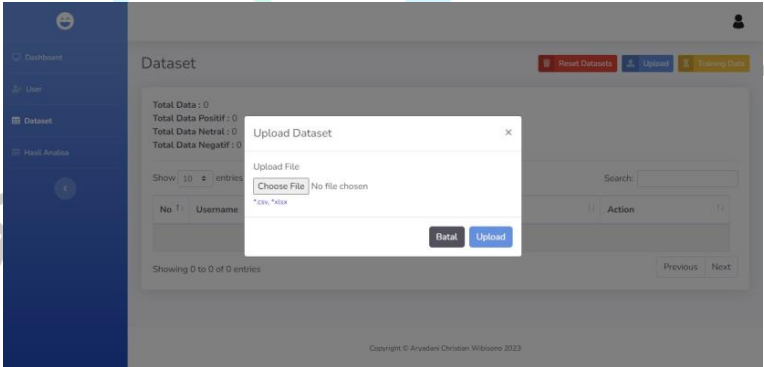
Pembahasan merupakan hasil dari evaluasi sistem yang dilakukan untuk mengetahui fungsi dari aplikasi pencarian penggunaan mobil listrik sesuai dengan rancangannya atau tidak.

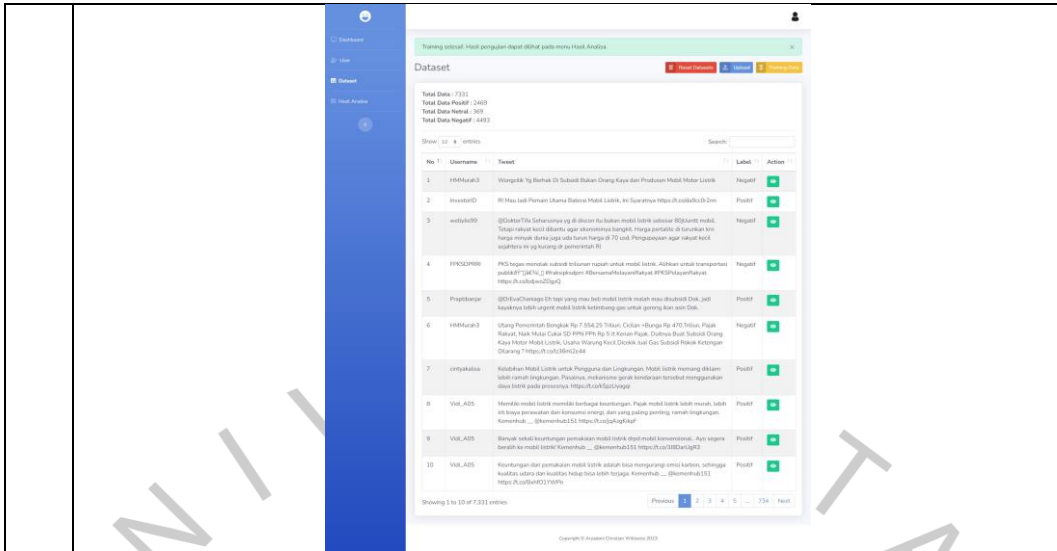
5.2.1 Hasil Pengujian *Blackbox*

Black box testing merupakan metode pengujian eksternal yang bertujuan untuk memastikan suatu sistem atau aplikasi agar dapat memenuhi kebutuhan dan harapan pengguna. Pengujian ini berfokus untuk menguji kualitas terhadap semua fitur aplikasi, sehingga penguji dapat memastikan bahwa sistem atau aplikasi bekerja dengan benar.

Tabel 5.1 Skenario Pengujian *Black Box*

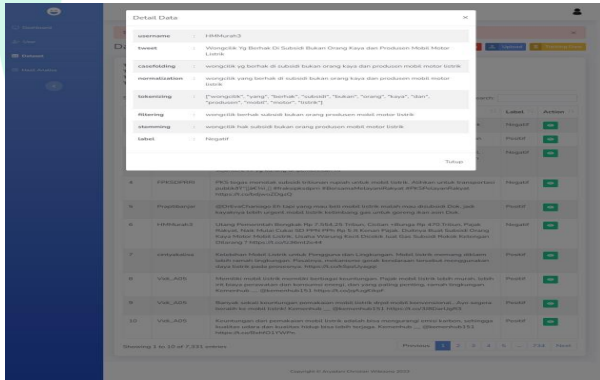
No	Skenario Pengujian	Hasil yang diharapkan
1	Pengguna klik button login untuk masuk ke Halaman utama	Muncul Halaman dashboard
	Hasil Pengujian	
		
2	Pengguna klik Tambah data untuk menambahkan data user.	Muncul Halaman Tambah data
	Hasil Pengujian	

		
3	<p>Pengguna klik Edit data untuk mengubah data user.</p>	<p>Muncul Halaman Edit data</p>
	<p>Hasil Pengujian</p> 	
4	<p>Pengguna klik upload dataset untuk menampilkan dataset</p>	<p>Muncul Halaman dataset</p>
	<p>Hasil Pengujian</p> 	
5	<p>Pengguna klik Training data untuk melakukan proses training dataset</p>	<p>Muncul training data</p>
	<p>Hasil Pengujian</p>	



6 Pengguna klik simbol mata untuk melihat data secara detail Muncul Halaman detail data

Hasil Pengujian



7 Pengguna memasukkan teks untuk melakukan analisis sentimen Muncul Halaman analisis sentimen

Hasil Pengujian

Masukkan teks untuk melakukan analisis sentimen

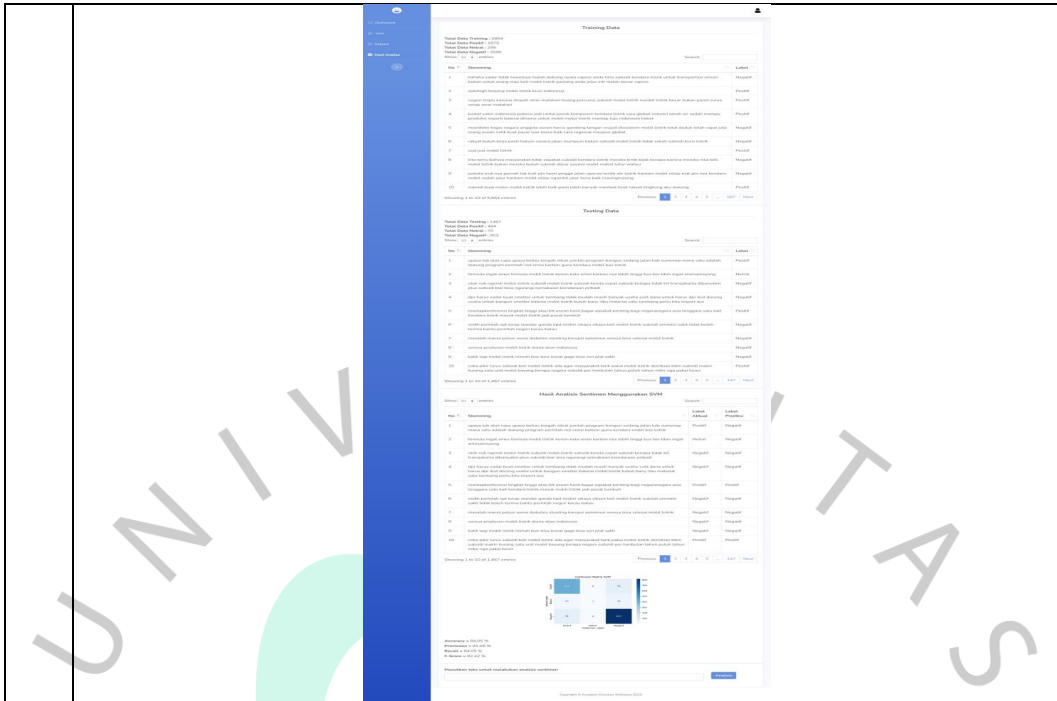
Analisis

Teks yang dianalisa :
Saya mendukung mobil listrik

Analisis Sentimen
Label Aktual :
Positif
Label prediksi :
Positif

8 Pengguna klik analisis untuk melakukan proses eksekusi analisis sentimen sesuai dengan teks yang dimasukkan. Muncul Halaman analisis sentimen

Hasil Pengujian



5.2.2 Hasil Pengujian White Box

White box testing merupakan metode pengujian internal dengan memastikan bahwa kode sistem atau aplikasi bebas dari *bug* dan memenuhi standar kualitas yang ditetapkan. Selain itu, pengujian ini bertujuan untuk memeriksa bagaimana masing-masing bagian sistem atau aplikasi bekerja secara internal. Pengujian ini didasarkan pada rincian prosedur dan alur logika dalam sistem.

Tabel 5.2 Skenario *White Box*

No	Hasil Yang Diharapkan	Source Code
1	Menambahkan data	<pre data-bbox="603 398 1353 734">def store(request): try: post = request.form password = bcrypt.hashpw(post['password'].encode('utf8'), bcrypt.gensalt()) checkUsername = User.get_by_username(post['username']) if checkUsername == None: data = { "nama" : post['nama'], "username" : post['username'], "password" : password } User.store(data) flash('Data berhasil di simpan!', 'success') return redirect(url_for('user_index')) else: flash('Username sudah terdaftar', 'danger') return redirect(url_for('user_create')) except Exception as e: return 'Something went wrong ' + str(e)</pre> <p data-bbox="603 757 970 846">Hasil Pengujian: Berhasil menambahkan data</p>
2	Melakukan edit data	<pre data-bbox="603 869 1353 1115">def update(request, id): try: post = request.form user = User.find(id) user.nama = post['nama'] user.username = post['username'] if post['password'] != "": password = bcrypt.hashpw(post['password'].encode('utf8'), bcrypt.gensalt()) user.password = password user.save() flash('Data berhasil di update!', 'success') return redirect(url_for('user_index')) except Exception as e: return 'Something went wrong ' + str(e)</pre> <p data-bbox="603 1137 970 1227">Hasil Pengujian: Berhasil mengubah data</p>
3	Melakukan hapus data	<pre data-bbox="603 1249 1353 1384">def delete(id): try: delete = User.find(id).delete() flash('Data berhasil di update!', 'success') return redirect(url_for('user_index')) except Exception as e: return 'Something went wrong ' + str(e)</pre> <p data-bbox="603 1406 970 1496">Hasil Pengujian: Berhasil menghapus data</p>
4	Melakukan upload file	<pre data-bbox="603 1518 1353 1944">input_data = list() uploaded_file = request.files['file'] filename = secure_filename(uploaded_file.filename) file_ext = os.path.splitext(filename)[1] if file_ext not in conf['UPLOAD_EXTENSIONS']: flash('Tipe file tidak sesuai!', 'danger') return redirect(url_for('index')) # Upload file to static with new name uploaded_file.save("static/import_data" + file_ext) if file_ext == '.csv': # Read uploaded file df = pd.read_csv("static/import_data.csv") df = df.replace(np.nan, 'EMPTY') elif file_ext == '.xlsx': # Read uploaded file df = pd.read_excel("static/import_data.xlsx") df = df.replace(np.nan, 'EMPTY')</pre>

		<p>Hasil Pengujian:</p> <p>Berhasil melakukan upload dataset</p>
6	Melakukan <i>Preprocessing</i>	<pre>input_data = [] stem_data = [] disable_duplicate = True for index, r in data.iterrows(): print('Processing Index '+str(index)+'... ', end = '') text = r['Tweet'] if text != 'EMPTY': _casefolding = caseFolding(text) _normalisasi = normalisasi_kata(_casefolding) _tokenizing = tokenizing(_normalisasi) _filtering = filtering(_tokenizing) _stemming = stemming(_filtering) _labelling = labelling_lexicon_indonesia(_stemming)</pre> <p>Hasil Pengujian:</p> <p>Berhasil melakukan preprocessing terhadap dataset</p>
7	Implementasi Algoritma SVM	<pre>svm_linear = {'C': [10, 100], 'kernel': ['linear']} svm_poly = {'degree': [1, 2], 'kernel': ['poly']} svm_others = {'C': [10, 100], 'gamma': [0.01, 0.1], 'kernel': ['rbf', 'sigmoid']} parameters = [svm_poly, svm_others, svm_linear] Svm = GridSearchCV(SVC(), param_grid=parameters) # Svm = SVC(C=10, gamma=0.1, kernel='rbf') Svm.fit(X_train_vec, y_train) # Menampilkan kombinasi parameter terbaik yang diperoleh print('Best parameter : ', Svm.best_params_) pred_svm = Svm.predict(X_test_vec)</pre> <p>Hasil Pengujian:</p> <p>Berhasil melakukan klasifikasi</p>
8	Melakukan Evaluasi	<pre>cm_svm = confusion_matrix(y_test, pred_svm, labels=['Positif', 'Netral', 'Negatif']) class_label = ['Positif', 'Netral', 'Negatif'] df_confusion = pd.DataFrame(cm_svm, index = class_label, columns = class_label) sns.heatmap(df_confusion, annot=True, fmt='d', cmap=plt.cm.Blues) plt.title('Confusion Matrix SVM') plt.xlabel('Prediction Label') plt.ylabel('Actual Label') plt.show()</pre> <p>Hasil Pengujian:</p> <p>Berhasil melakukan evaluasi</p>