

```

36 @app.route('/scrap_data/', methods=["GET", "POST"])
37 def halamanScrap():
38     sys.stdout.reconfigure(encoding='utf-8')
39     global komen_user
40     # global post_df_full
41     post_df_full = pd.DataFrame(columns=[])
42
43     komen_user = []
44
45     if request.method == "POST":
46         url_user = request.form['url']
47         string_url = str(url_user)
48         POST_ID = string_url
49         MAX_COMMENTS = 900
50
51         gen = fs.get_posts(
52             post_urls=[POST_ID],
53             cookies='D:\xampp\htdocs\tugas Akhir\Analisa Sentimen\suemangat-TA\yenv\cookies.txt',
54             options={"comments": MAX_COMMENTS, "progress": True}
55         )
56
57         post = next(gen)
58         comments = post['comments_full']
59
60         for comment in comments:
61             print(comment)
62
63             for reply in comment['replies']:
64                 print(' ', reply)
65
66         for comment in comments:
67             post_entry = comment
68             fb_post_df = pd.DataFrame.from_dict(post_entry, orient="index")
69             fb_post_df = fb_post_df.transpose()
70             post_df_full = post_df_full.append(fb_post_df)
71             # print(comment['comment_id'] + ' get')
72
73         post_df_full.to_csv("D:\xampp\htdocs\tugas Akhir\Analisa Sentimen\suemangat-TA\yenv\upload\temp.csv")
74
75         for a in post_df_full['comment_text']:
76             komen_user.append(a)
77
78         MAX_COOKIE_DATA = 55
79         limited_komen = komen_user[:MAX_COOKIE_DATA]
80         session['komen_user'] = limited_komen
81         # session['komen_user'] = komen_user
82         # print(komen_user)
83         komen_user = session.get('komen_user', [])
84
85         komen_dataframe = pd.DataFrame({'comment_text': komen_user})
86         komen_dataframe.to_csv("D:\xampp\htdocs\tugas Akhir\Analisa Sentimen\suemangat-TA\yenv\upload\komen_frame.csv", index=False,
87
88         # Menampilkan hasil scraping
89         post_data = {
90             'post_id': post['post_id'],
91             'text': post['text'],
92             'images': post['image'],
93             # 'shares': post['shares'],
94             'time': post['time']
95         }
96         return render_template('halaman_scrap.html', komen_user = komen_user, post_data = post_data)
97
98         return render_template('halaman_scrap.html', komen_user = komen_user, post_data = {})
99

```

Lampiran 1 Source Code Scrapping

```

180 def klasifikasiAlgo():
181
182     top_words = 7000
183     max_words = 450
184
185     # Read the training dataset
186     df_train = pd.read_excel('D:\xampp\htdocs\tugas Akhir\Analisa Sentimen\suemangat-TA\yenv\upload\DataTrain_baru (1).xlsx')
187
188     print(df_train)
189
190     # Preprocess the training data
191     X_train = df_train["Komentar"].values
192     y_train = df_train["Sentimen"].values
193
194     # Encode the Labels
195     label_encoder = LabelEncoder()
196     y_train = label_encoder.fit_transform(y_train)
197
198     tokenizer = Tokenizer(num_words=top_words)
199     tokenizer.fit_on_texts(X_train)
200
201     X_train = tokenizer.texts_to_sequences(X_train)
202     X_train = pad_sequences(X_train, maxlen = max_words)
203
204     model = Sequential()
205     model.add(Embedding(top_words, 32, input_length = max_words))
206     model.add(Conv1D(128, 5, padding='same', activation='relu'))
207     model.add(MaxPooling1D(5))
208     model.add(Conv1D(128, 5, padding='same', activation='relu'))
209     model.add(MaxPooling1D(5))
210     model.add(Conv1D(128, 5, padding='same', activation='relu'))
211     model.add(MaxPooling1D(5))
212
213     model.add(Flatten())
214     model.add(Dense(250, activation='relu'))
215     model.add(Dense(120, activation='relu'))
216     model.add(Dense(3, activation='softmax'))
217
218     model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
219     model.summary()
220
221     # Train the model
222     t_model = model.fit(X_train, y_train, batch_size=128, epochs=50, verbose=2)
223     # t_model
224
225     # Read the test dataset
226     df_test = pd.read_excel("D:\xampp\htdocs\tugas Akhir\Analisa Sentimen\suemangat-TA\yenv\upload\data_tes(dataset).xlsx")
227
228     # Preprocess the test data
229     X_test = df_test["comment_text"].values
230     X_test = [str(comment) for comment in X_test] # Ubah setiap komentar menjadi string
231
232     # Tokenize the test data
233     X_test = tokenizer.texts_to_sequences(X_test)
234     X_test = pad_sequences(X_test, maxlen=max_words)
235
236     # Predict the sentiment
237     y_test_pred = model.predict(X_test)
238     classes_x = np.argmax(y_test_pred, axis=1)
239
240     # Map the predicted Labels back to the original classes
241     sentiment_mapping = {0: "negatif", 1: "netral", 2: "positif"}
242     y_test_pred_mapped = [sentiment_mapping[pred] for pred in classes_x]
243
244     # Preprocess the test data
245     X_test = df_test["comment_text"].values
246     X_test = [str(comment) for comment in X_test] # Ubah setiap komentar menjadi string
247
248     # Tokenize the test data
249     X_test = tokenizer.texts_to_sequences(X_test)
250     X_test = pad_sequences(X_test, maxlen=max_words)
251
252     # Predict the sentiment
253     y_test_pred = model.predict(X_test)
254     classes_x = np.argmax(y_test_pred, axis=1)
255
256     # Map the predicted Labels back to the original classes
257     sentiment_mapping = {0: "negatif", 1: "netral", 2: "positif"}
258     y_test_pred_mapped = [sentiment_mapping[pred] for pred in classes_x]
259
260     # Add the predicted sentiment column to the test dataset
261     df_test["sentimen"] = y_test_pred_mapped
262
263     # Save the test dataset with predicted sentiment
264     df_test.to_csv("D:\xampp\htdocs\tugas Akhir\Analisa Sentimen\suemangat-TA\yenv\upload\dataset_test_predicted.csv", index=False)

```

Lampiran 2 Source Code Algoritma CNN