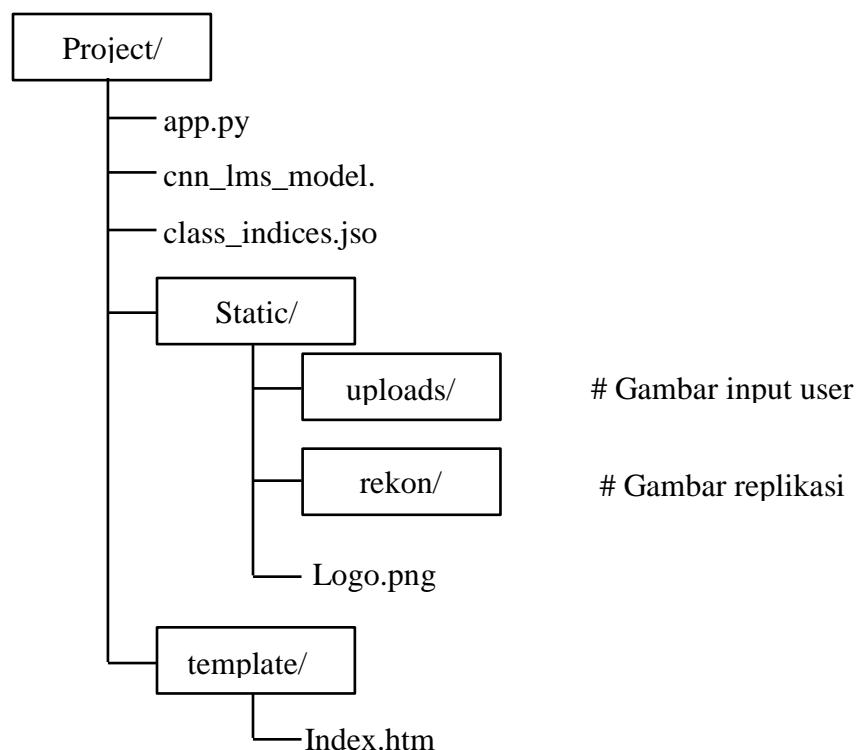


BAB VI

HASIL DAN PEMBAHASAN

4.1. Tampilan Struktur Direktori Program

Struktur direktori merupakan susunan folder yang terorganisasi dan digunakan dalam program agar sistem dapat berjalan dengan baik. Adapun struktur direktori yang digunakan dalam penelitian ini dapat dilihat pada gambar berikut:



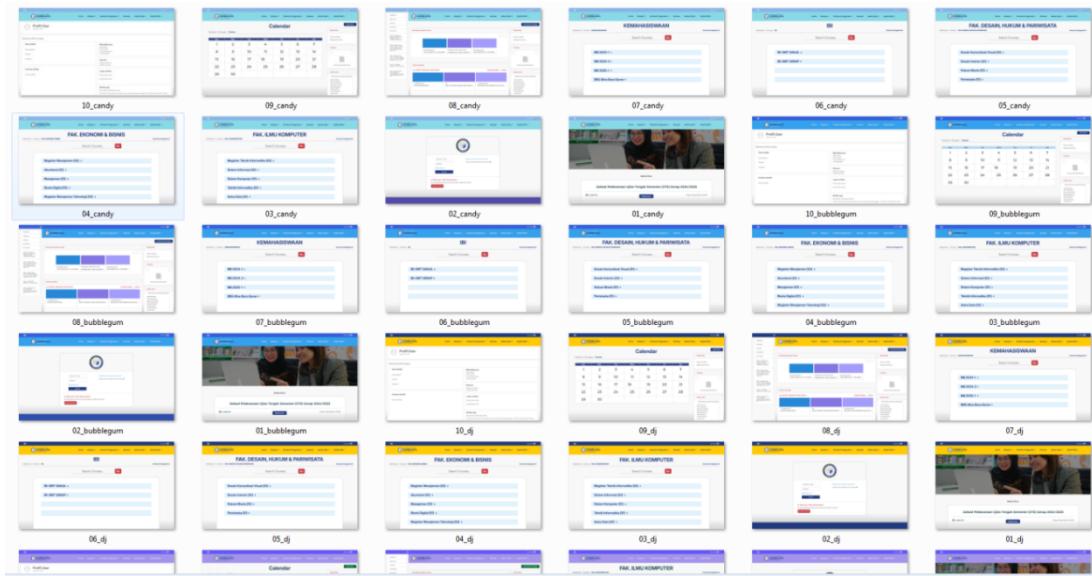
Gambar 4.1. Direktori Folder Sistem

Folder *project* merupakan direktori utama yang memuat seluruh *source code* dan dataset yang digunakan dalam sistem. Di dalam folder tersebut terdapat beberapa berkas penting, antara lain *app.py* sebagai backend sistem, *cnn_lms_model.h5*,

class_indices.json, serta dua folder utama yaitu *static* dan *templates*. Folder *static* berisi dua subfolder, yaitu *uploads* dan *rekon*.

Folder *uploads* digunakan untuk menyimpan gambar yang diunggah pengguna saat sistem dijalankan, sedangkan folder *rekon* berisi hasil rekonstruksi tampilan LMS yang menjadi bagian dari dataset penelitian. Sementara itu, folder *templates* digunakan untuk menyimpan berkas antarmuka (*user interface*) sistem yang akan ditampilkan melalui browser.

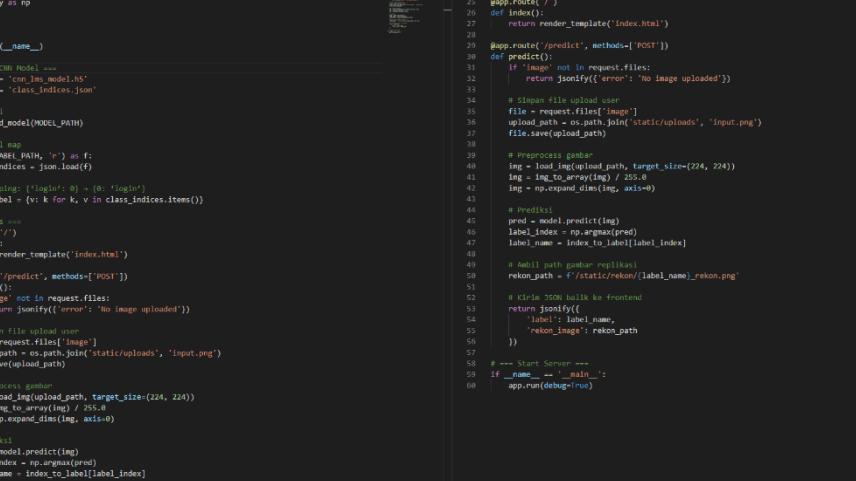
4.2. Tampilan Dataset Program



Gambar 4.2. Tampilan Dataset Program

Program ini dibangun menggunakan dataset primer sebanyak 50 gambar, yang digunakan untuk memperkuat proses pemodelan CNN. Setiap gambar merepresentasikan halaman LMS Darmajaya dengan variasi warna yang berbeda-beda pada setiap page.

4.3. Tampilan Source Code Backend Program



```
 1 #!/usr/bin/env python
 2 # BUKAS PENULISAN > University Semester 8 > SKDIPS > LARCHAN > BUAAYBERNANG > cadangan > project > app.py
 3
 4 from flask import Flask, render_template, request, jsonify
 5 from tensorflow.keras.models import load_model
 6 from tensorflow.keras.preprocessing.image import load_img, img_to_array
 7
 8 import numpy as np
 9
10 import json
11
12 import flask
13
14 app = Flask(__name__)
15
16 # === Load CNN Model ===
17 MODEL_PATH = 'cnn_imodel.h5'
18 LABEL_PATH = 'class_indices.json'
19
20 # Load model
21 model = load_model(MODEL_PATH)
22
23 # Load labels
24 with open(LABEL_PATH, 'r') as f:
25     class_indices = json.load(f)
26
27 # Build mapping: ('image': 0) + (0: 'image')
28 index_to_label = {v: k for k, v in class_indices.items()}
29
30 # === Routes ===
31 @app.route('/')
32 def index():
33     return render_template('index.html')
34
35 @app.route('/predict', methods=['POST'])
36 def predict():
37     if 'image' not in request.files:
38         return jsonify({'error': 'No image uploaded'})
39
40     # Store file uploaded image
41     file = request.files['image']
42     upload_path = os.path.join('static/uploads', 'input.png')
43     file.save(upload_path)
44
45     # Preprocess
46     img = load_img(upload_path, target_size=(224, 224))
47     img = img_to_array(img) / 255.0
48     img = np.expand_dims(img, axis=0)
49
50     # Predict
51     pred = model.predict(img)
52     label_index = np.argmax(pred)
53     label_name = index_to_label[label_index]
54
55     # Ambil path gambar rekласi
56     rekon_path = f'/static/rekon/{label_name}_rekon.png'
57
58     # Kirim JSON bulk ke frontend
59     return jsonify({
60         'label': label_name,
61         'rekon_image': rekon_path
62     })
63
64 # === Start Server ===
65 if __name__ == '__main__':
66     app.run(debug=True)
```

Gambar 4.3. Tampilan Source Code Backend Program

Bahasa pemrograman utama yang digunakan sebagai backend adalah *Python*. Pada sistem ini, *Python versi 3.9.10* digunakan dengan berkas utama backend bernama `app.py`.

4.4. Tampilan Running Program

```
[11:55:45] [User@User-OptiPlex-5090 tensorflow]$ ./tens...py
I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable "TF_ENABLE_ONEDNN_OPTS=0".
I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable "TF_ENABLE_ONEDNN_OPTS=0".
I tensorflow/core/platform/cp...uard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
I tensorflow/core/platform/cp...uard.cc:210] To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:absl:Compiled the loader model, but the compiled metrics have yet to be built. "model.compile_metrics" will be empty until you train or evaluate the model.
WARNING:absl: Debugger is active!
INFO:tensorflow: Debugger PIN: 854-929-173
INFO:werkzeug: Serving Flask app 'app'
INFO:werkzeug: * Debug mode: on
INFO:werkzeug: [31m[!]WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
INFO:werkzeug: * Running on http://127.0.0.1:5000
INFO:werkzeug: [31m[!]WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
INFO:werkzeug: * Restarting with stat
I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable "TF_ENABLE_ONEDNN_OPTS=0".
I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable "TF_ENABLE_ONEDNN_OPTS=0".
I tensorflow/core/platform/cp...uard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
I tensorflow/core/platform/cp...uard.cc:210] To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:absl:Compiled the loader model, but the compiled metrics have yet to be built. "model.compile_metrics" will be empty until you train or evaluate the model.
WARNING:werkzeug: Debugger is active!
INFO:werkzeug: Debugger PIN: 854-929-173

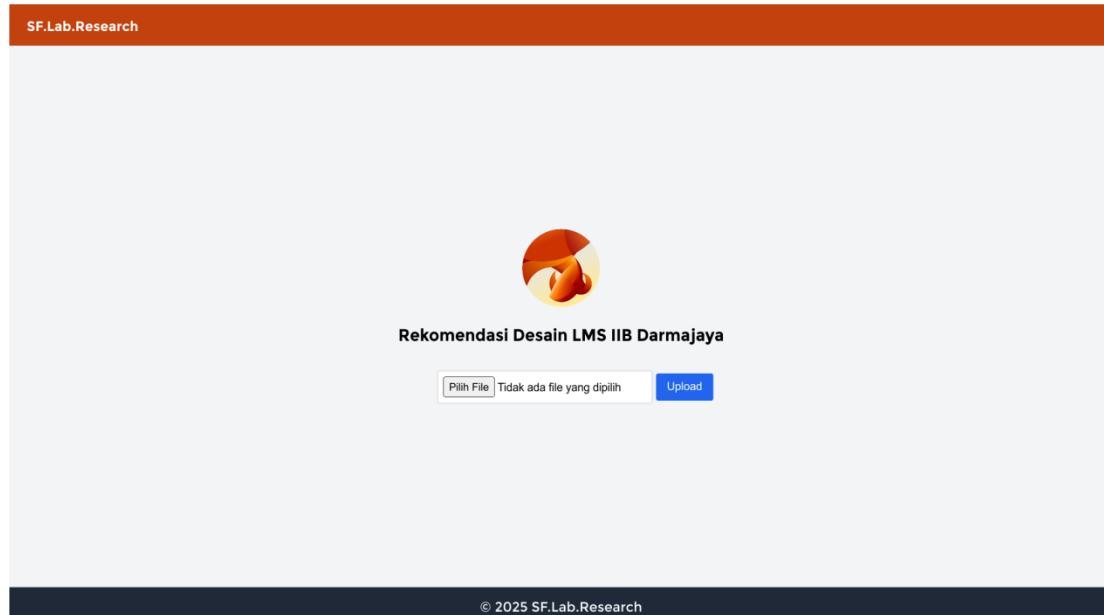
```

Gambar 4.4. Tampilan Running Program

Untuk menjalankan sistem ini, langkah awal yang dilakukan adalah mengeksekusi berkas app.py melalui *Command Prompt* (CMD). Pengguna harus terlebih dahulu mengarahkan direktori ke lokasi penyimpanan berkas app.py, kemudian menjalankan perintah berikut pada CMD : “python app.py

4.5.Tampilan Halaman Utama Program

Tampilan halaman utama program dibangun menggunakan bahasa pemrograman sisi klien, yaitu *HTML*, *CSS*, dan *Tailwind CSS*, sehingga antarmuka dapat ditampilkan langsung melalui browser. Dengan pendekatan ini, sistem dapat berjalan secara lintas platform pada berbagai perangkat dan sistem operasi.

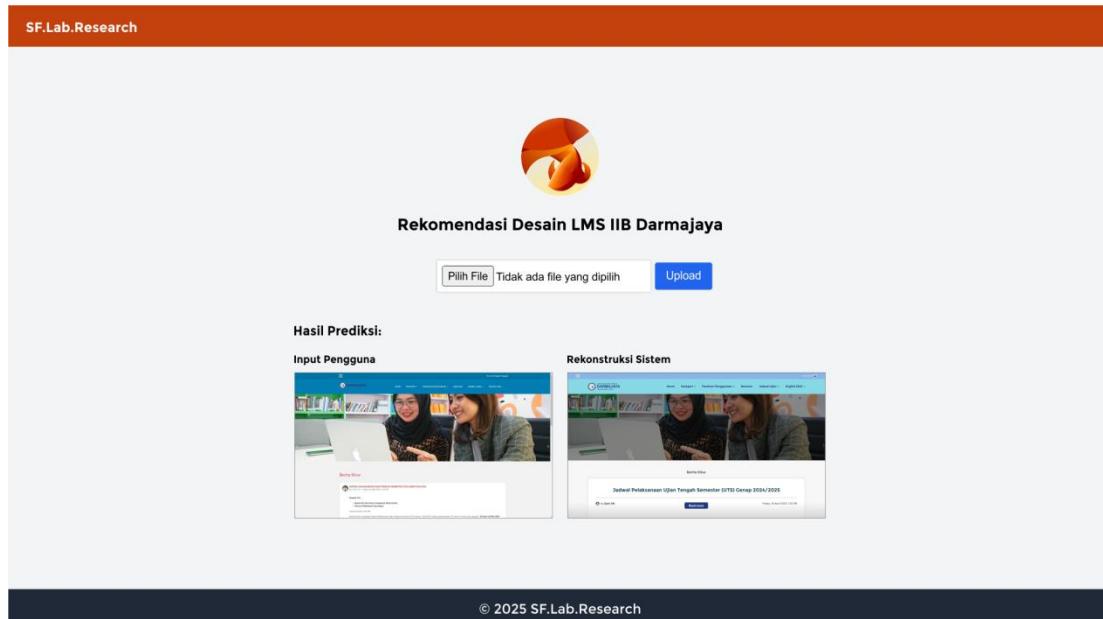


Gambar 4.5.Tampilan Halaman Utama Program

Halaman utama sistem berfungsi sebagai antarmuka pengguna yang memuat elemen-elemen seperti navigasi, logo, label, kolom pemilihan gambar, tombol upload, serta *footer*. Pada halaman ini, pengguna berinteraksi dengan sistem melalui proses pemilihan dan pengunggahan gambar.

4.6.Tampilan Halaman Result Program

Setelah pengguna memilih dan mengunggah gambar, sistem akan menampilkan hasil pemrosesan pada halaman yang sama. Pada bagian ini ditampilkan perbandingan antara gambar input dan output yang dihasilkan oleh program.



Gambar 4.6.Tampilan Halaman Result Program

Halaman result merupakan halaman yang digunakan untuk menampilkan hasil rekonstruksi sistem. Pada halaman ini ditampilkan gambar input yang diunggah pengguna beserta gambar hasil rekonstruksi, sehingga keduanya dapat dibandingkan dan digunakan sebagai umpan balik bagi pengguna.

4.7. Hasil Testing

Testing merupakan tahap pengujian dengan melakukan input gambar berupa screenshot tampilan LMS IIB Darmajaya. Pengujian dilakukan menggunakan data uji yang telah disiapkan sebelumnya dalam bentuk *screenshot* halaman LMS. Adapun gambar yang digunakan sebagai data uji dapat dilihat pada tabel berikut.

Tabel 4.7.1. Data Uji

NO.	Screenshoot	Rename
1	Home	home.png
2	Login	login.png
3	Dashboard	board.png
4	Profil	profil.png
5	Kalender	kalender.png
6	Kategori fakultas ilmu komputer	ilkom.png
7	Kategori fakultas ilmu ekonomi	ekonom.png
8	Kategori fakultas ilmu desain dan hukum	desain.png
9	Kategori kemahasiswaan	mhs.png
10	Kategori IBI Darmajaya	ibi.png

Hasil pengujian sistem ini dievaluasi berdasarkan tingkat ketepatan output yang dihasilkan. Ketepatan pemanggilan gambar tersebut dapat dilihat pada tabel berikut :

Tabel 4.7.2. Tabel Hasil Testing

NO.	Input	Output	Akurasi
1	home.png	home.png	100% (Output Tepat)
2	login.png	ekonom.png	Ekonomi Bisnis
3	board.png	board.png	100% (Output Tepat)
4	profil.png	ekonom.png	Ekonomi Bisnis
5	kalender.png	ekonom.png	Ekonomi Bisnis
6	ilkom.png	ilkom.png	100% (Output Tepat)
7	ekonom.png	ekonom.png	Ekonomi Bisnis
8	desain.png	ekonom.png	Ekonomi Bisnis
9	mhs.png	ekonom.png	Ekonomi Bisnis
10	ibi.png	ekonom.png	Ekonomi Bisnis