

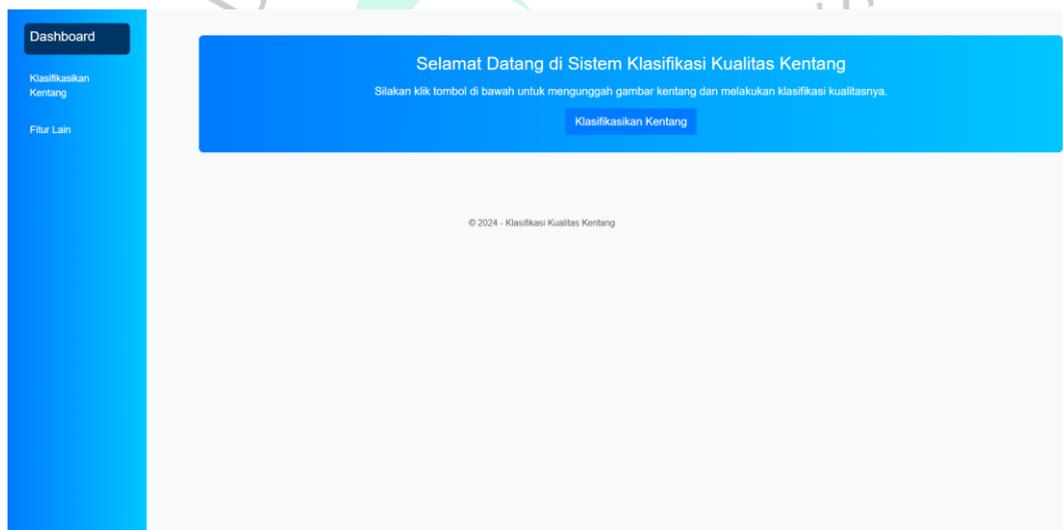
BAB V

HASIL DAN PEMBAHASAN

5.1 Hasil

5.1.1 Hasil Tampilan Antar Muka

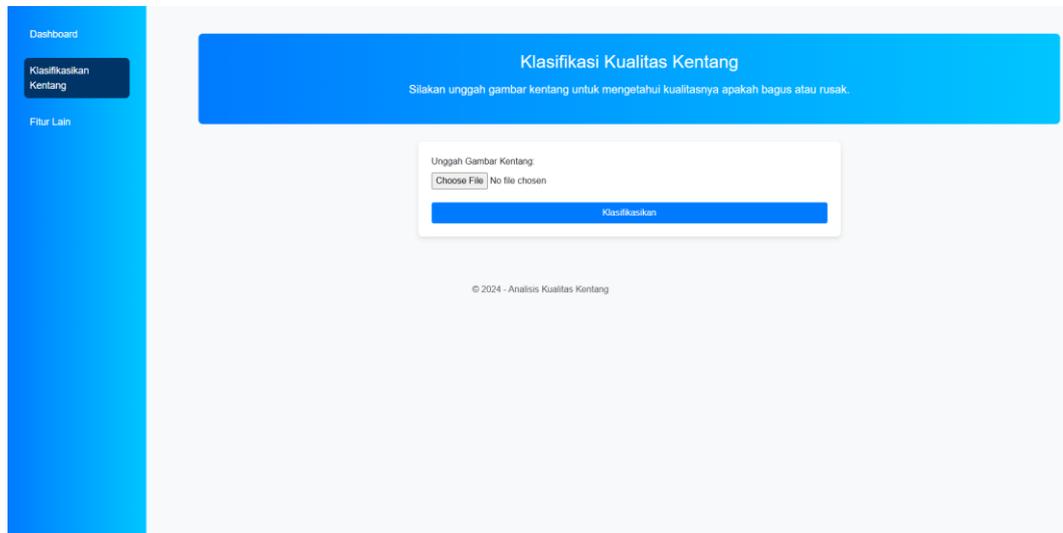
Perancangan antarmuka adalah proses merancang dan mengembangkan tampilan yang akan digunakan untuk memfasilitasi interaksi antara pengguna dan sistem dengan cara yang efisien. Dalam penelitian ini, antarmuka dirancang untuk mendukung aplikasi berbasis *MobileNetV2* yang digunakan untuk mengklasifikasikan kualitas kentang, dengan fitur utama berupa prediksi gambar menggunakan *MobileNetV2*. Berikut ini adalah hasil tampilan antarmuka dari aplikasi.



Gambar 5. 1 Tampilan Halaman Dashboard

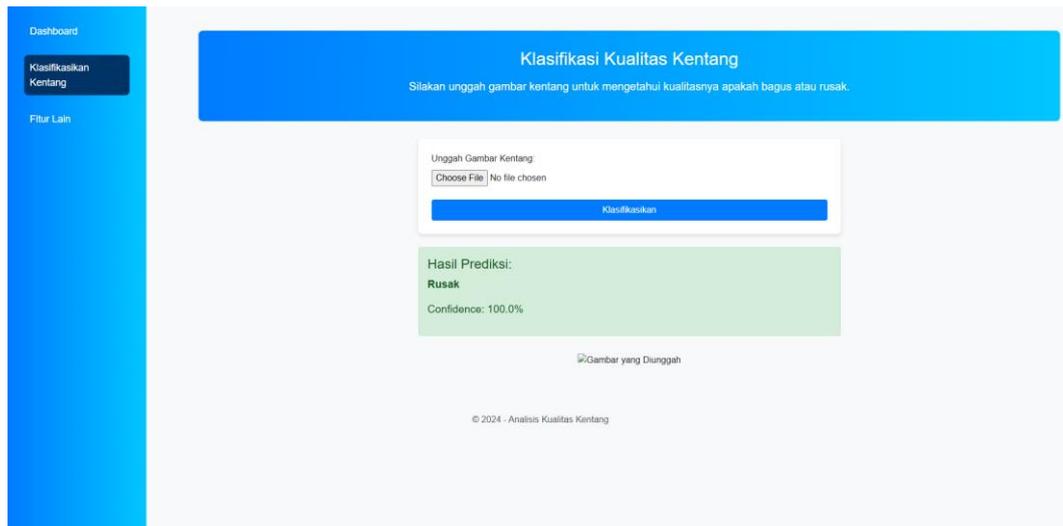
Gambar 5.1, menunjukkan tampilan antarmuka aplikasi yang menyambut pengguna dengan pesan "Selamat Datang di Sistem Klasifikasi Kualitas Kentang" yang jelas di bagian atas halaman. Di bawahnya terdapat tombol "Klasifikasikan Kentang" yang menjadi aksi utama untuk memulai proses pengunggahan gambar kentang dan melakukan klasifikasi kualitasnya. Di sisi kiri terdapat menu navigasi, termasuk opsi Dashboard, Klasifikasikan Kentang, dan Filter Lain, memudahkan pengguna untuk berpindah antar halaman dalam aplikasi. Desain halaman ini menggunakan kombinasi warna biru yang cerah dan sederhana, memberikan

tampilan yang bersih dan mudah dipahami, serta mengutamakan fungsionalitas yang efisien bagi pengguna.



Gambar 5. 2 Tampilan Halaman Klasifikasi

Tampilan antarmuka aplikasi pada gambar 5.2 menunjukkan halaman klasifikasi kualitas kentang, di mana pengguna dapat mengunggah gambar kentang untuk diprediksi kualitasnya, apakah bagus atau rusak. Di bagian atas halaman terdapat judul “Klasifikasi Kualitas Kentang”, dengan instruksi untuk mengunggah gambar kentang. Pengguna dapat memilih gambar kentang melalui tombol “Choose File”, dan setelah memilih gambar, mereka dapat menekan tombol “Klasifikasikan” untuk memulai proses prediksi. Desain ini sederhana dengan tombol yang jelas dan mudah diakses, serta menu navigasi di sisi kiri untuk berpindah antara halaman Dashboard, Klasifikasikan Kentang, dan Filter Lain.



Gambar 5.3 Tampilan Halaman Klasifikasi setelah Penggunaan Mengunggah Gambar

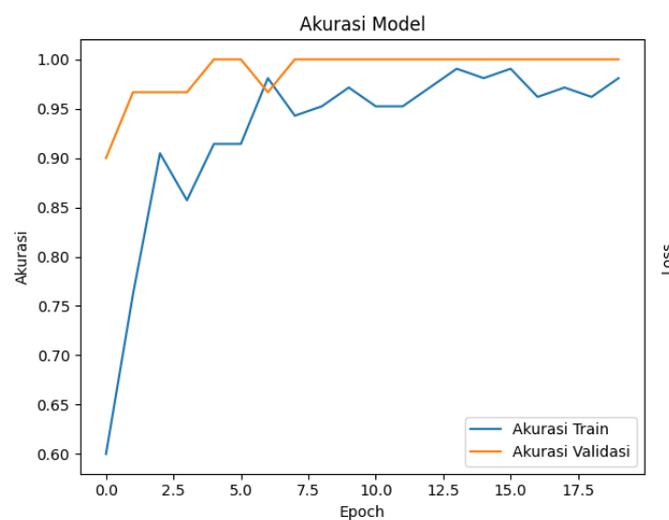
Pada gambar, setelah pengguna mengunggah gambar kentang, aplikasi menampilkan hasil prediksi, yang mencakup kualitas kentang, misalnya "Rusak", dan tingkat *confidence* dari prediksi tersebut.

8.1.2 Hasil Proses Pelatihan Model MobileNetV2

Berikut adalah hasil pelatihan model MobileNetV2:

1. Epoch 1 hingga Epoch 30:
 - Pada epoch pertama, akurasi pada data pelatihan mulai dari sekitar 50%. Ini menunjukkan bahwa pada awal pelatihan, model belum dapat mempelajari pola dengan baik.
 - Namun, seiring berjalannya waktu, pada epoch 30, akurasi pada data pelatihan mulai meningkat dengan tajam dan mencapai sekitar 90%, menunjukkan bahwa model mulai mengenali pola dalam data pelatihan.
 - Di sisi lain, akurasi pada data validasi awal sedikit lebih tinggi dibandingkan dengan pelatihan, tetapi meningkat lebih lambat. Pada epoch 30, akurasi validasi mencapai sekitar 85%, yang menunjukkan bahwa model masih belajar, namun kesulitan untuk menggeneralisasi pada data validasi di awal pelatihan.
2. Epoch 31 hingga Epoch 60:

- Pada epoch 31 hingga 60, model menunjukkan peningkatan yang lebih stabil pada data pelatihan, dengan akurasi mencapai 98% pada epoch ke-60.
 - Pada saat yang sama, akurasi pada data validasi juga terus meningkat. Hal ini mengindikasikan bahwa kemampuan model dalam memahami data pelatihan mengalami peningkatan, kemampuannya untuk menggeneralisasi pada data validasi tidak meningkat secepat pada data pelatihan.
 - Akurasi pada data validasi di epoch ke-60 mencapai sekitar 97%, yang masih menunjukkan kemampuan generalisasi yang baik, meskipun ada sedikit gap antara akurasi pada pelatihan dan validasi.
3. Epoch 61 hingga Epoch 100:
- Pada epoch ke-61 hingga 100, model mencapai akurasi yang sangat tinggi pada data pelatihan, hampir mencapai 100%, dengan hanya sedikit fluktuasi.
 - Sementara itu, akurasi pada data validasi terus stabil pada sekitar 99%, memperlihatkan bahwa model semakin baik pada data pelatihan, selama pelatihan, model kepada data yang tidak terlihat masih mampu menggeneralisasi dengan baik.
 - Fluktuasi kecil pada data validasi antara epoch 70 hingga 90 menunjukkan bahwa model mungkin mengalami sedikit kesulitan untuk mengklasifikasikan data validasi secara konsisten. Namun, hal ini tidak mengindikasikan overfitting yang signifikan.



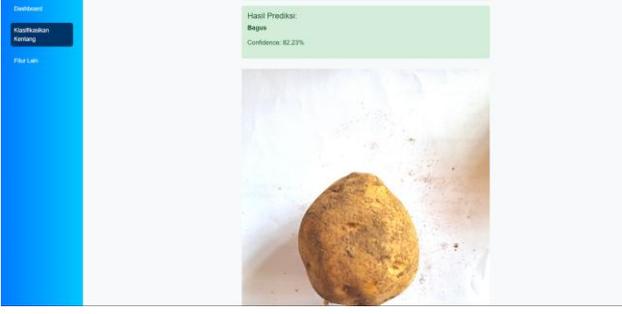
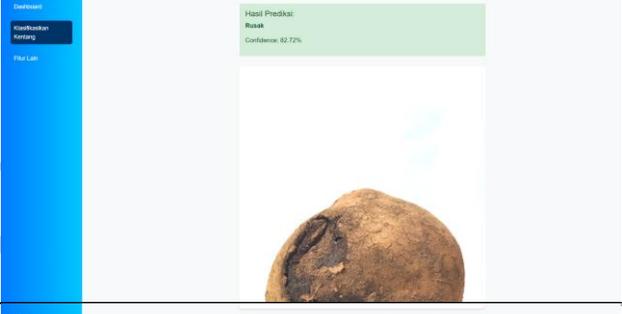
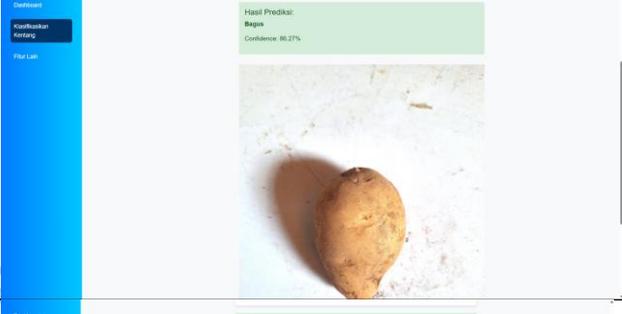
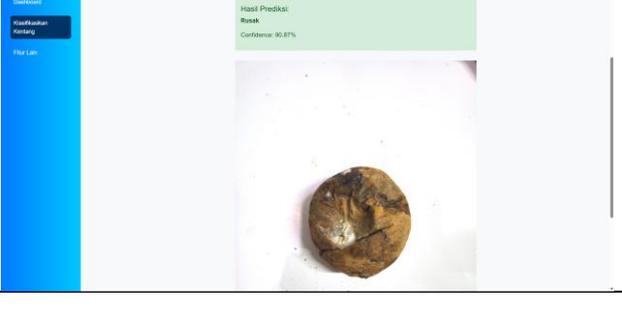
Gambar 5.4 Hasil Proses Pelatihan Model Accuracy MobileNetV2

Pada gambar 5.4, menunjukkan peningkatan yang signifikan dalam akurasi data pelatihan (train), yang mencapai 100%. Peningkatan yang tajam pada akurasi data pelatihan menunjukkan model memiliki kemampuan untuk mengenali pola dalam data pelatihan dan mengklasifikasikan gambar dengan sangat akurat. Lalu, untuk akurasi pada data validasi (val) juga mencapai sekitar 99% pada epoch terakhir, yang mengindikasikan bahwa model mampu memahami pola dari data pelatihan dan tetap menunjukkan performa tinggi saat diuji menggunakan data yang belum pernah digunakan sebelumnya. Meskipun demikian, terdapat fluktuasi kecil pada garis validasi pada beberapa epoch, yang mungkin menandakan adanya sedikit overfitting, meskipun model secara keseluruhan masih dapat mengklasifikasikan data validasi dengan sangat baik. Secara keseluruhan, model berhasil mengatasi overfitting dan menunjukkan kemampuan generalisasi yang baik pada data validasi, dengan akurasi yang tetap stabil dan tinggi.

5.1.3 Hasil Pengujian

Pada tahap ini, menunjukkan hasil pengujian aplikasi untuk menentukan kualitas kentang

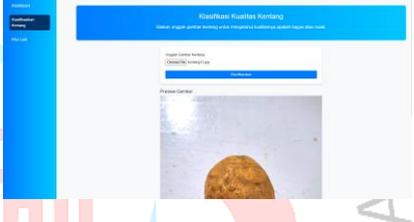
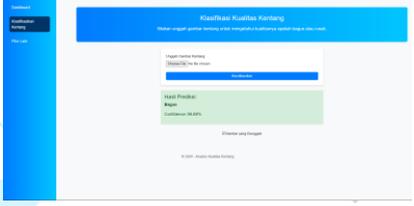
Tabel 5. 1 Hasil Pengujian

Gambar	Hasil Prediksi	Keterangan
 <p data-bbox="316 607 485 640">Kentang Bagus</p>		Berhasil
 <p data-bbox="316 943 485 976">Kentang Rusak</p>		Berhasil
 <p data-bbox="316 1279 485 1312">Kentang Bagus</p>		Berhasil
 <p data-bbox="316 1570 485 1603">Kentang Rusak</p>		Berhasil

5.1.4 Pengujian Black Box

Pada tahap ini, ditampilkan hasil pengujian *Black Box* terhadap aplikasi klasifikasi kualitas kentang sebagaimana disajikan pada Tabel 5.2.

Tabel 5. 2 Pengujian *black Box*

No	Pengujian	Hasil yang diharapkan	Hasil Pengamatan	Kesimpulan
1.	Pengguna membuka aplikasi untuk menentukan kualitas kentang	Menampilkan halaman dashboard		Berhasil
2.	Pengguna memilih halaman klasifikasi	Menampilkan halaman klasifikasi		Berhasil
3.	Pengguna mengunggah file	Dapat mengunggah file dan menampilkan <i>preview</i> gambar		Berhasil
4.	Pengguna mendapatkan hasil prediksi	Menampilkan hasil prediksi		Berhasil

5.1.5 Pengujian *White Box*

Pada tahap ini, hasil pengujian *White Box* terhadap aplikasi klasifikasi kualitas kentang ditampilkan pada Tabel 5.3 dan Tabel 5.4.

Tabel 5. 3 Pengujian *White Box*

No	Hasil yang diharapkan	Kode Program	Hasil Pengujian
1	Aplikasi berhasil mengkonfigurasi folder upload dan model dengan benar.	os.makedirs(UPLOAD_FOLDER, exist_ok=True) model = load_model(MODEL_PATH)	Berhasil: Folder upload ada, model berhasil dimuat tanpa error.
2	Fungsi <code>allowed_file</code> memeriksa ekstensi file dengan benar.	def allowed_file(filename): return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS	Berhasil: Hanya file dengan ekstensi png, jpg, jpeg yang diterima.
3.	Gambar berhasil disimpan di folder upload dengan nama unik.	file.save(file_path)	File gambar berhasil disalin ke folder <code>UPLOAD_FOLDER</code> dengan nama yang benar.
4.	Saat pengguna mengunggah gambar, sistem akan melakukan preprocessing.	def preprocess_image(img_path): img = cv2.imread(img_path) img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) img = cv2.resize(img, (224, 224)) # Ukuran input model img = img / 255.0 # Normalisasi gambar return np.expand_dims(img, axis=0)	Berhasil: Gambar dapat diproses
5.	Model menghasilkan prediksi yang benar berdasarkan gambar input	pred = model.predict(img) prediction = CLASS_NAMES[np.argmax(pred)]	Berhasil: Model memberikan hasil prediksi yang sesuai dengan kelas yang benar
6.	Fungsi <code>preprocess_image</code> menangani kesalahan dengan baik.	try: img = preprocess_image(file_path) except Exception as e:	Berhasil: Sistem dapat menangani kesalahan jika file tidak dapat dibaca atau diproses.

Tabel 5.4 Pengujian *White Box* Lanjutan

No	Hasil yang diharapkan	Kode Program	Hasil Pengujian
7.	Aplikasi menampilkan pesan error jika format file tidak didukung.	<pre>if file and allowed_file(file.filename): ... else: error_message = "Format file tidak didukung..."</pre>	Berhasil: Pesan error muncul jika format file selain png, jpg, atau jpeg diunggah.
8.	Model memberikan probabilitas prediksi dalam persen.	<pre>confidence = round(float(np.max(pred)) * 100, 2)</pre>	Berhasil: confidence menampilkan nilai prediksi dalam bentuk persen.

5.2 Pembahasan

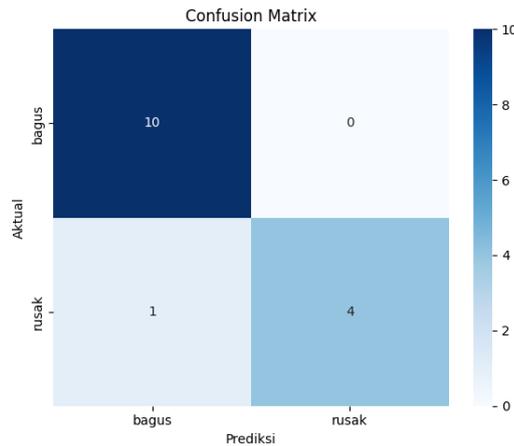
5.2.1 Tampilan Antar Muka

Hasil tampilan antarmuka aplikasi untuk menentukan kualitas kentang menunjukkan desain yang minimalis, namun tetap *user-friendly* dan mudah dipakai oleh pengguna.

5.2.2 Proses Pelatihan Model MobileNetV2

Secara keseluruhan, meskipun terdapat peningkatan tajam akurasi pada data pelatihan, model mampu mempertahankan kemampuan generalisasi yang baik pada data validasi. Ini menunjukkan bahwa model telah belajar secara efektif tanpa terjebak dalam overfitting, yang merupakan hasil yang baik untuk klasifikasi kualitas kentang dalam dataset ini.

5.2.3. Hasil Evaluasi Model



Gambar 5. 6 Hasil Evaluasi Model

Confusion matrix pada gambar 5.6, menunjukkan hasil prediksi model terhadap dua kelas, yaitu bagus dan rusak. Dari matriks ini, dapat dilihat bahwa model memprediksi dengan benar 10 sampel sebagai bagus (*True Positive*), dan tidak ada kesalahan dalam memprediksi kelas bagus sebagai rusak (*False Negative*). Di sisi lain, model memprediksi dengan benar 4 sampel sebagai rusak (*True Negative*), dan hanya terdapat 1 kesalahan dalam memprediksi kelas rusak sebagai bagus (*False Positive*). Secara keseluruhan, model memiliki kinerja yang sangat baik ditunjukkan oleh *confusion matrix*, dengan tingkat akurasi yang bagus dalam memprediksi kedua kelas, meskipun ada sedikit kesalahan dalam memprediksi kelas rusak.

Tabel 5. 5 Hasil Evaluasi Kinerja Model Klasifikasi

No	Metrik	Nilai
1.	Akurasi	Total: 93,33%
2.	<i>Precision</i>	Bagus: 90,91%

		Rusak: 100%
3.	<i>Recall</i>	Bagus: 100%
		Rusak: 80%
4.	<i>F1-Score</i>	Bagus: 95,24%
		Rusak: 88,89%

Berdasarkan tabel 5.5, *accuracy* yang diperoleh adalah 93.33%, yang menunjukkan bahwa model secara keseluruhan memiliki performa yang sangat baik. *Precision* untuk kelas bagus mencapai 90,91%, yang menandakan bahwa model tidak melakukan kesalahan dalam mengklasifikasikan kelas "bagus" sebagai "rusak", lalu untuk *precision* kelas rusak mencapai 100% yang artinya model masih melakukan kesalahan dalam mengklasifikasi kelas "rusak" menjadi kelas "bagus". Untuk kelas bagus, *recall* mencapai 100%. Lalu untuk kelas rusak, *recall* mencapai 80% yang berarti model dapat mengidentifikasi semua sampel "rusak" dengan baik. *F1-Score* untuk kelas bagus adalah 95.24%, sementara untuk kelas rusak adalah 88.89%, yang menunjukkan bahwa meskipun model mampu mengidentifikasi kelas "bagus", terdapat sedikit penurunan dalam kinerja untuk kelas "rusak". Secara keseluruhan, model ini memperlihatkan akurasi yang tinggi dan mampu mengklasifikasikan kedua kelas dengan baik, meskipun ada sedikit penurunan pada kelas "rusak".