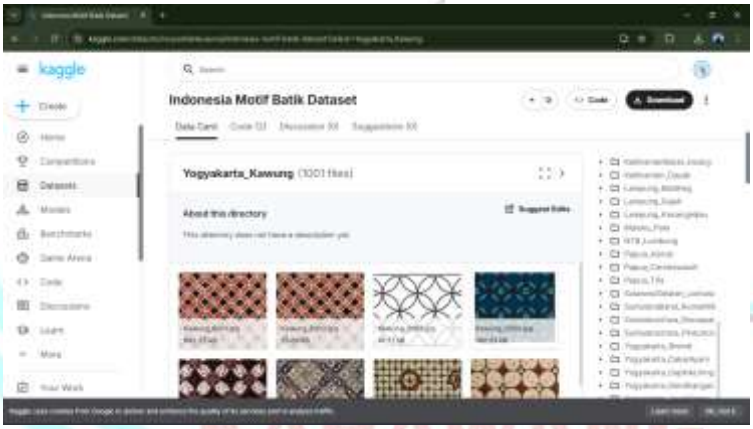
BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini adalah citra motif batik tulis tradisional Indonesia yang digunakan sebagai bahan untuk proses identifikasi dan klasifikasi motif secara otomatis. Batik dipilih karena memiliki karakter visual yang kompleks, terdiri dari pola berulang, garis halus, dan ornamen khas yang berbeda pada setiap daerah namun sering kali terlihat mirip. Kesulitan visual antar-motif ini menyebabkan proses pengenalan motif batik termasuk dalam kategori *fine-grained image classification*, sehingga menuntut metode yang mampu mengenali perbedaan detail secara lebih sensitif.



Gambar 3. 1 Tampilan Dataset Motif Batik Indonesia pada Platform Kaggle

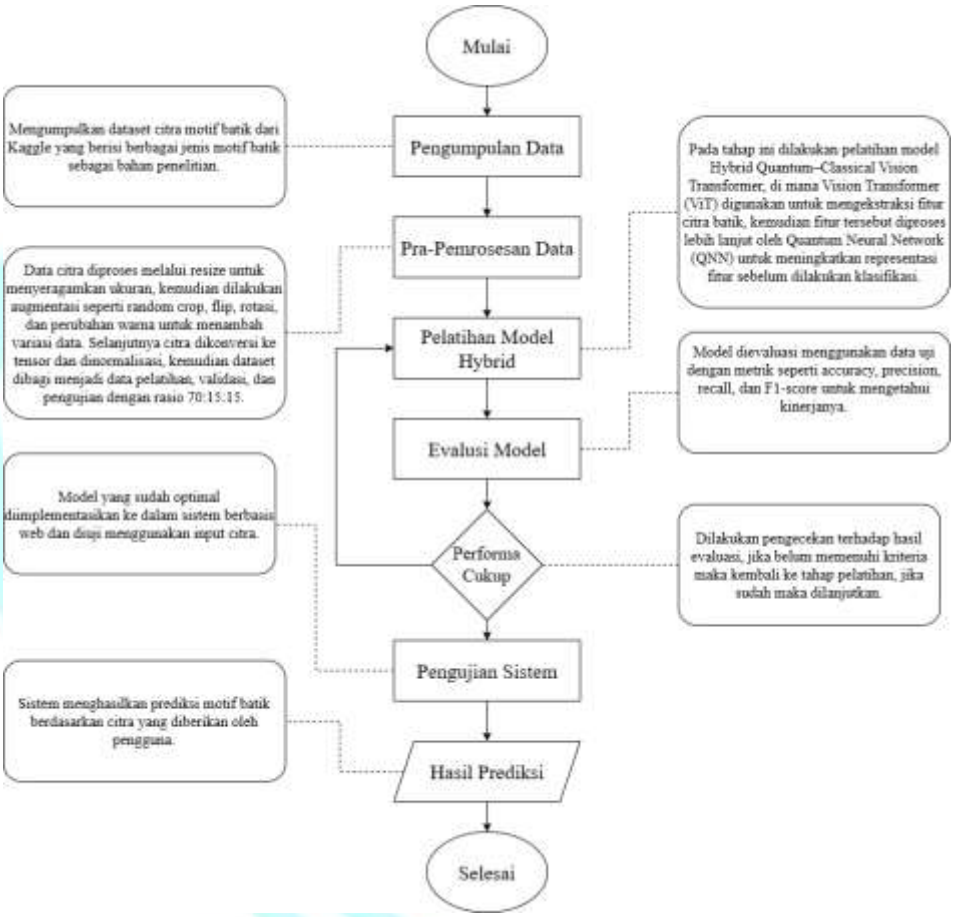
Sumber : https://www.kaggle.com/datasets/ravyardiankusuma/indonesia-motif-batik-dataset

Dataset yang digunakan pada penelitian ini adalah Indonesian Batik Motif *Dataset* oleh Ravy Ardian Kusuma yang berasal dari *Platform Kaggle*, yang berisi lebih dari seratus kategori motif batik dari berbagai daerah seperti Jawa Tengah, Jawa Barat, Betawi, Bali, dan lainnya. *Dataset* ini dipilih karena memiliki keragaman motif yang tinggi, resolusi citra yang cukup baik, serta jumlah data yang besar (±111.000 citra), sehingga representatif untuk mengevaluasi kemampuan Model dalam membedakan motif yang sangat mirip.

3.2 Prosedur Penelitian

Prosedur Prosedur penelitian ini disusun untuk menggambarkan alur kerja penelitian secara berurutan, mulai dari pengumpulan data hingga evaluasi Model

Hybrid Quantum–Classical Vision Transformer dan penerapannya pada sistem berbasis web. Alur penelitian dibuat agar mudah dipahami dan dapat diikuti kembali oleh peneliti lain. Gambaran umum tahapan penelitian ditunjukkan pada Gambar 3.2.



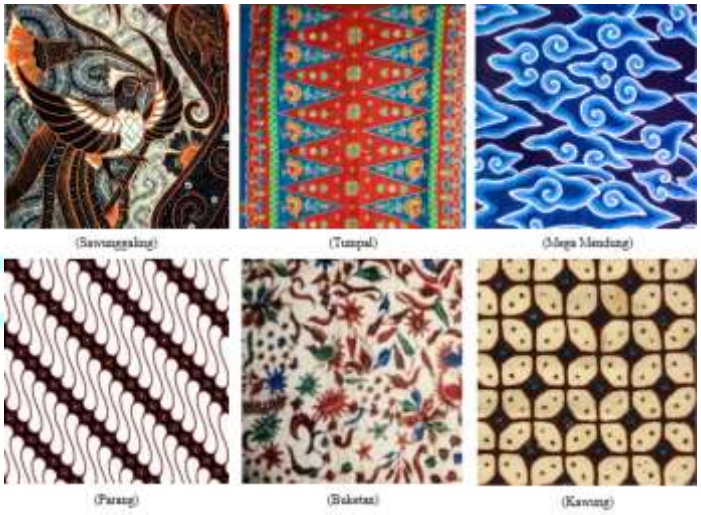
Gambar 3. 2 Alur Diagram Prosedur Penelitian

Gambar tersebut menggambarkan rangkaian aktivitas mulai dari pengumpulan dan pemrosesan *Dataset*, pengembangan dan pelatihan Model *hybrid*, *evaluasi performa*, hingga implementasi sistem klasifikasi batik berbasis web sebagai *proof-of-concept*.

3.2.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah Motif Batik Indonesia *Dataset* yang diperoleh dari *repositori* terbuka *Kaggle* dan dikurasi oleh Ravy Ardian Kusuma. *Dataset* ini terdiri dari lebih dari 111 kategori motif batik yang merepresentasikan berbagai daerah di Indonesia, dengan total jumlah citra lebih dari 111.000 gambar. Setiap kategori motif disimpan dalam satu folder terpisah, di

mana satu folder merepresentasikan satu jenis motif batik, sehingga struktur *Dataset* memudahkan proses pelabelan dan pengolahan data untuk keperluan klasifikasi.



Gambar 3. 3 *Dataset* Citra Batik

Sumber : Diankusuma, Ravyard I. Indonesia Motif Batik *Dataset*. Kaggle, 2023.

Gambar 3.3, menampilkan contoh citra motif batik dari beberapa daerah di Indonesia yang terdapat dalam *Dataset*, seperti Jawa Barat, Jawa Tengah, Yogyakarta, Bali, Betawi, Cirebon, dan Pekalongan, yang menunjukkan keberagaman pola, ornamen, dan tingkat kompleksitas visual antar motif. Citra-citra batik tersebut selanjutnya digunakan sebagai data masukan untuk tugas klasifikasi motif batik berdasarkan jenis atau asal motifnya.

3.2.2 Pra-Pemrosesan Data

Tahap pra-pemrosesan data dilakukan untuk menyiapkan citra motif batik agar memiliki format dan kualitas yang seragam sebelum digunakan sebagai masukan ke dalam Model *Hybrid Quantum–Classical*. Tahap ini diperlukan karena motif batik memiliki banyak detail visual, seperti garis-garis halus dan isen-isen, sehingga citra perlu disesuaikan terlebih dahulu agar proses pembelajaran dan ekstraksi fitur dapat berjalan dengan baik.

1. *Resize*

   Citra motif batik diubah ukurannya menjadi dimensi yang lebih besar dari ukuran input model, yaitu sekitar 225 × 255 piksel. Tujuan dari tahap ini adalah untuk menyeragamkan ukuran citra sekaligus menyediakan ruang bagi proses augmentasi, khususnya pada tahap *cropping*. Dengan ukuran yang lebih besar,

citra masih memiliki area yang cukup sehingga proses pemotongan acak tidak menghilangkan bagian penting dari motif batik. Contoh gambar resize dapat dilihat pada Gambar 3.4 yang mana gambar a merupakan sebelum *resize* dan gambar b merupakan gambar setelah *resize* .



(a) **(b)**

Gambar 3. 4 Contoh *Resize* Sebelum dan Sesudah

Sumber : GeeksforGeeks. (2024). Image Resizing using OpenCV in Python.

2. Augmentasi

Augmentasi data dilakukan menggunakan metode *online augmentation* (augmentasi *on-the-fly*) melalui pustaka *torchvision.transforms* dari *PyTorch*. Berbeda dengan augmentasi offline yang menyimpan gambar hasil transformasi secara fisik ke disk, metode ini menerapkan transformasi secara otomatis pada setiap *epoch* saat proses pelatihan berlangsung. Dengan demikian, jumlah gambar dalam dataset tidak bertambah secara fisik, namun model secara efektif melihat variasi visual yang berbeda-beda pada setiap *epoch*, sehingga membantu mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi model.

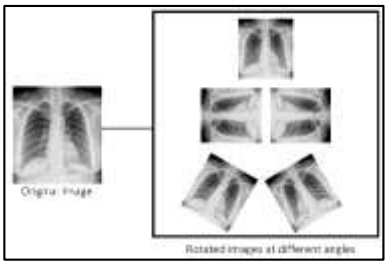
Augmentasi hanya diterapkan pada data pelatihan (*train_transforms*), sedangkan data validasi dan pengujian (*val_transforms*) hanya melalui proses *CenterCrop* dan normalisasi tanpa transformasi acak.



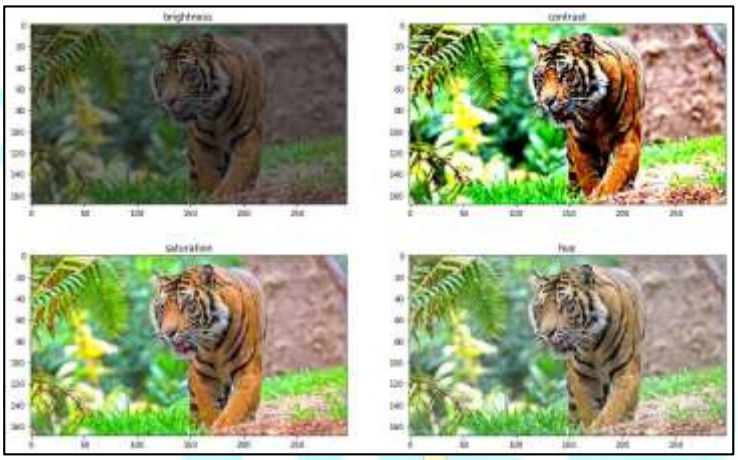
(a) **(b)**



(c)



(d)



(e)

Gambar 3. 5 Contoh Contoh Proses Augmentasi

Sumber : *PyTorch*. (2024). *Torchvision Transforms*

Berdasarkan Gambar 3.5 merupakan contoh-contoh augmentasi data yang dapat di jelaskan sebagai berikut :

a. *RandomCrop* : mengambil bagian acak dari citra sebagai ukuran akhir input model.
b. *Horizontal Flip* : membalik citra secara horizontal.
c. *Vertical Flip* : membalik citra secara vertical.
d. *Random Rotation* : memutar citra dalam rentang sudut tertentu.
e. *Color Jitter* : mengubah kecerahan, kontras, dan saturasi citra.

RandomCrop berperan penting dalam menentukan ukuran akhir citra yang digunakan sebagai input model, yaitu 224 × 224 piksel.

3. Konversi ke *Tensor* (*ToTensor*)

Setelah melalui proses augmentasi, citra motif batik dikonversi menjadi format *Tensor* menggunakan ToTensor(). Tahap ini diperlukan karena model *Hybrid Quantum-Classical Vision Transformer* (HQCViT) berbasis *PyTorch* hanya

dapat memproses data dalam bentuk numerik bertipe *Tensor*, bukan format gambar biasa. Pada proses konversi ini, nilai piksel citra diubah dari rentang 0–255 menjadi 0–1 agar skala nilainya seragam dan siap untuk diproses lebih lanjut oleh komponen *Vision Transformer* maupun lapisan kuantum (*Quantum Neural Network*) dalam arsitektur model.

Selanjutnya dilakukan normalisasi menggunakan nilai *mean* = [0.5, 0.5, 0.5] dan *standard deviation* = [0.5, 0.5, 0.5] pada setiap *channel* warna (RGB), sehingga nilai piksel diubah menjadi rentang -1.0 hingga 1.0 dengan distribusi terpusat di sekitar nol. Normalisasi ini penting dalam konteks klasifikasi motif batik karena citra batik memiliki variasi warna dan kecerahan yang tinggi antar motif maupun antar daerah asal. Dengan nilai piksel yang sudah terpusat dan berskala seragam, model dapat belajar lebih stabil, proses pelatihan menjadi lebih cepat konvergen, serta kemampuan model dalam membedakan detail halus antar motif batik dapat lebih optimal.

4. Pembagian Data

Dataset citra motif batik yang telah di proses dibagi menjadi data latih, data validasi, dan data uji sebagaimana ditunjukkan pada Tabel 3.2, yang selanjutnya digunakan dalam pengembangan dan pelatihan Model *Hybrid Quantum–Classical*.

Tabel 3. 1 Pembagian *Dataset*

No	Subset Data	Presentase
1.	Data Pelatihan (*Training*)	70%
2.	Data Validasi (*Validation*)	15%
3.	Data Pengujian (*Testing*)	15%

Tabel 3.2 menunjukkan pembagian dataset penelitian menjadi data pelatihan, validasi, dan pengujian dengan rasio 70%, 15%, dan 15% yang dibagi secara merata pada setiap kelas motif batik.Proporsi pembagian 70:15:15 dipilih karena memberikan keseimbangan antara jumlah data pelatihan yang cukup besar untuk mempelajari pola secara optimal dan data validasi serta pengujian yang cukup untuk evaluasi yang representatif. Pada dataset berskala besar seperti penelitian

ini, pembagian tersebut umum digunakan dan mampu menghasilkan evaluasi model yang stabil serta minim bias (Sathyanarayanan, 2024).

3.2.3 Perancangan Dan Pelatihan Model

Perancangan model pada penelitian ini mengembangkan arsitektur Hybrid Quantum–Classical yang mengintegrasikan Vision Transformer (ViT) sebagai komponen klasik dan Quantum Neural Network (QNN) sebagai komponen kuantum dalam satu alur pembelajaran end-to-end untuk klasifikasi motif batik tulis tradisional. Arsitektur ini terdiri dari dua tahap utama, yaitu ekstraksi fitur klasik dan klasifikasi kuantum, yang bekerja secara terpadu membentuk satu sistem saraf hibrida.

1. Perancangan dan Integrasi Model Hybrid (ViT–QNN)

   Tahap ini berfokus pada pengembangan arsitektur model *Hybrid Quantum–Classical* yang mengintegrasikan *Vision Transformer* (ViT) sebagai komponen klasik dan *Quantum Neural Network* (QNN) sebagai komponen kuantum dalam satu alur pembelajaran *end-to-end* untuk klasifikasi motif batik tulis tradisional.

   a. Ekstraksi Fitur Klasik

      Bagian klasik model menggunakan *Vision Transformer* (ViT) sebagai *feature extractor* untuk mengekstraksi representasi visual citra motif batik. Citra masukan diproses melalui *patch Embedding* dan *Encoder transformer* (MHSA dan FFN), kemudian representasi global citra diambil dari token [CLS] dan diproyeksikan ke 4 fitur agar sesuai dengan jumlah *Qubit* pada bagian kuantum.

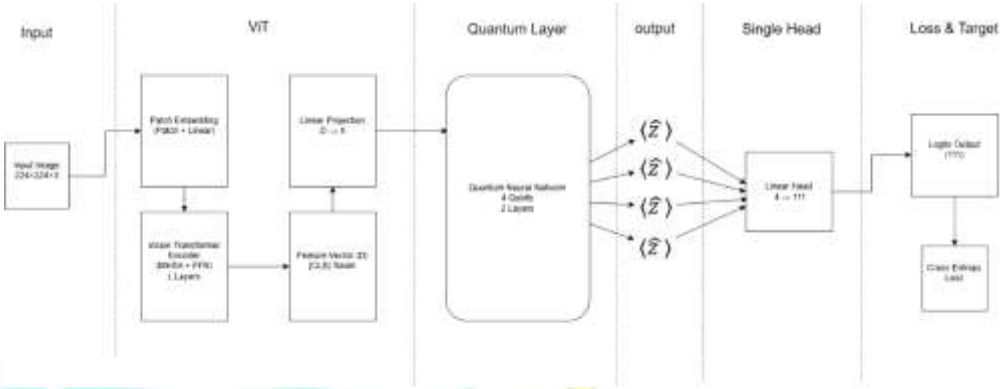
   b. Klasifikasi Kuantum

      Bagian kuantum model dikembangkan menggunakan *PennyLane* dengan 4 *Qubit*, di mana fitur hasil proyeksi dienkode menggunakan *angle Embedding* dan diproses oleh *variational quantum circuit* (VQC). Keluaran QNN diperoleh melalui pengukuran nilai harapan operator *Pauli-Z* pada setiap *Qubit* sebagai representasi fitur kuantum sebelum klasifikasi.

   c. Integrasi Model Hybrid

      Komponen *Vision Transformer* dan *Quantum Neural Network* diintegrasikan menjadi satu model hybrid *end-to-end* menggunakan mekanisme integrasi antara *PyTorch* dan *PennyLane*. *Output* dari QNN berupa vektor fitur

kuantum selanjutnya diteruskan ke lapisan klasifikasi berupa *single linear head* yang memetakan 8 fitur kuantum ke dalam 111 kelas motif batik.



Gambar 3. 6 Arsitektur *Hybrid Quantum–Classical* ViT

Seluruh komponen model, mulai dari *Vision Transformer*, lapisan proyeksi fitur, *Quantum Neural Network* , hingga *classifier*, dilatih secara bersamaan dalam satu proses optimasi menggunakan fungsi *Loss Cross-Entropy*. Integrasi ini memungkinkan aliran gradien dari lapisan klasifikasi hingga ke bagian kuantum dan klasik, sehingga model dapat belajar secara terpadu.

2. Pelatihan model

Setelah arsitektur model dirancang dan diintegrasikan, proses pelatihan dilakukan menggunakan data latih yang telah disiapkan pada tahap sebelumnya. Pelatihan dilakukan secara *end-to-end*, di mana parameter pada komponen klasik dan kuantum diperbarui secara bersamaan dalam satu proses optimasi. Fungsi *Cross-Entropy Loss* digunakan untuk mengukur kesalahan prediksi model terhadap label kelas motif batik.

Proses optimasi parameter dilakukan menggunakan *optimizer Adam* untuk meminimalkan nilai *Loss* selama pelatihan. Kinerja model selama proses pelatihan dipantau menggunakan data validasi untuk memastikan model belajar secara stabil dan tidak mengalami *overfitting*. Hasil dari tahap ini adalah model *Hybrid Quantum–Classical Vision Transformer* yang telah terlatih dan siap digunakan pada tahap pengujian dan evaluasi kinerja model.

3.2.4 Evaluasi Model

Tahap evaluasi model dilakukan untuk mengukur kinerja model *Hybrid Quantum–Classical Vision Transformer* dalam mengklasifikasikan motif batik tulis tradisional. Evaluasi dilakukan setelah proses pelatihan selesai dengan

menggunakan data uji yang tidak terlibat dalam proses pelatihan maupun validasi. Pendekatan evaluasi yang digunakan dalam penelitian ini terdiri dari dua bagian utama, yaitu metrik evaluasi global dan analisis kendala teknis.

1. Metrik Evaluasi

   Evaluasi global dilakukan untuk menilai performa model secara kuantitatif menggunakan metrik klasifikasi standar pada permasalahan multi-kelas. Metrik yang digunakan meliputi:

   a. *Accuracy*, untuk mengukur ketepatan prediksi secara keseluruhan.
   b. *Precision*, untuk mengukur ketepatan prediksi pada setiap kelas motif batik.
   c. *Recall*, untuk mengukur kemampuan model dalam mengenali seluruh sampel aktual pada setiap kelas.
   d. *F1-Score*, untuk menilai keseimbangan antara *Precision* dan *Recall*, Khususnya pada kondisi distribusi kelas yang tidak sepenuhnya seimbang.

   Selain itu, digunakan juga *confusion matrix* untuk memvisualisasikan hasil prediksi model terhadap label sebenarnya. *Confusion matrix* membantu dalam mengidentifikasi distribusi kesalahan klasifikasi antar kelas serta mengetahui kelas mana yang memiliki tingkat kesalahan tertinggi.

2. Analisis Kendala Teknis

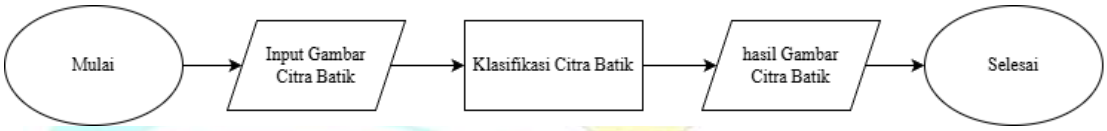
   Selain evaluasi kuantitatif, penelitian ini juga melakukan analisis kendala teknis untuk mengidentifikasi faktor-faktor yang mempengaruhi kinerja model, khususnya pada kasus kesalahan klasifikasi. Analisis ini dilakukan secara kualitatif dengan mengamati pola kesalahan yang terjadi pada hasil prediksi model.

3.2.5 Pengembangan dan Pengujian Sistem

Pengembangan sistem pada penelitian ini berupa sistem klasifikasi motif batik berbasis web yang digunakan untuk mengintegrasikan Model *Hybrid Quantum–Classical Vision Transformer* ke dalam aplikasi yang dapat diakses oleh pengguna. Sistem dikembangkan menggunakan *framework Flask* sebagai *Backend* karena bersifat ringan, *fleksibel*, dan mudah diintegrasikan dengan Model pembelajaran mesin berbasis Python. Sistem ini berfungsi sebagai penghubung

antara pengguna dan Model klasifikasi, mulai dari proses unggah citra hingga penampilan hasil prediksi motif batik.

Cara kerja sistem secara umum digambarkan pada Gambar 3.5, yang menunjukkan alur proses klasifikasi citra batik berbasis web. Alur tersebut menggambarkan tahapan utama yang dilalui sistem dalam memproses citra batik yang diberikan oleh pengguna hingga menghasilkan keluaran berupa hasil klasifikasi.



Gambar 3. 7 Alur Sistem Klasifikasi Motif Batik Berbasis Web

Berdasarkan alur pada Gambar 3.7, tahapan proses sistem dapat dijelaskan sebagai berikut:

1. *Input* Gambar Citra Batik

   Pada tahap ini, pengguna melakukan unggah citra motif batik melalui antarmuka web yang disediakan oleh sistem. Citra yang diunggah merupakan citra batik tulis tradisional yang akan diklasifikasikan. Sistem memastikan bahwa file gambar yang dimasukkan sesuai dengan format yang ditentukan sebelum diproses lebih lanjut.

2. Klasifikasi Citra Batik

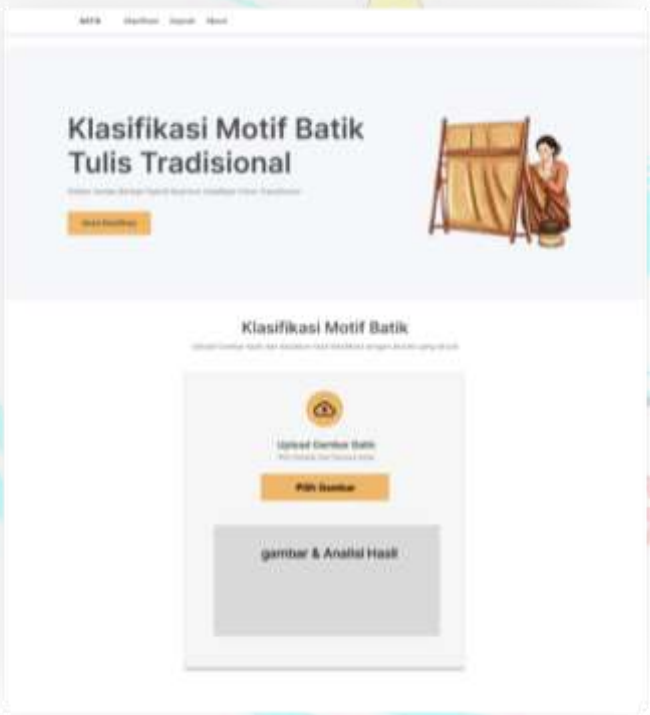
   Setelah citra berhasil diunggah, sistem mengirimkan data gambar ke *Backend* Flask untuk dilakukan pra-pemrosesan, seperti penyesuaian ukuran citra dan normalisasi. Citra yang telah dipra-proses kemudian diteruskan ke Model *Hybrid Quantum–Classical Vision Transformer* untuk dilakukan proses inferensi. Pada tahap ini, Model mengekstraksi fitur visual citra batik dan melakukan proses klasifikasi untuk menentukan kelas motif batik yang paling sesuai.

3. Hasil Klasifikasi Citra Batik

   Tahap akhir dari alur sistem adalah penyajian hasil klasifikasi. Sistem menerima keluaran dari Model berupa prediksi kelas motif batik beserta nilai tingkat kepercayaan (*confidence score*). Hasil tersebut kemudian ditampilkan pada halaman web agar dapat dilihat oleh pengguna. Informasi ini memberikan

gambaran mengenai jenis motif batik yang dikenali oleh sistem berdasarkan citra yang diunggah.

Sebagai bagian dari implementasi sistem berbasis web, tampilan antarmuka pengguna (*User Interface/UI*) dirancang untuk mendukung alur klasifikasi citra batik yang telah dijelaskan sebelumnya. Antarmuka sistem dibuat sederhana dan intuitif agar memudahkan pengguna dalam melakukan proses unggah citra serta memahami hasil klasifikasi yang ditampilkan oleh sistem. Gambar 3.7 menunjukkan rencana tampilan sistem klasifikasi motif batik berbasis web yang dikembangkan pada penelitian ini.



Gambar 3. 8 Rencana Antarmuka Pengguna

Pada halaman utama sistem disediakan area unggah gambar yang berfungsi sebagai *Input* citra batik. Pengguna dapat memilih dan mengunggah citra motif batik melalui komponen *Input* yang tersedia, kemudian sistem akan menampilkan pratinjau gambar sebelum proses klasifikasi dijalankan. Setelah proses klasifikasi selesai, sistem menampilkan hasil prediksi berupa nama motif batik beserta nilai tingkat kepercayaan (*confidence score*) pada halaman yang sama, sehingga pengguna dapat langsung melihat hasil klasifikasi dari citra yang diunggah.