

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Batik telah lama dikenal sebagai salah satu warisan budaya nusantara yang memiliki nilai seni tinggi. Selama berabad-abad, batik dikenal dunia sebagai budaya asli Indonesia. Kabupaten Rokan Hulu yang beribu kota di Pasir Pengaraian juga memiliki kekayaan budaya berupa batik daerah yang terus berkembang. Batik Rokan Hulu merupakan salah satu warisan budaya Indonesia yang memiliki nilai historis, filosofis, dan estetika yang tinggi. Batik merupakan seni tradisional kain dengan teknik pewarnaan menggunakan lilin untuk membentuk pola atau motif tertentu. Motif batik tidak hanya memiliki nilai keindahan, tetapi juga mengandung makna budaya yang mencerminkan identitas suatu daerah [1].

Salah satu pengrajin batik yang berada di Kecamatan Ujung Batu adalah Alifa Batik Rokan yang dimiliki oleh Bapak Agus Triyono, S.Sos., M.Pd. Usaha batik ini berlokasi di Jalan Batik RT.11 RW.01 Dusun Sei Sipoteh, Desa Ujung Batu Timur, Kecamatan Ujung Batu, Kabupaten Rokan Hulu, Provinsi Riau. Bergerak dalam produksi serta pengembangan motif batik khas daerah sejak tahun 2019 - sekarang. Berdasarkan hasil wawancara, permasalahan utama dalam pengelompokan motif Batik Rokan Hulu terletak pada variasi tekstur yang kompleks, perbedaan tingkat kerapatan pola, serta adanya citra dengan kualitas yang beragam. Kondisi ini menyulitkan proses identifikasi kesamaan motif apabila hanya mengandalkan pengamatan visual. Oleh karena itu, diperlukan

pendekatan berbasis ekstraksi fitur tekstur untuk merepresentasikan karakteristik citra secara kuantitatif, sehingga setiap motif memiliki nilai fitur yang dapat diolah secara komputasional.

Kondisi tersebut berpotensi menurunkan efektivitas proses identifikasi dan dokumentasi motif batik Rokan Hulu, serta menghambat upaya pelestarian dan pengembangan desain secara sistematis. Seiring dengan meningkatnya kebutuhan akan digitalisasi dan pengelolaan data berbasis teknologi dalam industri kreatif, diperlukan inovasi dalam metode analisis dan pengelompokan motif. Penerapan teknik komputasi seperti *clustering* berbasis ekstraksi fitur tekstur menjadi solusi yang strategis untuk menghasilkan pengelompokan motif yang lebih objektif, akurat, dan adaptif terhadap kompleksitas pola batik Rokan Hulu.

Untuk mengatasi permasalahan dalam pengelompokan motif batik Rokan Hulu, diperlukan pengembangan aplikasi berbasis analisis citra digital yang mampu mengelola data secara sistematis melalui tahapan ekstraksi fitur tekstur dan proses *clustering* otomatis. Pendekatan ini memungkinkan setiap citra motif direpresentasikan dalam bentuk vektor fitur numerik sebelum dilakukan pengelompokan berdasarkan karakteristik kepadatan data [2]. Salah satu *algoritma* yang relevan adalah *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN).

HDBSCAN merupakan metode analisis gerombol yang diperkenalkan sebagai alternatif dalam menangani data pencilan. Sebagai modifikasi dari *algoritma* OPTICS, HDBSCAN menggunakan parameter *input* tunggal, yaitu *MinPts* (*Minimum Points*). *Algoritma* ini tidak hanya mengidentifikasi gerombol

dengan kepadatan tinggi, tetapi juga secara otomatis mendeteksi pencilan sebagai data yang tidak termasuk dalam gerombol. HDBSCAN menggunakan *metrik mutual reachability distance* untuk mengidentifikasi titik data berdasarkan kepadatan dan membangun struktur gerombol *hirarki*. Titik yang tidak memenuhi ambang batas kepadatan akan ditandai sebagai pencilan, dan jika gerombol terputus dengan jumlah titik di bawah batas minimum, titik tersebut dianggap *noise*. [3]. Metode HDBSCAN di pilih karena memiliki keunggulan dalam mengidentifikasi struktur klaster yang lebih jelas dan memisahkan data secara lebih akurat.

Kajian hasil penelitian mengenai pengelompokan citra (*image clustering*) hingga saat ini telah banyak dilakukan oleh berbagai peneliti dengan memanfaatkan teknik ekstraksi fitur dan *algoritma clustering*. Penelitian terdahulu yang berkaitan dengan *clustering* citra dan ekstraksi fitur tekstur. salah satu penelitian dilakukan oleh Anugrah dkk, dengan judul “*Unsupervised Clustering of Handwritten Essay Answer Images Using Vision Transformer*”. Penelitian tersebut menerapkan *algoritma clustering*, termasuk HDBSCAN, untuk mengelompokkan citra tulisan tangan berdasarkan pola visual yang diperoleh dari proses ekstraksi fitur. Hasil penelitian menunjukkan bahwa penggunaan *algoritma* HDBSCAN mampu menghasilkan kualitas *cluster* yang baik dengan nilai *Silhouette Score* mencapai 0,772, sehingga metode ini dinilai efektif dalam mengelompokkan data citra secara otomatis. Penelitian ini menunjukkan bahwa *algoritma* HDBSCAN memiliki kemampuan yang baik dalam mengidentifikasi pola dan struktur data citra yang kompleks [4].

Penelitian lainnya dilakukan oleh Graham Stewart dan Mohammad Al-Khassaweneh, dengan judul “*An Implementation of the HDBSCAN Clustering Algorithm*”. Penelitian ini membahas penerapan *algoritma HDBSCAN* untuk mengidentifikasi *cluster* dan mendeteksi data penciran dalam suatu *dataset*. HDBSCAN merupakan pengembangan dari DBSCAN yang mampu membentuk *cluster* secara *hierarkis* berdasarkan kepadatan data. Hasil penelitian menunjukkan bahwa *algoritma HDBSCAN* efektif dalam menemukan struktur *cluster* pada data yang kompleks serta mampu memisahkan data yang dianggap sebagai *noise* sehingga menghasilkan pengelompokan data yang lebih optimal [5].

Selain itu, Penelitian lainnya yang telah diteliti oleh Zhiqiang Liang dkk. dengan judul “*An Effective Clustering Algorithm for the Low-Quality Image of Integrated Circuits via High-Frequency Texture Components Extraction*”. Dari hasil penelitian dapat disimpulkan bahwa proses pengelompokan citra dilakukan dengan memanfaatkan ekstraksi komponen tekstur frekuensi tinggi untuk merepresentasikan karakteristik citra sebelum dilakukan proses *clustering*. Metode tersebut diterapkan pada citra dengan kualitas rendah sehingga pola tekstur dapat dikenali dengan lebih jelas. Hasil penelitian menunjukkan bahwa penggunaan ekstraksi fitur tekstur yang dikombinasikan dengan *algoritma HDBSCAN* mampu menghasilkan pengelompokan citra yang lebih akurat serta dapat mengidentifikasi *cluster* data dan memisahkan data yang dianggap sebagai *noise* secara efektif [6].

Dari paparan di atas, berdasarkan beberapa penelitian terdahulu tersebut dapat disimpulkan bahwa kombinasi antara ekstraksi fitur tekstur dan *algoritma*

clustering berbasis kepadatan seperti HDBSCAN mampu memberikan hasil pengelompokan yang lebih efektif pada data citra. Oleh karena itu, pendekatan ini dinilai relevan untuk diterapkan dalam penelitian mengenai pengelompokan motif batik, Hal inilah yang melatar belakangi penulis dalam membuat laporan skripsi yang berjudul “*Clustering Motif Batik Menggunakan Algoritma Hierarchical Density-Based Spatial Clustering Of Applications With Noise (Hdbscan)* Berbasis Ekstraksi Fitur Tekstur”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka rumusan masalah dalam penelitian ini adalah bagaimana membangun dan menerapkan metode *clustering* menggunakan *algoritma* HDBSCAN untuk melakukan pengelompokan motif Batik Rokan Hulu?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun dan menerapkan metode *clustering* menggunakan *algoritma* HDBSCAN untuk melakukan pengelompokan motif Batik Rokan Hulu.

1.4 Batasan Masalah

Penelitian ini dibatasi pada pengelompokan motif batik sebagai data masukan yang terdiri dari 4.000 dataset motif Batik Rokan Hulu yang terbagi ke dalam 4 jenis kelas motif yaitu (Sicuriang, Tugu Ratik Togak, Sesuli Banja, dan Suluo Kekaik). Setiap jenisnya terdiri dari 1.000 dataset, adapun metode yang digunakan adalah *algoritma* HDBSCAN. Penelitian ini hanya berfokus pada tugas

clustering motif Batik Rokan Hulu tanpa mencakup identifikasi seniman batik serta analisis aspek historis batik.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat akademis berupa kontribusi terhadap pengembangan ilmu komputer, khususnya pada bidang *data mining* dan pengolahan citra digital dalam penerapan metode *clustering* untuk pengelompokan motif batik Rokan Hulu. Serta memberikan manfaat praktis berupa penyediaan solusi otomatis berbasis sistem yang dapat membantu pengerajin, pengelola galeri digital, maupun masyarakat umum dalam mengelompokkan dan mengenali jenis motif batik rokan hulu secara lebih cepat, objektif, dan konsisten.

1.6 Sistematika Penulisan

Sistematika penulisan ini terdiri dari lima bagian utama sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan dalam penelitian, seperti *data mining*, *clustering*, *algoritma* HDBSCAN, pengolahan citra digital, serta metode ekstraksi fitur tekstur.

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan penelitian yang meliputi pengumpulan data, proses ekstraksi fitur tekstur, penerapan *algoritma* HDBSCAN, serta metode analisis yang digunakan dalam penelitian.

BAB 4 ANALISA DAN PERANCANGAN

Bab ini berisi analisis kebutuhan sistem serta membangun aplikasi *clustering* motif batik menggunakan *algoritma* HDBSCAN.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi sistem yang telah dibangun serta pengujian terhadap aplikasi untuk mengetahui kinerja *algoritma* HDBSCAN dalam mengelompokkan motif batik.

BAB 6 PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan serta saran untuk pengembangan penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Data Mining

Data *mining* merupakan proses penggalian informasi atau pengetahuan baru dari sekumpulan data dalam jumlah besar dengan memanfaatkan teknik *statistik*, *machine learning*, dan basis data. Proses ini bertujuan untuk menemukan pola, hubungan, atau informasi tersembunyi yang sebelumnya tidak diketahui sehingga dapat digunakan sebagai dasar dalam pengambilan keputusan [7].

Menurut beberapa ahli, data *mining* didefinisikan sebagai gabungan dari berbagai disiplin ilmu seperti *artificial intelligence*, *machine learning*, *statistik*, dan sistem basis data yang digunakan untuk mengekstraksi pola dari data dalam skala besar [8]. Dengan demikian, *data mining* menjadi salah satu teknologi penting dalam pengolahan data modern karena mampu mengubah data mentah menjadi informasi yang bernilai.

Data *mining* merupakan suatu proses pencarian teknik analisa data yang besar untuk proses yang diperoleh dari berbagai macam basis data seperti data relasional data berorientasi objek dan data transaksi untuk mengetahui informasi yang baru didalam *database*. Data *mining* merupakan proses yang bersifat interaktif dan iteratif dalam menemukan pola atau model yang *valid*, baru, bermanfaat, dan dapat dipahami dari suatu basis data yang besar. Pola-pola yang ditemukan tersebut kemudian dapat digunakan untuk mendukung proses analisis dan pengambilan keputusan dimasa mendatang [9].

Menurut Gustiana (2024), *Data mining* memiliki beberapa metode yang umum digunakan meliputi [10]:

a. *Clustering*

Clustering merupakan metode yang digunakan untuk mengelompokkan data ke dalam beberapa kelompok berdasarkan tingkat kemiripan karakteristik data.

b. Klasifikasi

Klasifikasi adalah metode yang digunakan untuk mengelompokkan data ke dalam kategori atau kelas yang telah ditentukan sebelumnya.

c. *Regresi*

Regresi merupakan metode yang digunakan untuk memprediksi nilai *numerik* atau nilai *kontinu* berdasarkan hubungan antara beberapa *variabel* dalam data.

d. *Asosiasi*

Asosiasi adalah metode yang digunakan untuk menemukan hubungan atau pola keterkaitan antar item dalam suatu dataset.

Machine learning adalah istilah yang digunakan untuk merujuk pada cabang ilmu komputer yang mempelajari metode desain *algoritma* yang mampu mempelajari atau beradaptasi dengan pola data tanpa diprogram secara *eksplisit*. *Machine Learning* adalah sebuah mesin yang dibuat untuk dapat belajar dan melakukan pekerjaan tanpa arahan dari penggunanya [11].

2.1.1 Knowledge Discovery in Database (KDD)

Definisi *Knowledge Discovery in Database* (KDD) merupakan proses untuk menemukan pengetahuan yang berguna, pola, serta informasi penting dari kumpulan data dalam jumlah besar dengan memanfaatkan teknik analisis data [12]. Proses ini dilakukan secara sistematis melalui beberapa tahapan mulai dari pemilihan data hingga penyajian hasil.

KDD digunakan dalam penelitian ini sebagai kerangka kerja untuk mengolah data citra motif batik sehingga dapat menghasilkan informasi yang bermakna. Melalui proses KDD, data yang berasal dari basis data diolah dan dianalisis untuk mendukung pengambilan keputusan. KDD sering kali disamakan dengan *data mining*, namun keduanya memiliki perbedaan. *Data mining* merupakan salah satu tahapan dalam KDD yang berfokus pada proses ekstraksi pola atau informasi tersembunyi dari data, sedangkan KDD mencakup keseluruhan proses pengolahan data dari awal hingga menghasilkan pengetahuan yang dapat dipahami [13].



Gambar 2.1 Tahapan *Knowledge Discovery in Database* (KDD) [13].

Knowledge Discovery in Database (KDD) adalah keseluruhan proses untuk mencari dan mengidentifikasi pola dalam data, dimana pola yang ditemukan bersifat sah, baru, dapat bermanfaat dan dapat dimengerti. KDD berhubungan dengan teknik integrasi dan penemuan ilmiah, interpretasi dan visualisasi dari pola-pola sejumlah kumpulan data. KDD dan *data mining* sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi [13].

Beberapa tahapan dalam proses KDD adalah sebagai berikut [13] :

1. Pemilihan Data (*Data Selection*)

Tidak seluruh data dalam basis data digunakan dalam proses analisis. Hanya data yang relevan yang dipilih, seperti citra motif batik yang memiliki karakteristik tekstur untuk dianalisis. Informasi yang tidak berkaitan dengan proses pengelompokan motif batik tidak digunakan.

2. Pembersih Data (*Data Cleaning*)

Tahap ini dilakukan untuk meningkatkan kualitas data dengan menghilangkan *noise*, memperbaiki citra yang tidak jelas, serta menyaring data yang tidak relevan.

3. Integrasi Data (*Data Integration*)

Data yang berasal dari berbagai sumber digabungkan menjadi satu dataset yang terstruktur untuk mempermudah proses analisis. Integrasi data yang dilakukan secara tepat dapat mengurangi redundansi serta meningkatkan akurasi dan efisiensi analisis.

4. Transformasi Data (*Data Transformation*)

Pada tahapan ini, data citra motif batik ditransformasikan melalui proses ekstraksi fitur tekstur sehingga dapat direpresentasikan dalam bentuk nilai numerik. Hasil transformasi digunakan sebagai *input* dalam proses *clustering* menggunakan *algoritma* HDBSCAN.

5. Data Mining

Pada tahap ini diterapkan *algoritma* HDBSCAN untuk menemukan pola pengelompokan motif batik berdasarkan kemiripan fitur tekstur.

6. Evaluasi Pola (*Pattern Evaluation*)

Hasil *clustering* dievaluasi untuk memastikan kualitas dan kesesuaian pola yang dihasilkan.

7. Presentasi Pengetahuan (*Knowledge Presentation*)

Knowledge presentation merupakan tahapan akhir dalam proses data mining, hasil analisis disajikan dalam bentuk tabel, grafik, atau visualisasi *cluster* agar mudah dipahami.

2.2 Clustering

Clustering adalah proses pengelompokan objek data ke dalam kelompok atau *Cluster*, di mana objek dalam satu *Cluster* memiliki kesamaan tertentu, sedangkan objek di *Cluster* yang berbeda memiliki perbedaan yang signifikan [14]. *Clustering* merupakan proses pengelompokan sekumpulan data ke dalam beberapa *cluster* berdasarkan tingkat kemiripan tertentu. Objek yang memiliki kemiripan tinggi akan berada dalam satu *cluster* yang sama, sedangkan objek yang memiliki perbedaan signifikan akan berada pada *cluster* yang berbeda [15]. *Clustering* dapat diklasifikasikan berdasarkan model *cluster* yang digunakan,

seperti *clustering* berbasis *centroid*, *clustering* berbasis kepadatan, *clustering* berbasis distribusi, *clustering* berbasis hirarki, atau *clustering* berbasis konektivitas [16]. *Cluster* sendiri didefinisikan sebagai sekumpulan objek data yang memiliki tingkat kemiripan yang tinggi dalam satu kelompok dibandingkan dengan objek pada kelompok lainnya. Dalam proses *clustering*, setiap objek dikelompokkan sedemikian rupa sehingga tingkat kesamaan dalam satu *cluster* (*intra-cluster similarity*) tinggi dan tingkat kesamaan antar *cluster* (*inter-cluster similarity*) rendah [17].

Clustering termasuk dalam metode analisis data yang merupakan bagian dari *data mining* dan bersifat *unsupervised learning*, dimana proses pengelompokan dilakukan tanpa menggunakan label data. Tujuan utama dari metode ini adalah mengelompokkan data dengan karakteristik serupa ke dalam satu kelompok yang sama sehingga memudahkan dalam analisis pola dan pengambilan keputusan [18].

2.3 Pengolahan Citra Digital

Pengolahan citra digital merupakan teknik yang digunakan untuk meningkatkan kualitas citra, sehingga lebih mudah dipahami dan diinterpretasikan, baik oleh manusia maupun mesin [19]. Citra digital sendiri merupakan representasi dari suatu objek dalam bentuk dua dimensi yang tersusun atas elemen-elemen kecil yang disebut piksel (*picture element*). Setiap piksel memiliki nilai intensitas tertentu yang merepresentasikan tingkat kecerahan atau warna pada titik tersebut.

Dalam pengolahan citra digital, citra direpresentasikan dalam bentuk *matriks* dua dimensi, di mana setiap elemen *matriks* menunjukkan nilai intensitas *piksel*. Proses ini bertujuan untuk meningkatkan kualitas citra, mempermudah analisis, serta mengekstraksi informasi penting yang dapat digunakan dalam berbagai aplikasi seperti pengenalan pola, klasifikasi, dan *clustering* [20].

Tahapan dalam pengolahan citra digital terdiri dari beberapa proses utama yang dilakukan secara berurutan untuk menghasilkan data yang siap dianalisis. Tahapan tersebut meliputi [20]:

1. Akuisisi Citra (*Image Acquisition*)

Akuisisi citra merupakan tahap awal untuk memperoleh citra digital dari objek menggunakan perangkat seperti kamera atau sensor. Hasil yang diperoleh masih berupa citra mentah dan kualitasnya sangat mempengaruhi proses selanjutnya.

2. *Preprocessing* (Pra-pemrosesan)

Preprocessing bertujuan meningkatkan kualitas citra agar lebih mudah dianalisis, seperti mengurangi *noise* dan menyederhanakan data. Teknik yang digunakan meliputi:

- a. *Grayscale*: mengubah citra berwarna menjadi keabuan untuk mengurangi kompleksitas.
- b. *Resizing*: menyesuaikan ukuran citra agar seragam dan efisien.
- c. *Filtering*: mengurangi *noise* untuk meningkatkan kualitas citra.

3. Ekstraksi Fitur (*Feature Extraction*)

Ekstraksi fitur adalah proses mengambil informasi penting dari citra, seperti tekstur, warna, atau bentuk, yang digunakan sebagai representasi data dalam analisis lanjutan seperti klasifikasi atau *clustering*.

2.4 Batik

Batik merupakan warisan budaya indonesia yang memiliki nilai seni tinggi dan telah diakui oleh dunia internasional. Salah satu bentuk perkembangan batik di Indonesia adalah batik Rokan Hulu yang memiliki ciri khas tersendiri yang dipengaruhi oleh budaya melayu. Batik dihasilkan melalui proses pewarnaan kain dengan teknik tertentu sehingga membentuk pola atau motif yang khas. Batik juga menjadi salah satu identitas bangsa indonesia, di mana penggunaannya mencerminkan kekayaan budaya lokal dari berbagai daerah, termasuk Kabupaten Rokan Hulu [21].

Batik adalah sejenis kain dengan motif-motif yang khas yang dibuat dengan teknik khusus [22]. Motif batik dipengaruhi oleh berbagai faktor seperti letak geografis, kondisi alam, budaya, kepercayaan, serta adat istiadat suatu daerah. Pada Batik Rokan Hulu, motif yang digunakan umumnya terinspirasi dari unsur budaya melayu dan lingkungan sekitar, seperti motif sicuriang, suluo kekaik, sesuli banja, sicuriang ludai, zaleha dan tugu ratik tagak. Daerah Rokan Hulu menghasilkan motif yang berkaitan dengan kebudayaan melayu sungai Rokan secara umum hulu dan hilir. Tradisi lima luhak yaitu Tambusai, Rambah, Kepenuhan, Kunto Darusalam dan Rokan Iv Koto. Keunikan desain batik diangkat dari khazanah tradisi, sejarah, kekayaan alam di daerah. Hal ini

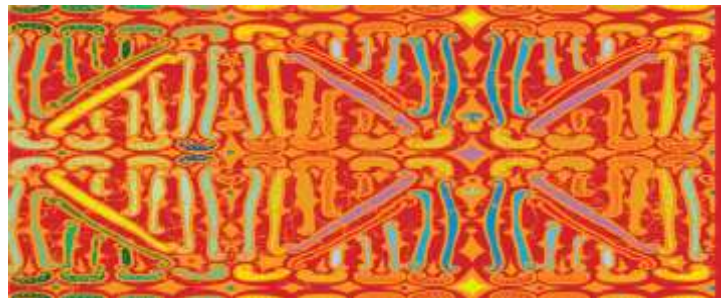
menyebabkan Batik Rokan Hulu memiliki keunikan tersendiri dibandingkan dengan batik dari daerah lain [23].

Pengertian motif dan makna filosofis Batik Rokan Hulu:

Sumber: Dokumen pemanfaatan batik Rokan Hulu tahun 2022 [24].

1. Sicuriang

Sicuriang dapat diartikan sebagai coretan garis-garis yang dipakai untuk menghitung jumlah, menggantikan simbol angka bilangan atau dapat disebut sebagai metode berhitung melayu.



Gambar 2.2 Sicuriang

Sicuriang dapat dimaknai sebagai tradisi orang melayu sungai rokan yang senantiasa mempergunakan perhitungan dalam segala hal. Sesuai dengan bunyi pepatah adat melayu ‘salah taksir hilang pelabaan’ artinya salah perhitungan akan beroleh kerugian.

2. Suluo Kekaik

Suluo kekaik dapat diartikan sebagai jenis akar-akaran yang tumbuh di tepi tepi rimba, tepi sungai, dan sosok ladang, tumbuh membentuk kait kemait yang unik, mirip pola dasar dalam ragam hias melayu. Dipilihlah akar kekaik sebagai basis ragam hias produk disain batik Rokan Hulu dengan nama suluo kekaik.



Gambar 2.3 Suluo Kekaik

Suluo kekaik dapat dimaknai sebagai hubungan kekerabatan antar orang dalam wilayah kebudayaan sungai rokan tanpa memandang suku bangsa dan asal usul sesuai dengan ungkapan adat ‘kaik kemaik bak kayu akaran’ artinya semakin banyak aspek terkaitan maka semakin erat kerekatan antar masyarakat Rokan Hulu.

3. Sesuli Banja

Sesuli banja dapat diartikan sebagai ragam jenis rumput atau semak liar di bekas banjar ladang dan salah satunya adalah rumput sulibanja; sejenis rumput yang mudah dijumpai tumbuh dan berkembang di banjar ladang, meski pun bunga semak liar kecil dan halus-halus namun membuat suasana banjar dan perkempungan melayu indah mempersona.



Gambar 2.4 Sesuli Banja

Sesuli banja dapat dimaknai sebagai simbolisasi pernak pernik keragaman tradisi dan budaya masyarakat di kabupaten rokan hulu tanpa memandang suku bangsa dan asal usul, sesuai dengan bunyi pepatah adat melayu ‘berubong dek nak panjang, bogoba dek nak leba’ (dihubungkan karena ingin panjang, dihamparkan karena ingin luas) artinya semakin beragam dan bervariasi kebudayaan masyarakat di kabupaten rokan hulu yang berasal dari berbagai suku bangsa tidaklah menjadi hambatan untuk persatuan dan kesatuan.

4. Sicuriang Ludai

Sicuriang ludai dapat diartikan sebagai pengembangan motif sicuriang yang digabung dengan konsep motif daun ludai, yakni makanan mamalia seperti kancil, kijang dan rusa, lazimnya dijadikan umpan perangkap atau tinjak.



Gambar 2. 5 Sicuriang Ludai

Sicuriang ludai dapat dimaknai sebagai mana dimaksud dalam pasal 4 huruf a, yaitu Sicuriang dapat dimaknai sebagai tradisi orang melayu sungai rokan yang senantiasa mempergunakan perhitungan dalam segala

hal. Sesuai dengan bunyi pepatah adat Melayu ‘salah taksir hilang pelabaan’ artinya salah perhitungan akan beroleh kerugian.

5. Zaleha

Zaleha dapat diartikan sebagai ragam hias yang diangkat dari hasil Sulaman istri yang dipertuan sakti Rokan Iv Koto, bernama zaleha. Desain ragam hias ini baru dan unik serta diberi nama motif zaleha.



Gambar 2. 6 Zaleha

Zaleha dapat dimaknai sebagai apresiasi terhadap benda cagar budaya yang bernilai seni unik, terkait sejarah kerajaan Rokan IV Koto.

6. Tugu Ratik Togak

Tugu ratik toga k dapat diartikan sebagai ragam hias merujuk kepada tradisi ratik togak (ratib berdiri), ratik bosa (ratib besar), ratik merontak (ratib merentak), ratik tulak bala (ratib tolak bala), merupakan tradisi yang dikekalkan oleh penganut thariqat naqsabandiyah. Tugu ratik togak pun telah menjadi ikon landmark Kabupaten Rokan Hulu sebagai negeri seribu suluk.



Gambar 2. 7 Tugu Ratik Togak

Tugu ratik togak dapat dimaknai sebagai representasi tradisi ratik togak (ratib berdiri), ratik bosa (ratib besar), ratik merontak (ratib merentak), ratik tulak bala (ratib tolak bala), sekaligus mewakili rokan hulu sebagai negeri seribu suluk.

2.5 Ekstraksi Fitur Tekstur

Ekstraksi fitur adalah salah satu proses pada pengolahan citra (*image processing*) bertujuan untuk mengambil informasi ciri-ciri tertentu dan karakter dalam suatu citra. Ekstraksi fitur penting dilakukan agar memperoleh informasi penting yang dapat memberikan deskripsi dan interpretasi mengenai sebuah objek [25].

Ekstraksi fitur tekstur terbagi menjadi dua yaitu GLCM dan LBP:

1. *Gray Level Co-occurrence Matrix (GLCM)*

GLCM merupakan metode *statistik* orde dua yang digunakan untuk menganalisis tekstur citra dengan memperhatikan hubungan spasial antar *piksel* berdasarkan jarak dan arah tertentu. Metode ini menghitung seberapa sering pasangan *piksel* dengan nilai intensitas tertentu muncul dalam suatu citra.

Dalam analisis ekstraksi fitur tekstur pada citra motif batik, fitur GLCM yang digunakan antara lain [26]:

1. Contrast

Contrast (C) merupakan ukuran yang digunakan untuk mengetahui tingkat perbedaan intensitas antara *piksel* dengan *piksel* tetangganya dalam suatu citra. Nilai *contrast* yang tinggi menunjukkan adanya perbedaan intensitas yang besar sehingga tekstur citra tampak lebih kasar. Perhitungan nilai *contrast* menggunakan Persamaan 1

$$Contrast = \sum_{i,j} (i - j)^2 P(i, j) \dots\dots\dots(1)$$

Keterangan Rumus:

Contrast: Nilai kontras tekstur citra.

i: Nilai intensitas piksel acuan / baris pada matriks GLCM.

j: Nilai intensitas piksel tetangga / kolom pada matriks GLCM.

P(i, j): Nilai probabilitas atau frekuensi relatif terjadinya pasangan intensitas *i* dan *j* pada GLCM.

(*i-j*)²: Selisih kuadrat antara dua nilai intensitas piksel yang bertetangga.

$\sum_{i,j}$: Penjumlahan untuk seluruh pasangan *i* dan *j* dalam matriks GLCM.

2. Correlation

Correlation (Cr) digunakan untuk mengukur tingkat hubungan atau ketergantungan *linear* antara *piksel* dalam suatu citra. Nilai *correlation* menunjukkan seberapa besar keterkaitan pola antar *piksel* dalam citra.

Perhitungan nilai *correlation* menggunakan Persamaan 2

$$Correlation = \sum_{i,j} \frac{(i-\mu)(j-\mu)P(i,j)}{\sigma_i \sigma_j} \dots\dots\dots(2)$$

Keterangan Rumus:

Correlation: Nilai korelasi tekstur antar piksel bertetangga.

i : Nilai intensitas piksel acuan / baris pada matriks GLCM.

j : Nilai intensitas piksel tetangga / kolom pada matriks GLCM.

$P(i, j)$: Nilai probabilitas atau frekuensi relatif terjadinya pasangan intensitas i dan j pada GLCM.

μ : Nilai rata-rata intensitas pada distribusi GLCM.

σ_i : Standar deviasi nilai intensitas pada arah baris / variabel i .

σ_j : Standar deviasi nilai intensitas pada arah kolom / variabel j .

$\sum_{i,j}$: Penjumlahan untuk seluruh pasangan i dan j dalam matriks GLCM.

3. *Energy*

Energy atau *Angular Second Moment (ASM)* merupakan ukuran yang menunjukkan tingkat keseragaman tekstur dalam citra. Nilai *energy* yang tinggi menunjukkan bahwa citra memiliki pola yang seragam atau *homogen*. Perhitungan nilai *energy* menggunakan Persamaan 3

$$ASM = \sum_{i,j} P(i, j)^2 \dots\dots\dots(3)$$

Keterangan Rumus:

ASM: Nilai Angular Second Moment atau *Energy*.

i : Nilai intensitas piksel acuan / baris pada matriks GLCM.

j : Nilai intensitas piksel tetangga / kolom pada matriks GLCM.

$P(i, j)$: Nilai probabilitas atau frekuensi relatif terjadinya pasangan intensitas i dan j pada GLCM.

$P(i, j)^2$: Kuadrat dari nilai probabilitas pasangan intensitas i dan j .

$\sum_{i,j}$: Penjumlahan untuk seluruh pasangan i dan j dalam matriks GLCM.

4. Homogeneity

Homogeneity (H) digunakan untuk mengukur tingkat kedekatan distribusi nilai *piksel* terhadap diagonal utama pada *matriks GLCM*. Nilai *homogeneity* yang tinggi menunjukkan bahwa citra memiliki tingkat keseragaman yang tinggi. *Homogeneity* memiliki hubungan berbanding terbalik dengan *contrast*, dimana semakin tinggi nilai *contrast* maka nilai *homogeneity* cenderung semakin rendah, dan sebaliknya. Perhitungan nilai *homogeneity* menggunakan Persamaan 4

$$Homogeneity = \sum_{i,j} \frac{P(i,j)}{1+|i-j|} \dots\dots\dots(4)$$

Keterangan Rumus:

Homogeneity : Nilai kehomogenan tekstur citra.

i: Nilai intensitas piksel acuan / baris pada matriks GLCM.

j: Nilai intensitas piksel tetangga / kolom pada matriks GLCM.

$P(i,j)$: Nilai probabilitas atau frekuensi relatif terjadinya pasangan intensitas *i* dan *j* pada GLCM.

$\sum_{i,j}$: Penjumlahan untuk seluruh pasangan *i* dan *j* dalam matriks GLCM.

2. Local Binary Pattern (LBP)

Local Binary Pattern (LBP) berfungsi sebagai descriptor tekstur yang mampu menangkap pola lokal pada citra. Nilai LBP untuk suatu *piksel* dihitung dengan membandingkan intensitas *piksel* pusat dengan *piksel* tetangganya, yang biasanya berjumlah delapan pada citra berukuran 3×3. Proses perbandingan dilakukan secara berurutan, baik searah jarum jam (*clockwise*) maupun berlawanan arah jarum jam (*counter-clockwise*) dengan aturan yang konsisten.

Secara *matematis*, nilai LBP dapat dinyatakan sebagai berikut [1]:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(I_p - I_c) 2^p \dots\dots\dots(5)$$

Keterangan Rumus:

$LBP_{P,R}$:Nilai kode LBP untuk suatu piksel pusat.

P :Jumlah piksel tetangga yang digunakan.

R :Jarak radius lingkaran dari piksel pusat ke piksel tetangga.

p :Indeks piksel tetangga, mulai dari $p = 0$ sampai $p = P - 1$.

I_c :Nilai intensitas piksel pusat atau center pixel.

I_p :Nilai intensitas piksel tetangga ke- p .

$I_p - I_c$:Selisih antara intensitas piksel tetangga dan piksel pusat.

$s(\cdot)$:Fungsi threshold atau fungsi biner.

2^p : Bobot biner berdasarkan posisi piksel tetangga ke- p .

$\sum_{p=0}^{P-1}$: Penjumlahan untuk seluruh piksel tetangga sebanyak P .

dengan fungsi:

$$s(x) = \begin{cases} 0, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases} \dots\dots\dots(6)$$

jika intensitas piksel tetangga lebih besar atau sama dengan intensitas piksel pusat, maka nilainya diubah menjadi 1. Jika lebih kecil, maka nilainya diubah menjadi 0.

2.6 **Algoritma HDBSCAN**

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) merupakan *algoritma clustering* berbasis kepadatan (*density-based clustering*) yang dikembangkan dari metode DBSCAN [14]. HDBSCAN

digunakan untuk mengatasi keterbatasan DBSCAN dalam menangani data dengan variasi kepadatan yang berbeda serta ketergantungan terhadap parameter *epsilon* (ϵ). *Algoritma* ini di rancang untuk mengidentifikasi klaster berdasarkan kepadatan data dengan parameter utama bernama minimum *points* (*MinPts*), yang menentukan sensitivitas *algoritma* terhadap pola data [27]. *Algoritma analisis* klaster ini banyak digunakan karena ketahanannya terhadap noise dalam kumpulan data [16]. *Algoritma* HDBSCAN untuk mengelompokkan pola akses dan mengidentifikasi aktivitas anomali. Proses analisis mencakup *preprocessing* *log*, *masking* dan normalisasi pola URL, chunking data, embedding berbasis transformer, serta *clustering* densitas [28].

Berikut adalah langkah-langkah penerapan HDBSCAN [29] :

1. Menentukan parameter:
 - a. *min_samples* (jumlah tetangga minimum)
 - b. *min_cluster_size* (ukuran cluster minimum)
2. Menghitung jarak antar seluruh titik data menggunakan jarak *Euclidean*.

$$d(p,q)=\sqrt{(p^1 - q^1)^2 + (p^2 - q^2)^2 + \cdots + (p_n - p_n)^2}.....(7)$$
3. Menghitung *core distance* untuk setiap titik berdasarkan nilai *min_samples*.
4. Menghitung *mutual reachability distance* antar titik menggunakan rumus:

$$d_{mreach}(a,b) = \max\{core_k(a), core_k(b), d(a,b)\}.....(8)$$
5. Membangun *Minimum Spanning Tree* (MST) berdasarkan *mutual reachability distance*.
6. Mengubah MST menjadi struktur *cluster hirarki* (*hierarchical clustering*).
7. Mengekstraksi *cluster* terbaik berdasarkan *stability* (kestabilan *cluster*).

8. Mengidentifikasi titik yang tidak termasuk *cluster* sebagai *noise*.

2.7 Evaluasi *Clustering* (*Silhouette Index*)

Evaluasi *clustering* merupakan proses untuk mengukur kualitas hasil pengelompokan data yang dihasilkan oleh suatu *algoritma clustering*. *Silhouette Index* adalah *metrik* evaluasi internal yang digunakan untuk mengevaluasi kualitas *Clustering*. *Silhouette Index* merupakan metode evaluasi *clustering* yang digunakan untuk mengukur kualitas hasil pengelompokan dengan membandingkan kedekatan suatu data terhadap *cluster*-nya sendiri (*kohesi*) dan *cluster* lainnya (*separasi*). Nilai *Silhouette Index* berada pada rentang -1 hingga 1, di mana nilai yang lebih tinggi menunjukkan bahwa data telah terkelompok dengan baik [14].

Hasil perhitungan menggunakan *Silhouette Index* dinyatakan dalam bentuk *Silhouette Score*. *Silhouette Score* merupakan nilai rata-rata dari seluruh nilai *silhouette* setiap data pada dataset yang digunakan. Nilai yang mendekati 1 menunjukkan bahwa objek berada pada *cluster* yang tepat dan memiliki pemisahan yang baik terhadap *cluster* lain. Nilai yang mendekati 0 menunjukkan bahwa objek berada di batas antara dua *cluster*, sedangkan nilai yang mendekati -1 menunjukkan bahwa objek kemungkinan lebih sesuai ditempatkan pada *cluster* lain [30].

Silhouette Index dihitung berdasarkan dua komponen utama, yaitu [30]:

1. Kohesi ($a(i)$)

Kohesi merupakan rata-rata jarak antara suatu data dengan seluruh data lain dalam *cluster* yang sama.

2. Separasi (b(i))

Separasi merupakan rata-rata jarak antara suatu data dengan *cluster* terdekat lainnya.

Rumus *Silhouette Index* adalah sebagai berikut:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \dots\dots\dots(9)$$

Keterangan:

$a(i)$ = rata-rata jarak data ke- i dalam *cluster* yang sama.

$b(i)$ = rata-rata jarak data ke- i ke *cluster* terdekat.

$S(i)$ = nilai silhouette untuk data ke- i , dengan rentang nilai -1 hingga 1 .

2.8 Bahasa pemrograman

2.8.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer, dikenal karena sintaksnya yang sederhana dan mudah dipahami [31]. Sehingga banyak digunakan dalam penelitian di bidang data *mining* dan *machine learning*. *Python* memiliki sifat fleksibel dan dapat digunakan untuk berbagai keperluan seperti pengembangan perangkat lunak, pengembangan *web*, serta analisis data. Selain itu, *Python* bersifat *open source*, sehingga dapat digunakan, dimodifikasi, dan didistribusikan secara bebas oleh penggunanya.

Python merupakan bahasa perograman tingkat tinggi yang mana dalam segi perancangan bahasa berfokus pada tingkat keterbacaan dari *programming code* sehingga mudah untuk dipelajari. *Python* bisa dijalankan melalui berbagai layanan *sistem operasi Windows (python)*, *Linux*, *Android (Google Play)*, *Mac OS* dan lainnya [32]. *Python* digunakan sebagai bahasa pemrograman utama untuk

mengimplementasikan proses pengolahan citra, ekstraksi fitur tekstur, serta proses *clustering* menggunakan *algoritma* HDBSCAN [33]. *Python* juga didukung oleh berbagai pustaka (*library*) yang mempermudah proses pengembangan sistem, seperti *OpenCV* untuk pengolahan citra, *Scikit-image* untuk ekstraksi fitur, serta HDBSCAN untuk proses *clustering*. Dengan dukungan *library* yang lengkap dan kemudahan dalam penggunaannya, *Python* menjadi salah satu pilihan utama dalam pengembangan sistem berbasis pengolahan citra dan *machine learning*.

2.8.2 HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) adalah bentuk bahasa kode yang biasa digunakan oleh programmer untuk membuat halaman *web* dan memungkinkan untuk mendesain *web* dengan menyajikan teks, gambar, video dan konten lainnya secara jelas dan ringkas. HTML merupakan bahasa standar yang mampu menampilkan halaman *web* dan menentukan *browser web* menampilkan konten sehingga HTML merupakan bagian penting untuk merancang suatu situs *web* [34].

HTML merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman *web*. HTML berfungsi untuk mempublikasi dokumen *online*. Statement dasar dari HTML disebut tags. Sebuah tag dinyatakan dalam sebuah kurung siku (<>). Tags yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen haruslah dibuat berupa pasangan. Terdiri dari tag pembuka dan tag penutup. Dimana tag penutup menggunakan tambahan tanda garis miring (/) di awal nama tag [35].

Dengan HTML kita dapat mendirikan sebuah *website* dengan coding tag –tag HTML. Atau dengan kata lain HTML merupakan tubuh dari sebuah *website* [36].

2.8.3 CSS (*Cascading Style Sheet*)

CSS adalah bahasa-bahasa yang merepresentasikan halaman *web*. Seperti warna, *layout*, dan *font*. Dengan menggunakan CSS, seorang web developer dapat membuat halaman *web* yang dapat beradaptasi dengan berbagai macam ukuran layar. Pembuatan CSS biasanya terpisah dengan halaman HTML. Meskipun CSS dapat disisipkan di dalam halaman HTML. Hal ini ditujukan untuk memudahkan pengaturan halaman HTML yang memiliki rancangan yang sama [35]. CSS adalah dokumen *web* yang cocok untuk organisasi elemen HTML dengan properti berbeda tersedia untuk ditampilkan dalam gaya apapun yang anda inginkan [37]. CSS digunakan untuk memperindah atau mempercantik sebuah tampilan *website*. Biasanya CSS membantu HTML dalam proses memperindah sebuah *website* [36].

2.8.4 JavaScript

Java Script adalah bahasa pemrograman *web* yang bersifat *Client Side Programming Language*. *Client Side Programming Language Java Script* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini* dan sebagainya. *Java Script* pertama kali dikembangkan pada pertengahan dekade 90'an. Meskipun memiliki nama yang hampir serupa, *Java Script* berbeda dengan bahasa *pemrograman Java*. Untuk

penulisannya, *Java Script* dapat disisipkan di dalam dokumen HTML ataupun dijadikan dokumen tersendiri yang kemudian diasosiasikan dengan dokumen lain yang dituju. *Java Script* mengimplementasikan fitur yang dirancang untuk mengendalikan bagaimana sebuah halaman *web* berinteraksi dengan penggunaannya [35].

JavaScript merupakan bahasa pemrograman yang dirancang khusus untuk pengembangan *web*. Bahasa ini berjalan di sisi klien (*client-side*), dengan kemampuan interaktif yang unggul, *JavaScript* memungkinkan pembuatan aplikasi *web* yang responsif. Selain itu, *JavaScript* sering dimanfaatkan dalam pengembangan *web*, baik pada bagian *front-end* maupun *back-end* [37]. *Javascript* digunakan sehingga *website* menjadi lebih dinamis dan interaktif seperti membuat suatu fungsi yang dijalankan dan lain-lain [38].

2.9 Alat Bantu Pemrograman

2.9.1 OpenCV (*Open Source Computer Vision Library*)

Open Source Computer Vision Library (OpenCV) adalah sebuah *software* yang bersifat *open-source* yang berisikan *library* yang mendukung pendeteksian objek yang mudah diimpor kedalam bahasa pemrograman *Java* [39]. Menurut Randy Moh Yusup dkk, OpenCV adalah perpustakaan *computer vision* sumber terbuka secara luas yang dikenal dengan koleksi fungsi dan algoritma yang luas [40].

Menurut D. Arniady, OpenCV adalah sebuah *library* (perpustakaan) yang digunakan untuk mengolah gambar dan video hingga kita mampu mengekstrak

informasi di dalamnya. OpenCV dapat berjalan di berbagai bahasa pemrograman, seperti *C*, *C++*, *Java*, *Python*, dan juga didukung di berbagai platform seperti *Windows*, *Linux*, *Mac OS*, *iOS* dan *Android* [41].

2.9.2 **Visual Studio Code**

Visual Studio Code merupakan *Integrated Development Environment* (IDE) yang dikembangkan oleh *Microsoft* dan digunakan secara luas dalam pengembangan aplikasi berbasis *web*. IDE ini menyediakan fasilitas editor kode sumber yang mendukung berbagai bahasa pemrograman seperti *HTML*, *CSS*, *JavaScript*, dan *PHP*, serta dilengkapi dengan fitur *syntax highlighting*, *code completion*, *debugging*, dan dukungan ekstensi yang dapat meningkatkan produktivitas pengembang sistem [42]. Dibangun dengan *Electron*, *VS Code* dilengkapi fitur seperti *Command Palette*, *Terminal*, dan *Extensions*, serta terintegrasi dengan berbagai tools pengembangan. Keunggulannya meliputi gratis, fleksibel, kaya ekstensi, dan mendukung berbagai bahasa pemrograman [43].

Visual Studio Code adalah perangkat lunak editor kode sumber yang dikembangkan oleh *Microsoft* dan dapat digunakan pada berbagai sistem operasi seperti *Linux*, *macOS*, dan *Windows*. Editor ini mendukung pengeditan berbagai bahasa pemrograman, termasuk *JavaScript*, *HTML*, *CSS*, dan lainnya [44].

2.9.3 **Web Browser**

web browser adalah perangkat utama yang akan kita gunakan untuk menampilkan halaman *web* yang pada dasarnya terbuat dari *HTML* dan *CSS*. Saya yakin di setiap komputer telah terinstall *Web Browser* bawaan seperti *Internet Explorer* (*Windows*), *Safari* (*Mac*) dan *Firefox* (*Linux Ubuntu*) [45]. *Web*

browser adalah aplikasi perangkat lunak yang digunakan untuk mengakses dan menampilkan sumber informasi. *Web browser* merupakan perangkat lunak yang dapat memproses paket HTTP dan menampilkannya kembali kepada *user* dengan format HTML [46]. *Web browser* merupakan perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi di *internet* yang dengan mudahnya, *browser* digunakan untuk menampilkan halaman-halaman *web*. Seluruh komponen *web* termasuk teks, gambar, dan komponen lain yang dibangun dengan teknologi *client-side scripting* dapat ditampilkan di *web browser* [47].

2.9.4 **Google Colab**

Google Colab adalah sebuah IDE untuk *pemrograman Python* dimana pemrosesan akan dilakukan oleh *server Google* yang memiliki perangkat keras dengan performa yang tinggi [48]. Alat ini bertujuan untuk membantu peneliti mengolah data demi keperluan belajar dan melakukan eksperimen dengan pengolahan data, terutama dalam bidang pembelajaran mesin. Penggunaan alat ini mirip dengan *Jupyter Notebook*. *Google Colab* dibangun di atas lingkungan *Jupyter*, sehingga tidak perlu mengatur apa pun sebelum digunakan, dan berjalan sepenuhnya di *Cloud* dengan menggunakan media penyimpanan *Google Drive* [49].

Keunggulan *Google Colab* meliputi kemudahan penggunaan, dukungan *library data science*, kolaborasi, dan reproduktibilitas penelitian. Namun, keterbatasan sumber daya, durasi sesi, serta aspek penyimpanan dan keamanan

data perlu diperhatikan. Oleh karena itu, pemanfaatan *Google Colab* harus disesuaikan dengan skala dan kompleksitas penelitian ETL yang dilakukan [50].

2.9.5 Dataset

Dataset adalah sekumpulan data yang terorganisir dan disusun dalam format tertentu, biasanya berupa tabel yang terdiri dari baris (data atau entri) dan kolom (atribut atau fitur). Dataset digunakan sebagai sumber informasi dalam proses analisis data, pelatihan model *machine learning*, dan penelitian [51]. Dataset digunakan dalam berbagai bidang seperti ilmu komputer, statistik, ilmu data, dan pembelajaran mesin. Mereka digunakan untuk melatih dan menguji model atau algoritma pembelajaran mesin. Dataset dapat ditemukan dalam berbagai sumber, termasuk basis data, perangkat lunak, situs *web*, repositori publik, dan penelitian ilmiah [52].

Dataset merupakan kumpulan data yang telah disusun supaya lebih mudah untuk di analisa. Dataset dapat digunakan untuk menjawab pertanyaan, membuat prediksi, dan membuat model. Kualitas kumpulan data sangat penting untuk keberhasilan *machine learning* atau ilmu data apa pun [53]. Dataset memiliki peran yang sangat penting karena kualitas, jumlah, dan keberagaman data akan sangat mempengaruhi hasil analisis yang dilakukan.

2.9.6 Roboflow

Roboflow merupakan platform *web* yang berfungsi untuk pengelolaan dataset secara komprehensif, mulai dari pengumpulan hingga optimasi data untuk pengembangan model *computer vision*. Platform ini menyediakan berbagai alat yang mempermudah proses *preprocessing*, seperti *augmentasi data*, pengubahan

ukuran gambar, anotasi, dan pengelolaan dataset secara kolaboratif [54]. *Roboflow* juga di kenal sebagai platform yang memberikan kemudahan dalam pengumpulan dan pengelolaan dataset dalam konteks pengolahan citra dan *computer vision* [55].

Pada tahap *preprocessing* dan pengelolaan dataset, citra motif batik diberikan label berdasarkan jenis motifnya masing-masing agar memudahkan proses identifikasi dan pengelompokan pola tekstur pada penelitian. Proses pengolahan dataset dilakukan menggunakan aplikasi *Roboflow* karena mudah diakses melalui *browser* serta menyediakan fitur *preprocessing* dan *augmentasi* citra yang mendukung pengolahan dataset sebelum dilakukan ekstraksi fitur tekstur dan *clustering* menggunakan algoritma HDBSCAN [56].

2.9.7 *Augmentasi data*

Augmentasi data adalah teknik yang digunakan dalam *machine learning* untuk memperbanyak data dari dataset asli yang berguna untuk meningkatkan akurasi dan menghindari *overfitting* pada model yang kita bangun [57]. *Augmentasi* data dilakukan untuk menambah variasi dataset motif batik sehingga proses *clustering* dapat menghasilkan pengelompokan yang lebih optimal. Pada penelitian ini, augmentasi diterapkan menggunakan metode *Outputs* per training example: 7, yaitu: flip, rotation, 90° rotate, brightness, exposure, untuk menghasilkan variasi citra dari data asli tanpa menghilangkan karakteristik utama motif batik. Proses augmentasi ini bertujuan untuk meningkatkan keberagaman data, memperkuat representasi fitur tekstur, serta membantu algoritma HDBSCAN dalam mengenali pola kemiripan antar motif batik Rokan Hulu dengan lebih baik [58].

Augmentasi data merupakan proses memodifikasi atau memanipulasi suatu citra sehingga citra asli dalam bentuk yang telah disiapkan berubah bentuk dan posisinya. *Augmentasi data* bertujuan untuk memungkinkan mesin belajar dan mengenali dari gambar yang berbeda, dan dapat digunakan untuk merekonstruksi data [59].

2.10 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [60].

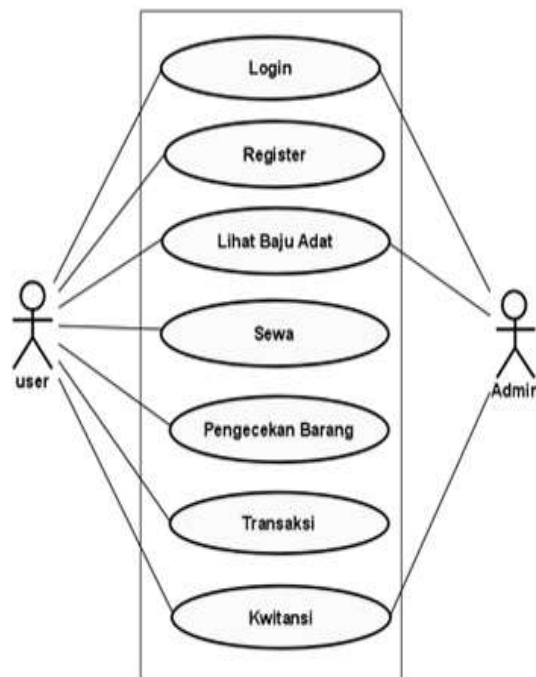
UML (*Unified Modeling Language*) adalah bahasa untuk visualisasi, spesifikasi, membangun sistem perangkat lunak, serta dokumentasi, UML menyediakan model-model yang tepat, tidak ambigu, dan lengkap, secara khusus UML menspesifikasi langkah-langkah penting dalam pengembangan keputusan analisis, perancangan, serta implementasi dalam sistem perangkat lunak [61].

Berikut beberapa *diagram-diagram* pada UML (*Unified Modeling Language*) [61] :

2.10.1 Use Case Diagram

Use case diagram merupakan visualisasi dari beberapa komponen, seperti *actor*, *use case*, dan relasi antar komponen. Beberapa simbol atau notasi digunakan dalam penggambaran fungsionalitas sebuah sistem dalam *use case diagram*. Melalui *use case diagram*, dapat membantu analisis dalam penyusunan kebutuhan (*requirement*) pengembangan sistem. *Use case diagram*

dipakai untuk menjelaskan perancangan sistem kepada *user* dan melakukan perancangan semua fitur yang ada pada sistem yang akan dibangun [60]. *Use case diagram* di sebut juga pemodelan perilaku (*behavior*) dari sistem informasi yang akan diimplementasikan. Interaksi antara satu atau lebih aktor dalam sistem informasi yang akan dibuat dapat digambarkan dengan *use case diagram* [62].

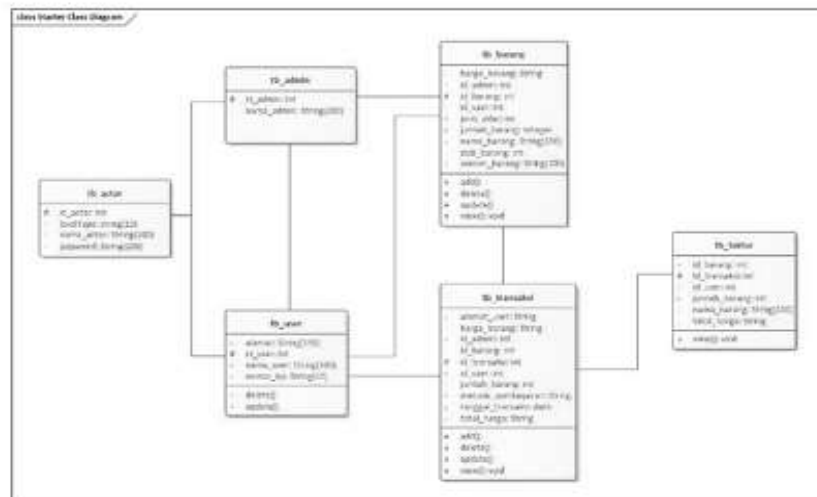


Gambar 2. 8 *use case diagram* [62].

2.10.2 *Class Diagram*

Class diagram adalah *diagam* yang menampilkan beberapa kelas dan paket paket yang ada dalam sistem atau perangkat lunak yang sedang kita gunakan. Selain itu, *class diagram* memberikan gambaran sistem atau perangkat lunak dalam bentuk *diagram* statis, serta hubungan yang ada di dalamnya [63]. *Class diagram* merupakan gambaran struktur sistem dari segi pendefinisian kelas- kelas yang akan dibuat untuk membangun sistem disebut dengan *Class Diagram* yang

terdiri dari atribut dan operasi. Tujuan dari *Class diagram* yaitu membuat program dapat berhubungan antara dokumentai perancangan dan perangkat lunak [60].

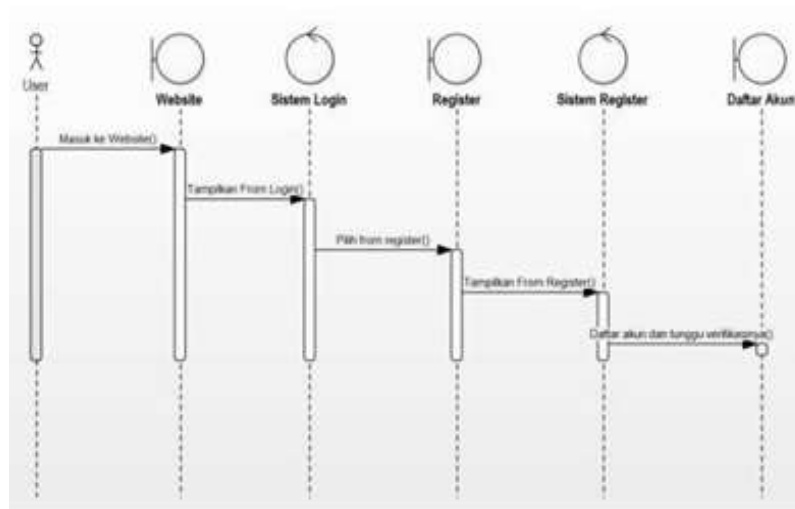


Gambar 2. 9 Class diagram [62].

2.10.3 Sequence Diagram

Sequence Diagram menggambarkan pesan (*message*) yang melewati antar *use case* setiap waktu. *Sequence diagram* memvisualisasikan semua objek yang berkaitan dalam sebuah *use case*. Pendapat lain menyatakan bahwa *sequence diagram* merepresentasikan kolaborasi yang dinamis antar beberapa objek dan memperlihatkan rangkaian pesan yang dikirimkan antar objek dan juga interaksi yang terjadi antar objek dalam sistem yang dibangun [60].

Objek pada *use case* digambarkan dengan *diagram Sequence* dengan waktu hidup objek dan pesan yang dideskripsikan sehingga dapat dikirimkan dan diterima antar objek. Gambaran *sequence diagram* dibuat sebanyak pendefinisian pada *use case*. Semakin banyak *use case* semakin banyak juga *Sequence diagram* yang dibuat [62].



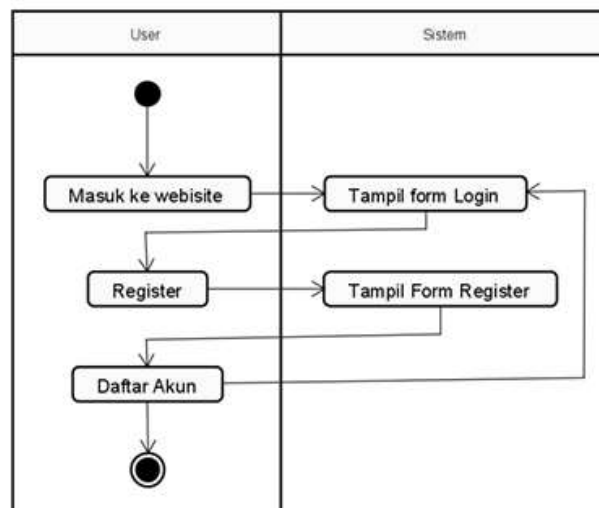
Gambar 2. 10 Sequence Diagram [62].

2.10.4 Activity Diagram

Activity Diagram adalah sebuah cara untuk memodelkan aliran kerja (*workflow*) dari *use case* dalam bentuk grafik, diagram ini menunjukkan langkah-langkah di dalam aliran kerja, titik-titik keputusan didalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas, dan objek-objek yang digunakan dalam aliran kerja. *Activity diagram* juga memvisualisasikan proses-proses paralel yang terjadi ketika sistem dieksekusi. Tahapan atau langkah-langkah yang terjadi di dalam sistem digambarkan dalam diagram ini. Setiap *use case* minimal terdapat satu *activity diagram*. *Activity diagram* dirancang berdasarkan satu atau beberapa *use case* yang ada pada *use case diagram*. *Activity diagram* merepresentasikan proses yang berjalan pada sebuah sistem, sementara *use case* merepresentasikan bagaimana *actor* memakai sistem untuk melakukan aktivitas [60].

Diagram aktivitas menggambarkan aliran aktivitas yang berbeda dalam sistem yang dirancang, bagaimana setiap aliran dimulai, keputusan yang dapat

terjadi, dan bagaimana itu berakhir. Suatu aktivitas dapat dilakukan dengan satu atau lebih menggambarkan *use proses case*. Aktivitas yang sedang berjalan [62].

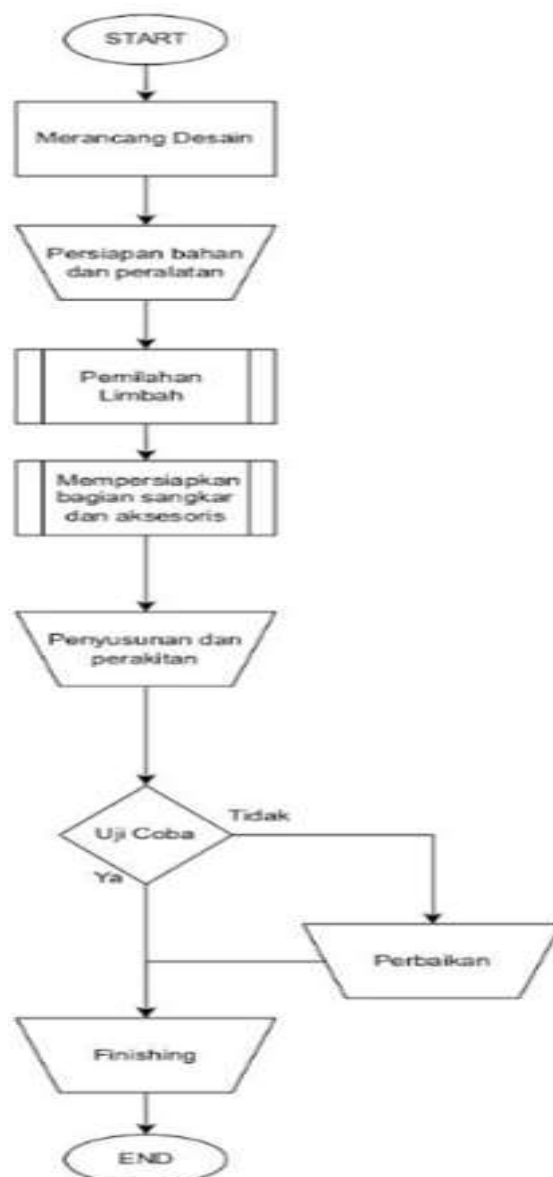


Gambar 2. 11 Activity Diagram [62].

2.10 Flowchart

Flowchart adalah gambaran berupa grafik yang memiliki urutan suatu proses atau langkah-langkah secara sistematis untuk menjalankan suatu program. *Flowchart* dapat memberi gambaran untuk dilakukan proses analisis, perancangan, pengkodean dalam memecahkan masalah yang lebih terperinci dalam proses operasional suatu kegiatan. *Flowchart* umumnya digunakan untuk memudahkan penyelesaian masalah saat dilakukan *evaluasi* lebih lanjut [64]. *Flowchart* atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan *algoritma* atau langkah-langkah instruksi yang berurutan dalam sistem. seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada *programmer* [65].

Flowchart dapat didefinisikan sebagai sebuah gambaran yang menjelaskan proses yang akan dilihat atau dikaji. Selain itu, *flowchart* biasanya digunakan untuk merencanakan tahapan suatu kegiatan. Jadi, *flowchart* atau bagan alur merupakan metode untuk menggambarkan tahap-tahap penyelesaian masalah (prosedur) beserta liran data dengan simbol-simbol standar yang mudah dipahami [66].



Gambar 2. 12 Flowchart [64].

2.11 Draw io

Draw.io atau yang dikenal sebagai *diagrams.net* merupakan alat perancangan sistem berbasis *web* yang digunakan untuk membuat berbagai jenis *diagram*, seperti *flowchart*, *diagram* konteks, dan *diagram* alur proses. *Tools* ini berfungsi untuk memvisualisasikan alur kerja dan struktur sistem informasi sehingga memudahkan pengembang dalam memahami kebutuhan sistem sebelum tahap implementasi dilakukan [42].

Draw.io adalah sebuah *website* yang didesain khusus untuk menggambarkan *diagram* UML secara *online*. Dimana punya tampilan yang sangat responsive dan terintegrasi dengan layanan penyimpanan file milik *google* yaitu *Google Drive* sehingga *draw.io* menjadi alternatif dalam pembuatan diagram UML dengan waktu yang lebih singkat [67]. *Draw.io* digunakan untuk membuat *flowchat*, *draw.io* berguna untuk merancang *Use Case diagram* maupun *activity diagram* [68].

2.12 Pengujian Aplikasi

2.12.1 Black Box Testing

Metode *Black Box Testing* merupakan pengujian untuk menunjukkan kesalahan pada sistem aplikasi seperti kesalahan pada fungsi sistem aplikasi, serta menu aplikasi yang hilang. Jadi *Black Box testing* merupakan metode uji fungsionalitas sistem aplikasi. Dalam melakukan pengujian menggunakan masukan data acak dengan tujuan untuk mendapatkan hasil yang pasti. Dikatakan pasti artinya bila salah, maka ditolak oleh sistem informasi atau data *input*

tersebut tidak dapat disimpan dalam database, sedangkan bila data input benar maka dapat diterima / masuk di database sistem informasi [69].

Black box testing merupakan pengujian pada fungsi operasional dalam perangkat lunak. *Black box testing* merupakan teknik pada *testing* perangkat lunak yang bertujuan menguji spesifikasi fungsional perangkat lunak. Teknik *black box testing* ini tidak melihat struktur kontrol dari perangkat lunak dalam penerapannya sehingga berfokus pada informasi di domain perangkat lunak. Berdasarkan kedua pengertian tentang pengujian tersebut maka *black box testing* merupakan salah satu teknik pengujian perangkat lunak yang hanya berfokus pada pengujian fungsi operasional pada perangkat lunak [70].

Pengujian *black box testing* digunakan untuk memastikan fungsi-fungsi sistem pada perangkat lunak dapat berjalan dengan sesuai dengan yang direncanakan. Pengujian *black box testing* menguji fungsionalitas tanpa perlu mengetahui detail internal kode atau struktur programnya, pengujian ini membantu mengidentifikasi masalah tanpa memperhatikan implementasi internal [71].

2.13 Penelitian Terdahulu

Adapun penelitian terkait dapat dilihat pada tabel berikut :

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Metode	Hasil Penelitian
1	Fitrianto dkk (2024)	Perbandingan Algoritma HDBSCAN dan Agglomerative Hierarchical Clustering dalam	HDBSCAN dan Agglomerative Hierarchical Clustering	Hasil penelitian menunjukkan bahwa algoritma HDBSCAN lebih efektif dalam mengelompokkan data dan mampu

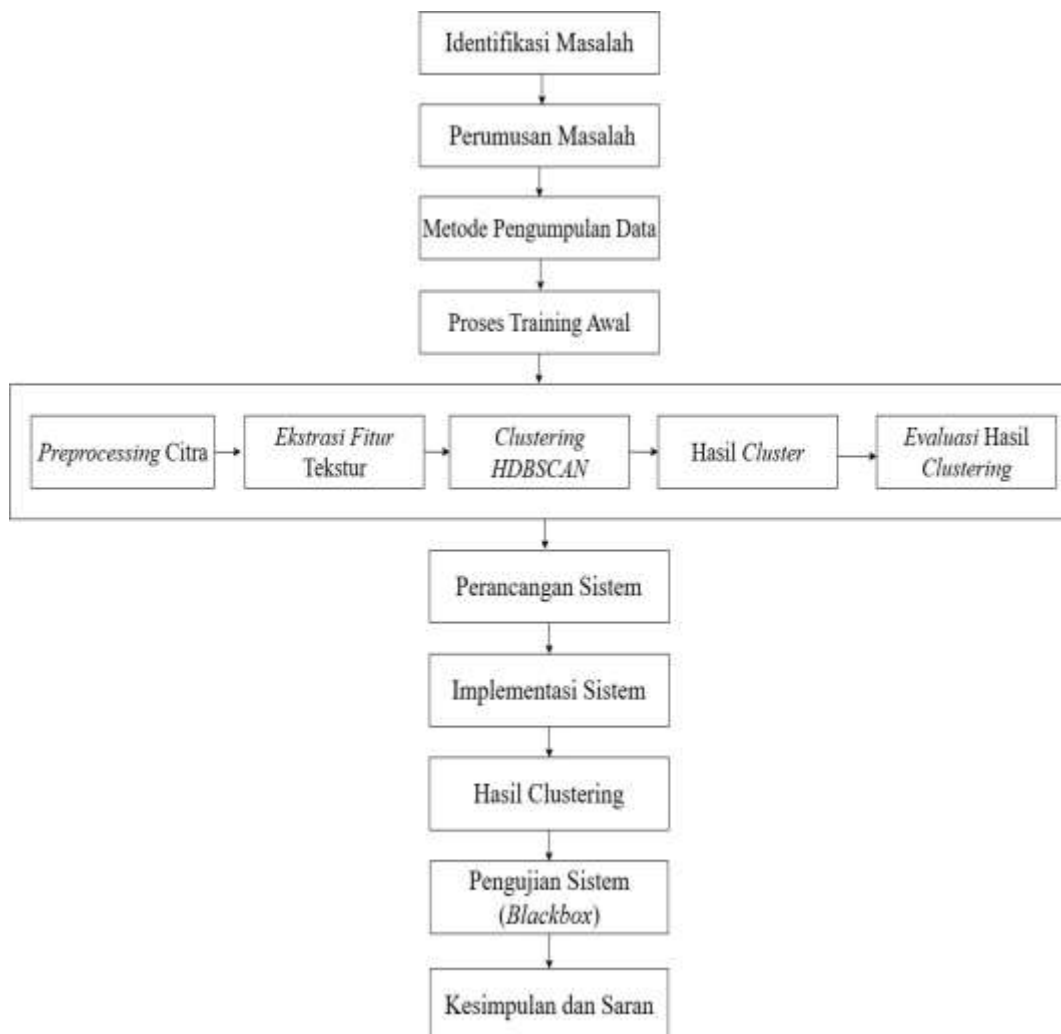
		Klasterisasi Data yang Mengandung Pencilan		mendeteksi pencilan (<i>outlier</i>) secara otomatis dibandingkan metode <i>hierarchical clustering</i> .
2	Anugrah, Wihardi, & Megasari (2025)	<i>Unsupervised Clustering of Handwritten Essay Answer Images Using Vision Transformer</i>	<i>Ekstraksi fitur (ResNet-50, Vision Transformer, Tr-OCR) + K-Means & HDBSCAN</i>	Penelitian menunjukkan bahwa HDBSCAN mampu menghasilkan kualitas <i>cluster</i> yang baik dengan nilai <i>Silhouette Score</i> sebesar 0,772, sehingga efektif dalam pengelompokan citra.
3	Stewart & Al-Khassaweneh (2022)	<i>An Implementation of the HDBSCAN Clustering Algorithm</i>	<i>Implementasi algoritma HDBSCAN</i>	Penelitian menunjukkan bahwa HDBSCAN mampu menemukan <i>cluster</i> pada data dengan kepadatan berbeda serta efektif dalam mendeteksi <i>noise</i> sehingga menghasilkan <i>clustering</i> yang optimal.
4	Liang dkk (2022)	<i>An Effective Clustering Algorithm for the Low-Quality Image of Integrated Circuits via High-Frequency Texture Components Extraction</i>	<i>Ekstraksi fitur tekstur + HDBSCAN</i>	Penelitian menunjukkan bahwa kombinasi <i>ekstraksi fitur</i> tekstur dan HDBSCAN mampu meningkatkan akurasi <i>clustering</i> serta efektif dalam mendeteksi <i>cluster</i> dan <i>noise</i> pada citra.
5	Zain dkk (2024)	<i>Clustering Motif Batik Menggunakan Ekstraksi Fitur dan Metode Clustering</i>	<i>Ekstraksi fitur citra + clustering</i>	Penelitian menunjukkan bahwa <i>ekstraksi fitur</i> citra mampu merepresentasikan karakteristik motif batik sehingga dapat digunakan dalam proses <i>clustering</i>

				untuk mengelompokkan pola secara lebih akurat.
6	Naufal Alif Rahmania Akmal, dkk (2025)	Pengembangan Sistem Pendukung <i>Investigasi Serangan Siber</i> Berdasarkan <i>Log Server Web</i> Menggunakan <i>Algoritma HDBSCAN</i>	<i>algoritma HDBSCAN</i>	Penelitian menunjukkan bahwa <i>HDBSCAN</i> mampu menghasilkan 802 <i>cluster</i> dan 325 <i>noise</i> dengan tingkat keberhasilan <i>clustering</i> sebesar 99,6%. <i>Noise</i> yang dihasilkan berhasil mengidentifikasi pola serangan seperti <i>SQL Injection</i> , <i>XSS</i> , <i>path traversal</i> , dan <i>RCE</i> . <i>Evaluasi</i> menggunakan <i>Silhouette Score</i> (0,0736), <i>Davies-Bouldin Index</i> (2,4756), dan <i>Calinski-Harabasz Score</i> (3473,67) menunjukkan bahwa struktur <i>cluster</i> cukup baik untuk mendukung deteksi anomali pada <i>log server web</i> .
7	Fitri Dwi Handayani & Isnaini Rosyida (2023)	<i>Clustering Review</i> Pengguna Aplikasi Zenius pada Layanan Google Play Store Menggunakan Metode DBSCAN dan HDBSCAN	<i>Clustering algoritma DBSCAN dan HDBSCAN</i> ,	Penelitian menunjukkan bahwa <i>HDBSCAN</i> lebih unggul dibandingkan DBSCAN dengan nilai <i>Silhouette Coefficient</i> sebesar 0,2941, sedangkan DBSCAN 0,1310. Metode ini mampu mengelompokkan ulasan pengguna serta mengidentifikasi pola keluhan dan ulasan positif.

BAB 3

METODOLOGI PENELITIAN

Tahapan penelitian yang akan dilakukan dan penyelesaian terhadap permasalahan dengan menerapkan Algoritma *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN). untuk pengelompokan motif batik, adapun tahapan metodologi yang dilakukan selama penelitian dapat dilihat pada Gambar 3.1 yang mana merupakan proses yang dimulai dari studi literatur hingga diperoleh kesimpulan.



Gambar 3. 1 Tahapan Metodologi Penelitian

3.1 Identifikasi Masalah

Identifikasi masalah merupakan langkah awal dalam melaksanakan penelitian ini. Pada tahap ini di jelaskan latar belakang dari permasalahan yang di temukan, dengan mengambil acuan dari berbagai jurnal penelitian yang pernah dilakukan sebelumnya, rumusan masalah, tujuan penelitian, dan batasan masalah agar masalah yang di kaji pada penelitian ini tidak berubah dari yang telah di terapkan sebelumnya.

a. Data Primer

Penelitian ini dilakukan untuk mendapatkan data primer dengan cara mengambil sampel dari beberapa data, dan persyaratan dalam *Clustering* Motif Batik Menggunakan Metode HDBSCAN.

b. Pengumpulan Data Sekunder

Peneliti mengumpulkan data dan informasi melalui studi pustaka yang bersifat sekunder, yaitu ada data-data yang diperoleh melalui *google* referensi tentang *Clustering* Motif Batik.

3.2 Perumusan Masalah

Berdasarkan hasil dari tahapan pengamatan pendahuluan sebelumnya, maka tahapan selanjutnya adalah tahapan perumusan masalah, permasalahan dalam penelitian ini adalah bagaimana menyiapkan data citra motif batik Rokan Hulu melalui tahap *preprocessing*, melakukan ekstraksi fitur tekstur, serta menerapkan *algoritma Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN) untuk proses pengelompokan, serta menganalisis dan mengevaluasi hasil *clustering* yang diperoleh.

3.3 Metode Pengumpulan Data

Pengumpulan data citra motif batik merupakan tahapan yang bertujuan untuk memperoleh data dan informasi yang berkaitan dengan penelitian ini. Pada tahap pengumpulan data ini juga dilakukan pengumpulan seluruh data citra motif batik yang akan digunakan sebagai bahan dalam proses pengolahan dan analisis menggunakan metode *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN) berbasis ekstraksi fitur tekstur. Data yang diperoleh nantinya akan digunakan sebagai dataset dalam proses *clustering* motif batik. Pengumpulan data tersebut dilakukan dengan beberapa cara, antara lain:

1. Observasi

Penulis melakukan pengamatan secara langsung ke mitra usaha Alifa Batik Rokan Kecamatan Ujung Batu, Kabupaten Rokan Hulu, Provinsi Riau, observasi ini dilakukan dengan mengamati berbagai jenis motif batik yang digunakan sebagai data penelitian, khususnya yang berkaitan dengan pola dan tekstur pada citra batik.

2. Wawancara

Wawancara dilakukan oleh penulis dengan tanya jawab langsung kepada pemilik mitra usaha Alifa Batik Rokan Kecamatan Ujung Batu, galeri batik, dinas pariwisata dan kebudayaan, dan perancang batik Kabupaten Rokan Hulu, Provinsi Riau.

3. Studi Dokumen

Teknik pengumpulan data ini dilakukan dengan mengandalkan dokumen sebagai sumber *data algoritma* pengumpulan data ini

dilakukan untuk melengkapi penelitian, adapun gambar yang dikumpulkan berupa data citra motif batik yang akan digunakan sebagai dataset dalam proses ekstraksi fitur tekstur.

Tabel 3. 1 Dokumen Batik

NO	Sumber Data	Banyaknya Data
1	Sicuriang	1.000
2	Tugu Ratik Togak	1.000
3	Sesuli Banja	1.000
4	Suluo Kekaik	1.000
Total		4.000

4. Studi Kepustakaan

Tahapan ini dilakukan dengan mengumpulkan dan menganalisis literature yang *relevan* mengenai *data mining*, *clustering*, *algoritma HDBSCAN*, serta ekstraksi fitur tekstur. Informasi dan pembahasan mengenai metode tersebut diperoleh dari berbagai sumber ilmiah seperti jurnal ilmiah, paper, artikel, buku, serta *referensi* akademik lainnya yang berkaitan dengan penelitian ini.

3.4 Proses Training Awal

Pada tahap ini dilakukan proses training awal mengenai *Clustering* motif batik menggunakan metode *HDBSCAN*. Selanjutnya, dilakukan beberapa tahapan pengolahan data sebagai berikut:

3.4.1 *Preprocessing* Citra

Tahapan *preprocessing* citra merupakan proses awal yang bertujuan untuk menyiapkan data citra sebelum dilakukan *ekstraksi fitur* dan proses *clustering*. Pada tahap ini, citra motif batik yang telah dikumpulkan akan melalui beberapa langkah pengolahan seperti pengubahan ukuran (*resize*), *konversi* warna (misalnya dari RGB ke grayscale), serta normalisasi citra untuk meningkatkan kualitas dan konsistensi data. Selain itu, dilakukan juga proses pengurangan *noise* (*noise reduction*) guna meminimalkan gangguan pada citra yang dapat mempengaruhi hasil *analisis*. Setelah melalui proses tersebut, citra menjadi lebih seragam dan siap untuk dianalisis lebih lanjut. *Preprocessing* ini penting dilakukan agar *fitur* tekstur yang diekstraksi benar-benar *merepresentasikan* pola asli dari motif batik tanpa terpengaruh oleh *variasi* pencahayaan, ukuran, maupun gangguan lainnya. Dengan demikian, hasil *clustering* menggunakan metode *HDBSCAN* dapat menjadi lebih akurat dalam mengelompokkan motif batik berdasarkan kemiripan *karakteristik* teksturnya.

3.4.2 Ekstraksi Fitur Tekstur

Ekstraksi fitur tekstur pada penelitian ini dilakukan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM), *Local Binary Pattern* (LBP), Histogram HSV, dan Gabor Filter, di mana GLCM digunakan untuk menganalisis hubungan spasial antar *piksel* melalui parameter seperti *contrast*, *correlation*, *energy*, dan *homogeneity*. LBP digunakan untuk menangkap pola tekstur lokal dengan membandingkan nilai intensitas *piksel* terhadap *piksel* tetangganya, sehingga menghasilkan representasi fitur yang lebih komprehensif sebagai *input*

dalam proses *clustering*. Histogram HSV digunakan untuk memperoleh informasi distribusi warna pada citra berdasarkan komponen hue, saturation, dan value. Sedangkan gabor filter digunakan untuk menangkap pola tekstur berdasarkan frekuensi dan orientasi tertentu sehingga mampu merepresentasikan karakteristik motif batik secara lebih baik.

3.4.3 *Clustering HDBSCAN*

Clustering pada penelitian ini dilakukan menggunakan *algoritma Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)*, yaitu metode berbasis kepadatan yang mampu mengelompokkan data dengan *variasi densitas* tanpa menentukan jumlah *cluster* di awal, melalui tahapan pembentukan *core distance*, perhitungan *mutual reachability distance*, pembangunan *minimum spanning tree*, pembentukan *hierarki cluster*, serta *ekstraksi cluster* yang stabil, dengan memanfaatkan parameter *min_cluster_size* dan *min_samples* untuk menghasilkan *label cluster* dan mengidentifikasi data *noise*.

3.4.4 *Hasil Cluster*

Hasil *cluster* pada penelitian ini merupakan keluaran dari proses pengelompokan menggunakan *algoritma HDBSCAN* yang berupa *label cluster* untuk setiap data citra motif batik, di mana data dengan *karakteristik* serupa akan berada dalam satu kelompok yang sama, sedangkan data yang tidak memenuhi kriteria kepadatan akan *diklasifikasikan* sebagai *noise (cluster -1)*, sehingga hasil tersebut dapat digunakan untuk *analisis pola* dan *evaluasi kualitas clustering*.

3.4.5 *Evaluasi Hasil Clustering*

Evaluasi hasil *clustering* pada penelitian ini dilakukan untuk mengukur kualitas pengelompokan data motif batik yang dihasilkan oleh *algoritma HDBSCAN* dengan menggunakan *metrik evaluasi* seperti *Silhouette Score*, yang menilai tingkat kesamaan data dalam satu *cluster* dibandingkan dengan *cluster* lainnya, sehingga dapat diketahui seberapa baik struktur *cluster* yang terbentuk, semakin baik nilai evaluasi yang diperoleh maka semakin baik kualitas cluster yang dihasilkan.

3.5 **Perancangan Sistem**

Perancangan Sistem di gunakan untuk mentransformasikan kebutuhan detail menjadi kebutuhan yang lengkap, dokumen desain fokus pada bagaimana dapat memenuhi fungsi-fungsi yang di butuhkan, tahapan perancangan sistem terdiri dari :

3.5.1 **Proses Pemodelan (*Modeling Process*)**

Rancangan struktur menu ini kita perlu untuk memberikan gambaran terhadap menu-menu atau *fitur* pada sistem yang akan beroperasi dan mengilustrasi aktifitas yang di lakukan. Cara yang digunakan adalah dengan menggunakan *UML (Unified Modeling Language)*.

3.5.2 **Perancangan Antar Muka (*Interface*)**

Dalam mempermudah komunikasi antara sistem dengan pengguna, maka perlu dirancang antar muka (*interface*), dalam perancangan *interface* hal terpenting yang harus dilakukan ialah bagaimana menciptakan tampilan yang baik dan mudah dimengerti oleh pengguna.

3.6 Implementasi Sistem

Setelah proses clustering selesai dilakukan, hasil penelitian diimplementasikan ke dalam aplikasi berbasis web. Aplikasi dirancang agar pengguna dapat mengunggah citra batik dan memperoleh hasil pengelompokan berdasarkan kemiripan tekstur yang dimiliki. Implementasi sistem terdiri atas beberapa bagian, yaitu:

1. Halaman Beranda.
2. Halaman Proses *Clustering*.
3. Halaman Informasi Filosofi Batik.
4. Halaman statistik dataset

Sistem dikembangkan menggunakan bahasa pemrograman *Python* serta *framework web* yang mendukung integrasi dengan model *clustering*.

Beberapa komponen pendukung yang memiliki peran yang sangat penting dalam implementasi sistem diantaranya adalah perangkat keras (*hardware*) dan perangkat lunak (*software*) adapun spesifikasi dari perangkat keras (*hardware*) dan perangkat lunak (*software*) yang di gunakan sebagai berikut :

1. Perangkat keras (*hardware*), antara lain :

Prosesor	: 12th Gen Intel(R) Core(TM) i3-1215U (1.20 GHz)
Memory (RAM)	: 12 GB RAM
System type	: 64-bit operating system, x64-based processor
Harddisk	: 500 GB
2. Perangkat Lunak (*Software*), antara lain :

Sistem operasi	: Windows 11
----------------	--------------

3.7 Hasil *Clustering*

Hasil *clustering* merupakan keluaran yang dihasilkan oleh sistem setelah proses pengelompokan data citra motif batik menggunakan metode HDBSCAN selesai dilakukan. Pada tahap ini, sistem menampilkan informasi berupa label cluster dari setiap citra motif batik yang telah diproses berdasarkan karakteristik fitur teksturnya. Hasil pengelompokan ditampilkan dalam bentuk tabel dan visualisasi sehingga pengguna dapat melihat kelompok motif batik yang memiliki kemiripan karakteristik. Selain itu, sistem juga menampilkan data yang teridentifikasi sebagai *noise* apabila terdapat citra yang tidak termasuk ke dalam cluster tertentu. Hasil *clustering* ini selanjutnya digunakan sebagai bahan dalam proses pengujian sistem untuk memastikan seluruh fungsi aplikasi berjalan sesuai dengan yang diharapkan.

3.8 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan adalah *Black Box Testing*. Pengujian dilakukan pada setiap fitur sistem tanpa melihat struktur kode program. Hasil pengujian digunakan untuk mengetahui apakah sistem telah berjalan sesuai dengan rancangan yang dibuat.

3.8.1 *Black Box Testing*

Ada beberapa langkah yang harus dilakukan dan diperhatikan pada cara kerja *Black Box Testing* yaitu sebagai berikut :

- a. Memahami kebutuhan dan spesifikasi perangkat lunak atau aplikasi yang akan diuji

- b. Menentukan *input* yang akan digunakan dalam pengujian
- c. Menentukan *output* yang akan dihasilkan dalam pengujian
- d. Menerapkan *test case* atau skenario buatan untuk menguji *input* yang sudah dipilih
- e. Melakukan pengujian
- f. Melakukan *review* dan *evaluasi*

Yang akan diuji dengan menggunakan *black box testing* adalah semua fungsi menu dan tombol pada aplikasi yang dibangun, menguji apakah semua berjalan dengan sesuai dengan *output* yang diinginkan.

Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak, pengujian *black box testing* bertujuan untuk menemukan fungsi yang tidak benar kesalahan antarmuka kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi.

3.9 Kesimpulan dan Saran

Tahap terakhir adalah penarikan kesimpulan berdasarkan hasil penelitian yang telah dilakukan.

Kesimpulan berisi ringkasan hasil penerapan *algoritma Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN) berbasis ekstraksi fitur tekstur dalam proses clustering motif batik serta hasil analisis terhadap kualitas cluster yang terbentuk. Sementara itu, saran diberikan sebagai rekomendasi untuk pengembangan penelitian selanjutnya, seperti

penggunaan metode ekstraksi fitur yang lebih beragam, optimasi parameter HDBSCAN, atau penambahan jumlah dan variasi dataset motif batik.