

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini:

Pihak Pertama

Nama : Muchamad Rizqi Pratama
NIM : 22090104
Program Studi : Teknik Informatika

Pihak Kedua

Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Status : Dosen Tetap
NIDN : 0623108801
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Tk.I /III-d

Pada hari ini Rabu tanggal 15 April 2026, telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua dan menyelesaikannya paling lambat tanggal 31 Juli 2026. Pihak Pertama menyatakan tidak akan menggunakan jasa joki dalam penyelesaian Skripsi tersebut. Apabila Pihak Pertama tidak memenuhi kesepakatan ini, maka Pihak Kedua berhak untuk memutuskan proses bimbingan Skripsi. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 15 April 2026

Pihak Pertama

Pihak Kedua



Muchamad Rizqi Pratama

Muhammad Fikri Hidayattullah, S.T., M.Kom.

Mengetahui,

Ketua Program Studi D IV Teknik Informatika



Dyah Apriliani, S.T., M.Kom.

NIPY: 09.015.225

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini:

Pihak Pertama

Nama : Muchamad Rizqi Pratama
NIM : 22090104
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Taufiq Abidin, S.Pd., M.Kom.
Status : Dosen
NIDN : 0603088305
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata /III/C

Pada hari ini, Rabu, 1 April 2026, telah terjadi kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama, dengan syarat bahwa Pihak Pertama wajib melakukan bimbingan skripsi minimal 8 kali kepada Pihak Kedua, dengan hari dan waktu yang konsisten.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 1 April 2026

Pihak Pertama



Muchamad Rizqi Pratama

Pihak Kedua



Taufiq Abidin, S.Pd., M.Kom.
NIPY: 06.014.184

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik informatika



Dyah Apriyanti, S.T., M.Kom.
NIPY: 09.015.225

Lampiran 2. Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini, pemegang hak cipta:

Nama : Muchamad Rizqi Pratama
Kewarganegaraan : Indonesia
Alamat : Jl. Perkatut RT 03/ RW 01, Desa Slawi Kulon, Kecamatan
Slawi, Kabupaten Tegal, Provinsi Jawa Tengah, 52419
Nama : Muhammad Fikri Hidayattullah
Kewarganegaraan : Indonesia
Alamat : Jalan Glatik No. 68, Kelurahan Randugunting, Kecamatan Tegal
Selatan, Kota Tegal 52125
Nama : Taufiq Abidin, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Kaligangsa Asri Timur VIII no 33 RT 3 RW 7 Kel. Kaligangsa
Kec. Margadana Kota Tegal 52147

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:

Berupa : Program Komputer
Berjudul : Sibatikgal: Sistem Informasi Mobile Berbasis Kecerdasan Buatan
untuk Identifikasi Motif Batik Tegal

Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);

- Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
- Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
- Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
- Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
- Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.

3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:

- a. Permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau

- b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia RI dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam perkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tanggal 23 Juni 2026

 Muchlisau Rizqi Ratama
 Pemegang Hak Cipta*


 Muhammad Fikri Hidayatullah
 Pemegang Hak Cipta*


 Taufiq Abidin
 Pemegang Hak Cipta*

Pemegang Hak Cipta*

*Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 3. Surat Pengalihan HAK Cipta

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini:

- 1 Nama : Muchamad Rizqi Pratama
Kewarganegaraan : Indonesia
Alamat : Jl. Perkutut RT 03/ RW 01, Desa Slawi Kulon, Kecamatan Slawi, Kabupaten Tegal, Provinsi Jawa Tengah, 52419
- 2 Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jalan Glatik No. 68, Kelurahan Randugunting, Kecamatan Tegal Selatan, Kota Tegal 52125
- 3 Nama : Taufiq Abidin, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Kaligangsa Asri Timur VIII no 33 RT 3 RW 7 Kel. Kaligangsa Kec. Margadana Kota Tegal 52147

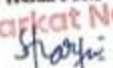
Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M) Universitas Harkat Negeri
Alamat : Jl. Mataram No 9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku pemegang Hak Cipta berupa Program Komputer dengan Judul "Sibatikgal: Sistem Informasi Mobile Berbasis Kecerdasan Buatan untuk Identifikasi Motif Batik Tegal" untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia

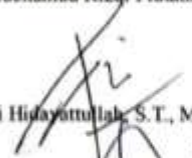
Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 23 Juni 2026

Pemegang Hak Cipta
Ur. Ketua P3M
Harkat Negeri

(Sharfina Febbi Handayani, M.Kom.)

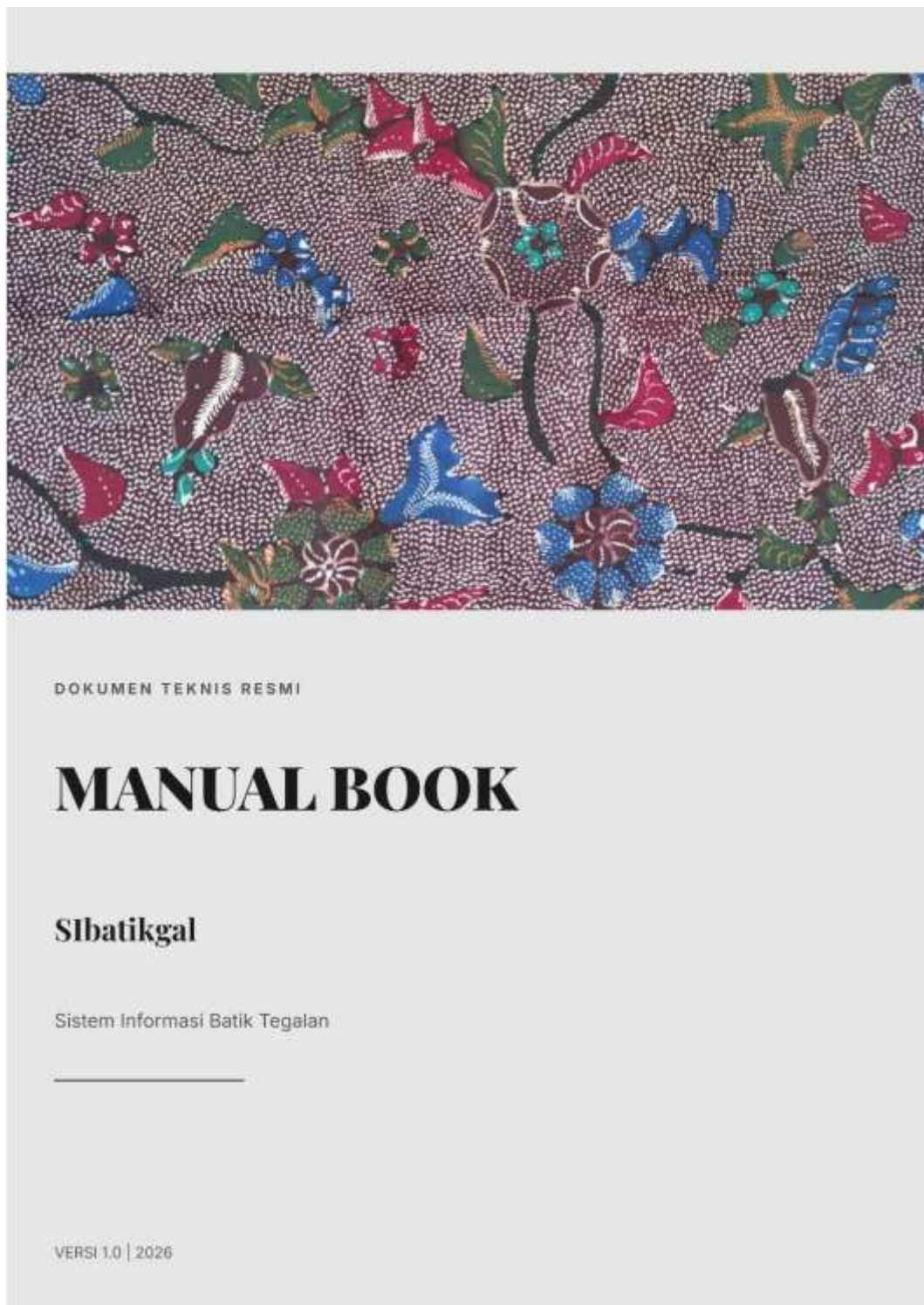


(Muchamad Rizqi Pratama)


(Muhammad Fikri Hidayattullah, S.T., M.Kom.)


(Taufiq Abidin, S.Pd., M.Kom.)

Lampiran 4. Manual Book



PENDAHULUAN

1. Tujuan Pembuatan Dokumen

Dokumen *Manual Book* (Panduan Pengguna) ini disusun dengan tujuan utama untuk memberikan petunjuk teknis dan panduan operasional langkah demi langkah yang sistematis bagi pengguna dalam mengoperasikan aplikasi Sibatikgal. Melalui dokumen ini, diharapkan pengguna dapat memahami, menjelajahi, dan memanfaatkan seluruh fitur utama yang tertanam pada sistem secara tepat guna. Panduan ini dirancang sedemikian rupa agar pengguna dapat menjalankan aplikasi dengan performa maksimal, sekaligus meminimalkan terjadinya kesalahan operasional (*human error*) saat sistem sedang digunakan.

2. Nilai Inovasi

Aplikasi Sibatikgal menghadirkan inovasi digitalisasi budaya lokal lewat pemanfaatan teknologi *computer vision* (Kecerdasan Buatan) berbasis Android. Sistem ini mampu mengidentifikasi motif Batik Tegal secara otomatis dan *real-time* hanya melalui jepretan kamera, menjadikannya media edukasi budaya yang modern, interaktif, dan mudah dibawa ke mana saja.

3. Dampak Pemanfaatan Aplikasi Sibatikgal

Dengan digunakannya aplikasi ini, masyarakat luas dan wisatawan dapat dengan mudah mengenali serta memahami nilai filosofis di balik setiap motif tradisional Batik Tegal. Dampak luasnya adalah meningkatnya literasi budaya digital, mendukung eksistensi pengrajin batik lokal lewat fitur peta sebaran, serta menjaga warisan budaya Tegal agar tetap lestari di era digital.

PEMBAHASAN APLIKASI

1. INSTALASI DAN KEBUTUHAN SISTEM

Sebelum memulai, pastikan perangkat Anda memenuhi persyaratan berikut untuk dapat menjalankan aplikasi Sibatikgal dengan optimal.

1.1. Persyaratan Sistem

- Sistem Operasi : Android 8.0 atau versi lebih baru
- Perangkat Keras : Handphone dengan kamera aktif
- Koneksi : Koneksi Internet Stabil

1.2. Cara Install Aplikasi

- 1) Pastikan Anda telah mengaktifkan izin "Instalasi dari sumber tidak dikenal" pada perangkat Android Anda (Pengaturan > Keamanan > aktifkan Unknown Sources atau Sumber Tidak Dikenal).
- 2) Unduh file Sibatikgal.apk melalui tautan yang telah disediakan.
- 3) Setelah selesai diunduh, buka file Sibatikgal.apk dari folder unduhan Anda.
- 4) Klik "Install" dan tunggu hingga proses instalasi selesai.
- 5) Aplikasi Sibatikgal siap digunakan.

2. DESKRIPSI FUNGSIONAL APLIKASI SIBATIKGAL

Aplikasi Sibatikgal dirancang untuk membantu mengenali dan mempelajari ragam budaya Batik Tegalan secara mandiri melalui pemindaian motif citra kain yang disesuaikan dengan kebutuhan pengguna. Aplikasi ini dilengkapi fitur klasifikasi otomatis, katalog edukasi ragam hias beserta informasi filosofisnya, pemetaan lokasi melalui peta sebaran, serta galeri visual Batik Art agar proses pengenalan budaya lebih praktis, tepat, dan terukur.

3. PENJALASAN DETAIL FITUR

Untuk mendukung fungsi utama tersebut dan memastikan pengalaman pengguna yang lengkap dan nyaman, Sibatikgal menyediakan serangkaian fitur utama berikut sebelum Anda masuk ke langkah-langkah penggunaannya:

- **Pendaftaran dan Verifikasi Akun**
Proses pendaftaran akun baru yang aman melalui verifikasi email untuk memastikan validitas pengguna.
- **Login Pengguna**
Users dapat melakukan login dengan menggunakan username dan password atau menggunakan opsi login menggunakan Google.
- **Fitur Deteksi Motif Batik**
Proses pengenalan otomatis terhadap motif ikonik Batik Tegalana secara *real-time* menggunakan kamera perangkat atau melalui unggahan foto dari galeri untuk mendeteksi jenis motif beserta persentase akurasi dan user dapat langsung mengarah ke detail motif hasil klasifikasi.
- **Fitur Katalog Ragam Motif Batik**
Halaman edukasi digital yang memuat daftar lengkap motif Batik Tegalana beserta informasi mendalam mengenai sejarah, karakteristik visual, dan nilai filosofis di setiap motifnya.
- **Fitur Peta Sebaran**
Fitur pemetaan interaktif yang menyajikan informasi mengenai lokasi geografis dan persebaran sentra industri serta para pengrajin Batik Tegalana untuk memudahkan pencarian lokasi.
- **Fitur Batik Art**
Ruang galeri visual khusus yang menampilkan dokumentasi, karya seni kreatif, serta informasi tambahan terkait perkembangan kebudayaan Batik Tegalana.

4. PENDAFTARAN DAN LOGIN AKUN

Untuk dapat mengakses fitur fitur Sibatikgal, anda perlu memiliki akun.

4.1. Bagi Pengguna Baru (Registrasi)

Bagi pengguna yang belum memiliki akun, users dapat melakukan registrasi untuk melakukan proses pendaftaran.

- 1) Lengkapi formulir pendaftaran dengan mengisi data atau jika tidak ingin mengisi field dapat menggunakan dengan Google
- 2) Setelah data terisi, klik tombol “Daftar” untuk membuat akun.
- 3) Sistem akan mengirimkan kode OTP ke email yang telah di daftarkan.
- 4) Lakukan verifikasi menggunakan kode OTP.

4.2. Bagi Pengguna Terdaftar (Login)

- 1) Jikas sudah memiliki akun, masukan email dan password pada halaman login.
- 2) Klik tombol “Masuk” atau “Login Menggunakan Google” untuk masuk ke Aplikasi.

4.3. Mengatasi Lupa Password

- 1) Jika anda mengalami lupa password saat login, klik tautan “Lupa kata sandi?” yang terdapat pada halaman login.
- 2) Anda akan diarahkan ke halaman “Lupa kata sandi”. Lalu masukan Alamat email yang terdaftar pada akun anda.
- 3) Klik tombol “Kirim Kode” lalu kode OTP akan dikirimkan ke email anda. Kemudian lakukan verifikasi menggunakan OTP yang dikirimkan melalui email.



- 4) Setelah melakukan verifikasi OTP, anda akan diarahkan ke halaman reset password.
- 5) Pada halaman tersebut, isi kolom “Password baru” dan “Konfirmasi Password baru” lalu klik “Reset Password”.
- 6) Klik tombol “Reset password” untuk menyimpan.
- 7) Sekarang Anda dapat Kembali ke Aplikasi Sibatikgal dan login menggunakan password baru anda

5. HALAMAN BERANDA DAN NAVIGASI UTAMA

Setelah pengguna berhasil melakukan proses login, sistem akan secara otomatis mengarahkan pengguna ke halaman Beranda (Home) sebagai gerbang utama aplikasi Sibatikgal. Halaman ini berfungsi sebagai pusat navigasi untuk mengakses berbagai menu dan informasi seputar Batik Tegalan.



Langkah-langkah untuk memahami dan menggunakan halaman Beranda adalah sebagai berikut:

- 1) **Melihat Informasi Beranda:** Pada halaman ini, pengguna dapat melihat teks sambutan hangat "Sugeng Dalu!", menggunakan fitur pencarian data batik, melihat banner *event* kebudayaan terbaru, serta membaca artikel informasi terkini yang berada di bagian bawah halaman.
- 2) **Menggunakan Navigasi Utama (Bottom Navigation Bar):** Untuk berpindah ke fitur utama lainnya, pengguna cukup menekan ikon menu yang terletak pada bar navigasi bagian bawah layar, yaitu:
 - **Ikon Beranda:** Untuk kembali ke halaman utama aplikasi.
 - **Ikon Galeri:** Untuk membuka katalog ragam motif Batik Tegalan.
 - **Ikon Deteksi:** Untuk masuk ke fitur pemindaian kamera/*computer vision*.
 - **Ikon Peta:** Untuk melihat sebaran lokasi para perajin batik.
 - **Ikon Pengaturan:** Untuk mengelola profil dan konfigurasi akun.

6. KATALOG RAGAM MOTIF BATIK

Fitur ini berisi daftar lengkap koleksi motif tradisional Batik Tegal. Melalui halaman ini, pengguna bisa melihat detail informasi visual, sejarah, serta makna filosofis dari setiap corak batik.



Langkah-langkah penggunaan fitur Galeri:

- 1) Pada bar navigasi bagian bawah halaman utama, pilih dan klik ikon **Galeri**. Anda akan diarahkan ke halaman utama fitur yang menampilkan daftar (katalog) berbagai macam jenis motif Batik Tegal yang tersedia di dalam sistem.
- 2) Cari dan pilih salah satu motif batik yang ingin Anda pelajari lebih lanjut dengan cara mengklik pada gambar atau nama motif tersebut.
- 3) Cari dan pilih salah satu motif batik yang ingin Anda pelajari lebih lanjut dengan cara mengklik pada gambar atau nama motif tersebut.
- 4) Sistem akan secara otomatis menampilkan halaman detail yang memuat karakteristik visual, sejarah singkat, serta makna filosofis yang terkandung di dalam motif batik yang Anda pilih.
- 5) Menyukai Motif: Klik tombol Like (ikon hati) untuk menyimpan motif tersebut ke dalam daftar favorit Anda.
- 6) Klik tombol arahkan ke pembatik untuk beralih secara otomatis ke fitur Peta Sebaran guna melihat rute lokasi pengrajin motif tersebut.



7. MEMULAI DETEKSI MOTIF BATIK TEGALAN

Fitur Deteksi merupakan menu utama berbasis *computer vision* untuk mengenali jenis motif Batik Tegalan secara otomatis dan *real-time*.

7.1. Proses Pemindaian Motif (Kamera & Galeri)

Langkah-langkah melakukan pemindaian citra kain batik:



- 1) Pada navigasi bawah, pilih dan klik ikon **Deteksi**.
- 2) **Opsi Buka Kamera:** Klik tombol **Buka Kamera** untuk mengambil foto kain batik secara langsung. Pastikan posisi kamera fokus dan pencahayaan cukup.
- 3) **Opsi Upload Galeri:** Klik tombol **Upload Galeri** jika ingin menggunakan foto motif batik yang sudah tersimpan di memori HP Anda.

- 4) Setelah gambar diambil atau dipilih, sistem akan menampilkan animasi proses pemindaian (*Gambar diterima > Pra-pemrosesan gambar > Mendeteksi motif...*). Mohon tunggu sebentar hingga sistem selesai menganalisis.



7.2. Tampilan Hasil Deteksi

Setelah proses analisis selesai, sistem akan menampilkan halaman hasil pemeriksaan dengan komponen sebagai berikut:

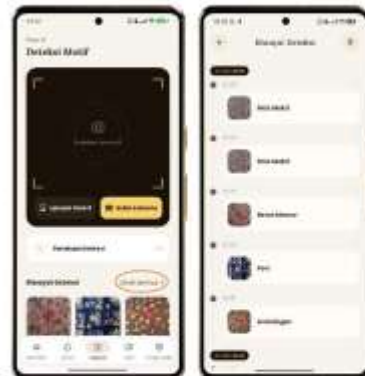


Halaman ini memuat Tingkat Keyakinan (persentase akurasi deteksi sistem), Nama Motif Batik Tegalan yang teridentifikasi (contoh: *Beras Mawur*), serta Makna Motif yang menjelaskan nilai filosofisnya. Selain itu, terdapat tombol Lihat Detail Motif di bagian bawah layar yang dapat diklik untuk beralih langsung ke halaman katalog Galeri guna membaca informasi secara lengkap

7.3. Riwayat Deteksi

Fitur Riwayat Deteksi menyimpan daftar pemindaian motif batik yang telah dilakukan sebelumnya secara urut berdasarkan waktu.

- 1) Pada halaman utama fitur Deteksi, gulir layar ke bawah menuju bagian Riwayat Deteksi.
- 2) Klik Lihat semua untuk membuka seluruh daftar riwayat pemindaian.
- 3) Pilih salah satu kartu riwayat untuk melihat kembali detail persentase tingkat keyakinan serta makna filosofis dari motif kain tersebut.



8. PETA SEBARAN

Fitur Peta Sebaran menyediakan peta interaktif untuk melihat lokasi geografis sentra industri, toko, dan persebaran para pengrajin Batik Tegal.

8.1. Langkah Mengakses Halaman Peta

Berikut adalah langkah untuk membuka peta:



- 1) Pada navigasi bar bagian bawah layar, pilih dan klik ikon **Peta**.
- 2) Sistem akan memuat peta interaktif secara digital yang dipenuhi oleh titik penanda penunjuk lokasi (*marker*) persebaran batik di wilayah Tegal.

8.2. Menggunakan Filter Kategori Tempat

Gunakan kategori filter di bagian atas layar untuk menyaring jenis lokasi yang ingin Anda tampilkan pada peta:

Gunakan kategori filter di bagian atas layar untuk menyaring jenis lokasi yang tampil pada peta. Kategori **Semua** menampilkan seluruh titik lokasi, **Pengrajin** memfokuskan pada rumah produksi lokal, **Toko** menunjukkan gerai penjualan Batik Tegal, **Sentra Batik** memetakan kawasan pusat industri, serta **Edukasi** untuk menampilkan lokasi pembelajaran atau museum terkait Batik Tegal.



8.3. Melihat Detail Informasi Tempat

Untuk mengetahui informasi lengkap mengenai tempat yang dipilih serta rute menuju ke sana, ikuti langkah berikut:

- 1) Klik salah satu titik penanda (*marker*) pada peta, atau gulir panel **Terdekat dari lokasimu** untuk memilih tempat yang dicari.
- 2) Pada halaman detail tempat, pilih tab **Info** pada *top navbar* bagian atas layar untuk melihat profil lengkap, jam operasional, informasi kontak, serta ulasan singkat pengunjung.
- 3) Klik tombol **Arahkan ke sini** untuk membuka rute navigasi peta langsung (*Google Maps*) menuju lokasi tersebut.



8.4. Fitur Memberikan dan Memperbarui Ulasan

Pengguna dapat melihat seluruh penilaian orang lain serta berpartisipasi memberikan ulasan tempat dengan langkah sebagai berikut:



- 1) Pada halaman detail tempat, pilih tab **Ulasan** pada *top navbar* bagian atas layar untuk memuat seluruh daftar ulasan pengunjung secara penuh.
- 2) Gulir layar ke bagian paling bawah, berikan rating bintang sesuai penilaian Anda, dan tulis cerita pengalaman Anda pada kolom komentar yang disediakan, lalu klik **Kirim ulasan**.
- 3) **Memperbarui Ulasan:** Jika Anda sudah pernah memberikan penilaian di tempat tersebut, Anda dapat mengubah isi rating dan komentar kapan saja dengan mengeklik tombol **Perbarui ulasan**.

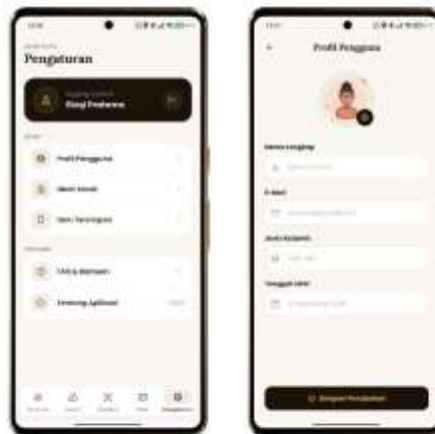
9. PENGATURAN DAN MANAJEMEN AKUN

Menu pengaturan memungkinkan anda untuk mengelola profil, menjaga keamanan akun anda, dan melihat item yang tersimpan seperti motif batik yang anda sukai. Untuk mengakses nya, klik ikon gerigi yang berada di pojok kanan bawah pada menu navigasi

9.1. Mengelola Profil Pengguna

Pada menu pengaturan, pilih “Profil Pengguna” untuk melihat dan mengelola informasi pribadi anda. Di halaman ini, Anda dapat:

- Melihat informasi profil seperti nama lengkap, email, jenis kelamin dan tanggal lahir.
- Mengubah foto profil anda dapat mengklik icons kamerean pada avatar untuk mengubah foto profil.



9.2. Keamanan Akun

Bagian ini dipakai kalau kamu mau mengganti kata sandi biar akunmu lebih aman dan tidak dibajak orang lain.

- Pilih menu ubah kata sandi
- Masukkan sandi lama Anda, kemudian masukan sandi baru dan konfirmasi masukan sandi baru untuk menyelesaikan proses



9.3. Item tersimpan

Masuk ke menu **Item Tersimpan** kalau kamu pengen melihat lagi motif-motif batik yang pernah kamu sukai. Semua corak Batik Tegalan yang pernah kamu klik ikon "Hati" (Like) saat jalan-jalan di menu Galeri akan otomatis tersimpan rapi di sini, jadi kamu tidak perlu repot-repot mencarinya dari awal lagi.



9.4. FAQ & Bantuan



Bagi kamu yang masih bingung cara pakai aplikasi atau tiba-tiba menemukan kendala, langsung saja buka menu **FAQ & Bantuan**. Bagian ini sudah menyediakan daftar pertanyaan yang paling sering ditanyakan oleh pengguna lain lengkap dengan solusi praktisnya, jadi kamu bisa mengatasi masalahmu sendiri dengan cepat.

9.5. Keluar Aplikasi (Logout)

Kalau ingin keluar dari akun, kamu tinggal klik tombol Logout (ikon panah keluar) di sebelah kanan nama profilmu pada bagian paling atas halaman. Setelah diklik, aplikasi akan otomatis menutup akunmu dan kembali ke halaman login awal.



10. QUICK ACTION

10.1. Event

Fitur ini menyajikan info agenda, festival, lomba, dan pameran promosi Batik Tegal.



- 1) Ketuk ikon **Event** pada halaman Beranda untuk masuk ke halaman **Event Batik**.
- 2) Temukan acara yang diinginkan menggunakan kolom pencarian "**Cari event...**" atau ketuk kategori filter yang tersedia (**Semua**, **Lomba**, **Festival**, **Pameran**).

- 3) Klik pada salah satu kartu agenda untuk memuat halaman **Detail Event** guna melihat spanduk visual, informasi waktu, harga tiket, dan deskripsi lengkap acara.



Pada bagian bawah halaman detail, tombol aksi akan muncul secara dinamis sesuai kebutuhan jenis acara. Untuk acara gratis atau tanpa registrasi, Anda dapat langsung mengetuk tombol lebar **Arahkan ke Lokasi** untuk memuat rute peta. Sedangkan untuk acara yang memerlukan pendaftaran, ketuk ikon arah kompas **A** di sebelah kiri untuk navigasi lokasi, atau tekan tombol **Daftar Sekarang** di sebelah kanan untuk memproses registrasi kepesertaan.

10.2. Video

Fitur ini menyajikan koleksi tayangan multimedia edukatif seputar kebudayaan Batik Tegal.



- 1) Ketuk ikon **Video** pada halaman Beranda untuk masuk ke menu **Video Batik**.
- 2) Cari video menggunakan kolom pencarian atau filter kategori yang tersedia (**Semua, Edukasi, Event, Motif & Makna**).
- 3) Klik salah satu kartu video untuk langsung dialihkan dan memutar tayangannya di aplikasi YouTube.

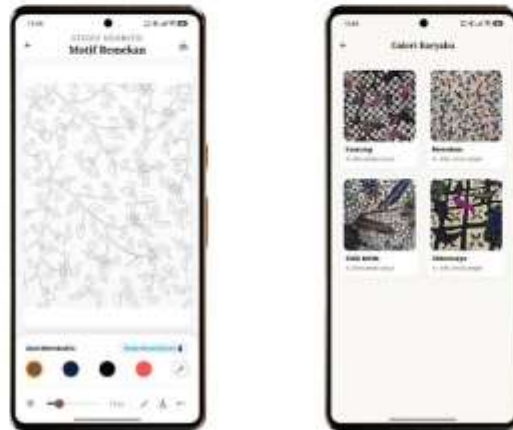
10.3. Batik Art

Fitur ini menyediakan ruang studio kreatif digital untuk menyalurkan warna dan menorehkan motif pada sketsa kain Batik Tegal.



- 1) Ketuk ikon **Batik Art** pada halaman Beranda untuk masuk ke halaman **Studio (Pilih Motif)**.
- 2) Pilih satu ornamen batik favorit Anda dan tekan tombol **Mulai Membatik** untuk membentangkan pola sketsa digital siap warna.
- 3) Sapukan warna pilihan Anda menggunakan baris alat di bawah layar, lalu ketuk ikon unduh di kanan atas untuk menyimpan karya.

- 3) Untuk melanjutkan goresan lama, ketuk ikon palet di kanan atas halaman studio menuju **Galeri Karyaku**, lalu klik "**Klik untuk Lanjut**" pada motif pilihan anda.



10.4. Sejarah



- 1) Ketuk ikon **Sejarah** pada halaman Beranda untuk membuka menu **Sejarah Batik Tegal**.
- 2) Gulir layar ke bawah untuk menelusuri narasi runtutan peristiwa dan rekam jejak perkembangan kebudayaan batik secara utuh.
- 3) Cermati detail infografis di setiap bagian halaman untuk mendalami kisah masa lalu hingga upaya pelestariannya di era digital saat ini.

Lampiran 5. Dokumen Teknis



PENDAHULUAN

1. Latar Belakang
2. Tujuan Pembuatan Dokumen
3. Nilai Inovasi

Aplikasi Sibatikgal menghadirkan inovasi digitalisasi budaya lokal lewat pemanfaatan teknologi *computer vision* (Kecerdasan Buatan) berbasis Android. Sistem ini mampu mengidentifikasi motif Batik Tegal secara otomatis dan *real-time* hanya melalui jepretan kamera, menjadikannya media edukasi budaya yang modern, interaktif, dan mudah dibawa ke mana saja.

4. Dampak Pemanfaatan Aplikasi Sibatikgal

Dengan digunakannya aplikasi ini, masyarakat luas dan wisatawan dapat dengan mudah mengenali serta memahami nilai filosofis di balik setiap motif tradisional Batik Tegal. Dampak luasnya adalah meningkatnya literasi budaya digital, mendukung eksistensi pengrajin batik lokal lewat fitur peta sebaran, serta menjaga warisan budaya Tegal agar tetap lestari di era digital.

5. Spesifikasi Teknis

Spesifikasi Teknis dari Aplikasi Sibatikgal meliputi:

- Manual Pengguna
- Source Code

Berikut uraian spesifikasi untuk pembangunan aplikasi:

- Python (Flask)
- Dart (Flutter)
- Google Colab
- Visual Studio Code
- Web Browser
- MongoDB

SOURCE CODE

1. Authentication Service

Authentication Service merupakan komponen inti yang menangani seluruh siklus manajemen pengguna (User Management), proses pembuktian keaslian identitas (*login*), pendaftaran akun baru (*register*), hingga pengamanan jalur akses API (*route protection*) menggunakan skema JWT (*JSON Web Token*) Bearer pada aplikasi Sibatikgal.

a. Komponen user_model.py

File ini mendefinisikan skema data dokumen pengguna di dalam database NoSQL (MongoDB). Karena menggunakan MongoDB melalui Flask-PyMongo, skema tidak menggunakan kelas ORM kaku, melainkan dikelola langsung lewat fungsi pembentuk struktur dokumen (*dictionary structure*).

```
def create_user(data):
    bcrypt = current_app.bcrypt
    hashed_password =
    bcrypt.generate_password_hash(data['password']).decode('utf-8')

    user = {
        'username': data['username'],
        'email': data['email'],
        'password': hashed_password,
        'role': data.get('role', 'user'),
        'api_key': data['api_key'],
        'is_verified': data.get('is_verified', False),
        'otp': data.get('otp', None), # Skip OTP = digit
        'otp_expiry': data.get('otp_expiry', None), # Skip otp
        'health_provider': data.get('health_provider', None),
        'profile_picture': data.get('profile_picture', ''),
        'gender': data.get('gender', ''),
        'tanggal_lahir': data.get('tanggal_lahir', ''),
        'created_at': datetime.utcnow()
    }

    current_app.mongo.db.users.insert_one(user)
    return user
```

Fungsi ini bertindak sebagai *entry point* untuk menyusun skema sekaligus menyimpan data dokumen pengguna baru ke dalam *collection* MongoDB. Menerima input data mentah dari form registrasi, mengamankan kredensial, lalu menerapkan nilai bawaan (*default*) untuk field opsional.

Alur Kerja Fungsi:

- **Ekstraksi Encryptor:** Fungsi mengambil *instance* berypt dari konteks aplikasi Flask yang sedang berjalan (*current_app*) untuk kebutuhan pengamanan sandi.
- **Hashing Password:** String kata sandi murni dari *data['password']* diacak menggunakan algoritma *cryptographic hashing* satu arah melalui Bcrypt dan didekode ke format utf-8 agar aman disimpan di database.
- **Inisialisasi Hak Akses:** Menentukan *role* pengguna dengan *fallback value* "users" jika tidak ada parameter khusus yang dikirimkan.
- **Penyusunan Kredensial & OTP:** Menyematkan string *api_key* unik, status *is_verified* (default False), serta field penampung *otp* dan *otp_expiry* untuk alur verifikasi akun.
- **Database Insertion:** Objek dokumen user yang sudah rapi disisipkan langsung ke database MongoDB pada *collection* users menggunakan perintah bawaan driver PyMongo (*insert_one(user)*).

b. Endpoint register(methods=['POST'])

Fungsi ini bertindak sebagai *entry point* bagi pengguna baru untuk mendaftarkan akun ke dalam sistem Sibatikgal. Fungsi ini menerima input data berformat JSON (seperti nama, email, dan password), memvalidasi ketersediaan email di database MongoDB, menghasilkan kode OTP dengan masa kedaluwarsa, lalu mengirimkannya ke email pengguna.



```

def register(request):
    """
    Endpoint untuk mendaftarkan pengguna baru.
    """
    data = request.get_json()
    email = data.get('email')

    # Validasi email (sudah ada atau belum)
    if not validate_email(email):
        return jsonify({'msg': 'Email sudah terdaftar'}), 400

    # Generate OTP
    otp_expiry = datetime.utcnow() + timedelta(minutes=5)

    # Generate API Key
    api_key = generate_api_key()

    # Hash Password
    hashed_password = bcrypt.hashpw(
        data.get('password').encode('utf-8'),
        bcrypt.gensalt()
    ).decode('utf-8')

    # Simpan ke database
    user = {
        'email': email,
        'password': hashed_password,
        'api_key': api_key,
        'otp': otp,
        'otp_expiry': otp_expiry,
        'is_verified': False
    }

    # Simpan ke database
    db.users.insert_one(user)

    # Kirim OTP ke email
    send_email(email, otp)

    return jsonify({'msg': 'Registrasi berhasil. Silakan cek OTP di email'}), 201

```

c. Derivations: Dates: Five

- endpoint `verify_otp(methods=['POST'])` telah melakukan registrasi, akun pengguna berstatus tidak aktif (`_verified: False`) hingga mereka memasukkan kode OTP yang valid. Fungsi ini bertugas memvalidasi kesesuaian kode OTP dan memastikan bahwa masa berlaku kode tersebut belum melewati batas kedaluwarsa.

Setelah melakukan registrasi, akun pengguna berstatus tidak aktif (`is_verified: False`) hingga mereka memasukkan kode OTP yang valid. Fungsi ini bertugas memvalidasi kesesuaian kode OTP dan memastikan bahwa masa berlaku kode tersebut belum melewati batas kedaluwarsa.

Penjelasan Alur Logika Program:

- **Pengecekan Masa Aktif:** Sistem membandingkan waktu saat ini (`datetime.utcnow()`) dengan `otp_expiry` yang tersimpan di database. Jika waktu sekarang lebih besar, sistem menolak verifikasi dengan status 400 Bad Request.
- **Pencocokan Kode:** Jika kode `otp_input` dari pengguna cocok dengan `user['otp']`, status akun diubah menjadi `is_verified: True`. Kolom `otp` dan `otp_expiry` dikosongkan (`None`) kembali demi keamanan, lalu mengembalikan status 200 OK.

d. Endpoint login(methods=['POST'])

Fungsi login digunakan untuk mengautentikasi pengguna yang ingin masuk ke dalam aplikasi Sibatikgal. Sistem akan memeriksa apakah email terdaftar, apakah akun sudah lolos verifikasi OTP, dan verifikasi kecocokan *password* yang di-hash.

```
def login(request):  
    data = request.get_json()  
    email = data.get('email')  
    password = data.get('password')  
    user = User.objects(email=email)  
    if not user:  
        return resp.status(400, message='User not found', password=1,  
                           password=0)  
    if not user.get('is_verified', False):  
        return resp.status(400, message='Silahkan verifikasi akun  
            anda dulu email', 0)  
    # Cek password  
    password_hash = generate_password_hash(password, 128)  
    if not user.check_password(password_hash):  
        return resp.status(400, message='Password salah', 0)  
    # Login berhasil  
    token = generate_jwt_token({  
        'email': email,  
        'password': password,  
        'is_verified': user.get('is_verified', False)  
    })  
    return resp.status(200, message='Login berhasil', token=token)
```

Penjelasan Alur Logika Program:

- **Penerimaan Input:** Menerima parameter email dan password dari form login dalam format JSON.Keamanan Kredensial: Fungsi `check_password_hash()` membandingkan *password* teks murni inputan user dengan *password* yang terenkripsi di MongoDB.

- **Pencarian & Validasi Kredensial:** Sistem mencari data pengguna menggunakan `find_user_by_email()`. Jika user tidak ditemukan atau *password* yang di-hash via `bcrypt.check_password_hash()` tidak cocok, sistem mengembalikan pesan error "Email atau password salah" dengan status
- **Validasi Verifikasi Akun:** Jika kredensial benar, sistem memeriksa status akun melalui field `is_verified`. Jika bernilai `False` (belum verifikasi OTP), sistem menolak login dengan status 403 dan menampilkan pesan "Silakan verifikasi akun kamu via email"
- **Pembuatan Token JWT:** Jika semua validasi lolos, sistem menerbitkan `access_token` dan `refresh_token` menggunakan ID unik user (`_id`) sebagai identitasnya.
- **Pengembalian Respons:** Mengembalikan HTTP status 200 beserta kedua token JWT dan data profil dasar user (email dan username).

e. Endpoint `forgot_password()`

Fungsi ini digunakan ketika pengguna lupa kata sandi dan meminta kode OTP baru untuk melakukan reset password.

```
def forgot_password(request):
    """Endpoint for forgot password"""
    data = request.get_json()
    email = data.get('email')

    user = find_user_by_email(email)
    if not user:
        return jsonify({'msg': 'Email tidak ditemukan'}), 404

    otp = generate_otp()
    otp_expiry = datetime.utcnow() + timedelta(minutes=5)

    # Save user state and otp
    update_user_state(email, otp_expiry)
    send_reset_password_email(email, otp)

    return jsonify({'msg': 'Email OTP telah dikirim ke email Anda'}), 200
except Exception as e:
    return jsonify({'msg': 'Gagal mengirim email',
                    'error': str(e)}), 500
```

- **Penerimaan Input:** Menerima parameter email dalam format JSON melalui *request*.

- **Pengecekan Pengguna:** Sistem mencari data akun menggunakan `find_user_by_email()`. Jika tidak terdaftar, mengembalikan status 404 dengan pesan "User tidak ditemukan".
- **Pembuatan & Pengiriman OTP:** Jika akun ada, sistem membuat OTP baru beserta masa kedaluwarsanya (5 menit), memperbarui data tersebut di database lewat `update_user_otp()`, dan mengirimkan email instruksi via `send_reset_password_email()`.
- **Respons:** Mengembalikan HTTP status 200 dengan pesan "OTP untuk reset password telah dikirim ke email kamu".

f. Endpoint `verify_reset_otp()`

```

1 @auth_bp.route('/verify-reset-otp', methods=['POST'])
2 def verify_reset_otp():
3     data = request.get_json()
4     email = data.get('email')
5     otp_input = data.get('otp')
6
7     user = find_user_by_email(email)
8     if not user:
9         return jsonify({"msg": "User tidak ditemukan"}), 404
10
11     if datetime.utcnow() > user.get('otp_expiry'):
12         return jsonify({"msg": "OTP sudah kadaluarsa"}), 400
13
14     if user.get('ota') != otp_input:
15         return jsonify({"msg": "OTP salah"}), 400
16
17     return jsonify({"msg": "OTP Valid, silakan masukkan kata sandi baru"}), 200

```

- Fungsi ini bertugas memvalidasi kode OTP yang dimasukkan pengguna sebelum mereka diizinkan mengisi kata sandi baru.
- **Penerimaan Input:** Menerima parameter email dan otp (input pengguna) dalam format JSON.
- **Validasi OTP & Masa Aktif:** Sistem mencocokkan otp input dengan yang ada di database, sekaligus memastikan waktu saat ini belum melewati `otp_expiry`.
- **Pengembalian Respons:** * Jika valid, mengembalikan status 200 dengan pesan "OTP Valid, silakan masukkan kata sandi baru".
 - Jika salah atau kadaluarsa, mengembalikan status 400 beserta pesan *error* terkait.

g. Endpoint `reset_password()`

Fungsi akhir yang memproses pembaruan kata sandi pengguna ke database setelah semua validasi terpenuhi.

```
from flask import Flask, request, jsonify
from flask_jwt import JWT
from flask_jwt_extended import JWTManager, jwt_required, create_jwt_token
from flask_cors import CORS
from datetime import datetime
from werkzeug.security import generate_password_hash, check_password_hash
from flask import g

app = Flask(__name__)
CORS(app)
JWTManager(app)

@app.route('/reset-password', methods=['POST'])
def reset_password():
    data = request.get_json()
    email = data.get('email')
    otp_input = data.get('otp')
    new_password = data.get('password')

    user = find_user_by_email(email)
    if not user:
        return jsonify({"msg": "User tidak ditemukan"}), 404

    if datetime.utcnow() > user.get('otp_expiry') or user.get('otp') != otp_input:
        return jsonify({"msg": "Verifikasi OTP gagal atau kadaluarsa"}), 401

    hashed_password = generate_password_hash(new_password, method='pbkdf2')
    current_app.bcrypt.generate_password_hash(new_password, method='pbkdf2')
    update_user_password(email, hashed_password)

    return jsonify({"msg": "Kata sandi berhasil diperbarui, silakan login kembali!"}), 200
```

- **Penerimaan Input:** Menerima parameter email, otp (untuk verifikasi ulang keamanan di sisi server), dan new_password.
- **Verifikasi Akhir & Enkripsi:** Sistem memastikan kembali kevalidan OTP. Jika sah, string new_password akan di-hash menggunakan `current_app.bcrypt.generate_password_hash()` demi keamanan data kredensial.
- **Pembaruan Data:** Kata sandi baru yang telah terenkripsi disimpan ke database via `update_user_password()`, lalu kolom OTP dibersihkan (None).
- **Respons:** Mengembalikan HTTP status 200 dengan pesan "Kata sandi berhasil diperbarui, silakan login kembali".

2. Galeri Service

Layanan ini bertanggung jawab untuk mengelola, menampilkan, serta mencari data dokumentasi motif Batik Tegal di dalam sistem **Sibatikgal**. Melalui layanan ini, aplikasi *client* dapat menyajikan informasi edukasi budaya berupa

daftar galeri motif tradisional, fitur pencarian, sistem pembagian halaman (*pagination*), hingga detail filosofi dari setiap motif batik secara terintegrasi.

a. Endpoint: `get_all_batik()`

```

def get_all_batik():
    """
    Endpoint: get_all_batik()
    """
    # Inisialisasi parameter
    page = request.args.get('page', 1, type=int)
    search_query = request.args.get('q', '')
    per_page = 20

    # Query Database
    all_data = batik_model.get_all(search_query)

    # Hitung total item dan total halaman
    total_items = len(all_data)
    total_pages = math.ceil(total_items / per_page)

    # Hitung indeks halaman
    start = (page - 1) * per_page
    end = start + per_page
    data_to_get = all_data[start:end]

    # Format data
    formatted_data = []
    for d in data_to_get:
        d['_id'] = str(d['_id'])
        d['image_url'] = f"/static/images/{d['image_filename']}"
        d['detail_url'] = f"/detail/{d['_id']}"
        formatted_data.append(d)

    # Response
    return jsonify({
        "status": "success",
        "message": "Data galeri batik berhasil diambil",
        "data": formatted_data,
        "pagination": {
            "current_page": page,
            "total_page": total_pages,
            "total_item": total_items
        }
    })

```

Penjelasan Alur Logika Program:

- **Inisialisasi Parameter:** Sistem membaca parameter query eksternal dari URL berupa `page` (halaman aktif, default 1) dan `q` (kata kunci pencarian motif batik). Jumlah item per halaman dibatasi sebanyak 20 data (`per_page = 20`).
- **Query Database:** Memanggil fungsi `batik_model.get_all(search_query)` untuk menarik data motif batik dari MongoDB yang sesuai dengan kata kunci pencarian.
- **Proses Paginasi (*Pagination Law*):** Sistem menghitung total item dan total halaman menggunakan `math.ceil()`, lalu memotong baris array data (`all_data[start:end]`) sesuai dengan indeks halaman yang diminta agar beban muat data pada aplikasi *client* tetap ringan.

- **Transformasi & Pemformatan Data:** * Mengonversi tipe data `_id` bawaan MongoDB (ObjectID) menjadi bentuk *string* agar bisa ditransmisikan via JSON.
 - Membentuk URL absolut gambar motif batik (`image_url`) dengan menggabungkan host URL sistem ke direktori statis gambar galeri.
 - Mengubah format waktu `created_at` menjadi format string tanggal standar (YYYY-MM-DD HH:MM:SS).
- **Respons Sistem:** Mengembalikan HTTP Status 200 beserta array data batik yang sudah diformat dan objek metadata paginasi untuk kebutuhan komponen *scrolling* di Flutter.

b. Endpoint: `get_batik_detail(batik_id)`

```

1 @gallery_api.route('/gallery/estring:batik_id', methods=['GET'])
2 def get_batik_detail(batik_id):
3     try:
4         batik = batik_model.get_by_id(batik_id)
5         if not batik:
6             return jsonify({'status': 'error', 'message': 'Data batik tidak
7             ditemukan'}), 404
8
9         batik['_id'] = str(batik['_id'])
10        batik['image_url'] = ""
11        [request.host_url, static/img/gallery/batik.get('image_url',
12        'default_batik.png')]
13
14        if 'created_at' in batik and batik['created_at']:
15            batik['created_at'] = batik['created_at'].strftime('%Y-%m-%d
16            %H:%M:%S')
17
18        return jsonify({
19            "status": "success",
20            "data": batik
21        }), 200
22    except Exception as e:
23        return jsonify({'status': 'error', 'message': str(e)}), 500

```

Penjelasan Alur Logika Program:

- **Pencarian Entitas Spesifik:** Sistem menerima parameter `batik_id` dari segmen URL, kemudian melakukan pencarian dokumen Tunggal ke MongoDB melalui fungsi `batik_model.get_by_id(batik_id)`.

- **Validasi Keberadaan Data:** Jika fungsi mengembalikan nilai kosong (*None*), sistem langsung menghentikan proses dan mengirimkan respons *error* "Data batik tidak ditemukan" dengan HTTP Status 404 Not Found.
- **Standardisasi Format Payload:** Jika data ditemukan, sistem melakukan penyesuaian format payload yang sama dengan layanan galeri utama, yaitu konversi *_id* ke *string*, konstruksi URL gambar statis secara lengkap, dan pemformatan tanggal *created_at*.
- **Respons Sistem:** Mengembalikan data detail lengkap dari motif Batik Tegalan yang dipilih (seperti nama motif, deskripsi, filosofi, dan gambar) dengan HTTP Status 200 OK.

3. Deteksi Service

Layanan deteksi merupakan komponen kecerdasan buatan utama pada sistem **Sibatikgal** yang berfungsi untuk mengidentifikasi citra motif Batik Tegalan secara otomatis. Berbeda dengan modul statis lainnya, layanan ini menjamin efisiensi komputasi server lewat pengikatan modul bobot (*weight model*) arsitektur Xception langsung pada RAM server saat *startup* awal, sehingga mampu memotong waktu tunggu pemrosesan gambar secara signifikan. Modul ini menerima unggahan gambar fisik dari aplikasi mobile Android, melakukan normalisasi matriks piksel, mengeksekusi prediksi, hingga memetakan filosofi makna kebudayaan lokal secara *real-time* untuk kemudian disimpan ke dalam database MongoDB.

a. Endpoint: predict()

```

def predict():
    jwt_token = request.headers.get('JWT')
    if not jwt_token:
        return jsonify({'message': 'JWT token is required'})

    user_id = get_jwt_identity()
    safe_filename = secure_filename(request.files['image'].filename)
    new_filename = f'history_{user_id}_{datetime.now().strftime('%Y%m%d_%H%M%S')}.jpg'
    save_path = os.path.join(safe_filename, new_filename)
    permanent_path = os.path.join('static/img/history', new_filename)
    request.files['image'].save(os.path.join('static/img/history', new_filename))

    img = image.load_img(permanent_path, target_size=(224, 224))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0) / 255.0

    preds = model.predict(x)
    result_label = CATEGORIES[np.argmax(preds)]
    confidence = float(preds[0][np.argmax(preds)] / 100)

    detail_data = batik_db.get_by_nama(result_label)
    batik_output = detail_data['nama'] if detail_data else "Bukan Batik Tegalan"
    history_db.create_history(user_id, result_label, f'{confidence:.2f}%', batik_output,
                           True, f'{permanent_path}')

    return jsonify({'nama': result_label, 'confidence': f'{confidence:.2f}%', 'pesan':
                    batik_output}), 200

```

- **Autentikasi & Amankan File:** Sistem memvalidasi token JWT pengguna, menyaring nama berkas dengan secure filename untuk mencegah celah *path traversal*, lalu menyimpannya ke direktori lokal menggunakan nama unik berbasis gabungan *user_id* dan *timestamp*.
- **Prapemrosesan Citra:** Berkas gambar dimuat dan diubah ukurannya secara paksa menjadi dimensi 224 x 224 piksel dengan format matriks warna RGB, lalu diperluas dimensinya serta dibagi terhadap nilai 255.0 untuk proses normalisasi tensor.
- **Klasifikasi Model AI:** Matriks tensor diproses oleh fungsi `model.predict(x)`, di mana indeks dengan probabilitas tertinggi diekstrak menggunakan fungsi `np.argmax()` untuk menentukan label nama motif batik beserta persentase akurasi (*confidence score*).
- **Kueri Database & Respons:** Sistem mencari penjelasan filosofis motif yang cocok melalui kueri ke MongoDB via `batik_db.get_by_nama()`, merekam hasil penafsiran lengkap ke dalam koleksi riwayat `history_db`, lalu mengembalikan respons JSON sukses berstatus HTTP 200 ke aplikasi Flutter.

b. Endpoint: `get_history()`

```
1 @detail_bp.route('/history', methods=['GET'])
2 @jwt_required()
3 def get_history():
4     try:
5         user_id = get_jwt_identity()
6         user_history = history_db.get_by_user(user_id)
7         return jsonify({"status": "success", "data": user_history}), 200
8     except Exception as e:
9         return jsonify({"status": "error", "message": str(e)}), 500
```

Penjelasan Alur Logika Program:

- **Ekstraksi Token:** Sistem mengekstrak parameter `user_id` klien yang sedang aktif dari enkripsi payload akses token JWT yang dikirimkan.
- **Pengambilan Data:** Backend melakukan kueri ke MongoDB menggunakan fungsi `history_db.get_by_user(user_id)` untuk menarik seluruh data dokumen riwayat milik pengguna tersebut secara terisolasi.
- **Pengiriman Payload:** Seluruh data array dokumen riwayat yang ditemukan langsung dibungkus ke dalam objek respons JSON sukses untuk dikirimkan kembali menuju aplikasi Flutter.

4. Mapping Service

Layanan ini bertanggung jawab untuk mengelola data pemetaan lokasi perajin dan toko Batik Tegal (fitur peta sebaran) guna mendukung eksistensi pengrajin lokal. Selain menyajikan koordinat dan informasi tempat, layanan ini memproses interaksi ulasan (*review*) dari pengguna lengkap dengan sistem proteksi anti-spam di database MongoDB.

a. Endpoint: `get_all_locations()`

Sistem menerima parameter kata kunci pencarian `q` via URL untuk menyaring nama toko atau wilayah perajin Batik Tegal. Database MongoDB kemudian ditarik menggunakan fungsi `mapping_model.get_all()`, lalu seluruh hasilnya dilewatkan ke fungsi pembantu `serialize_doc()` untuk mengubah tipe data bawaan BSON (seperti ObjectId dan datetime) menjadi format teks JSON yang aman. Data yang

telah bersih tersebut dikirimkan kembali ke aplikasi Flutter dengan kode status HTTP 200 untuk dipetakan ke dalam Google Maps.

```

1 @mapping_model.route('/mappings', methods=['GET'])
2 def get_all_locations():
3     try:
4         search_query = request.args.get('g', '')
5         all_data = mapping_model.get_all(search_query)
6
7         # Success response to the client
8         serialized_data = [serialize_obj(obj) for obj in all_data]
9
10        return jsonify({
11            "status": "success",
12            "message": "Berhasil mengambil data lokasi botih",
13            "data": serialized_data
14        }), 200
15    except Exception as e:
16        return jsonify({"status": "error", "message": str(e)}, 500)

```

b. Endpoint: `get_detail_location(mapping_id)`

Ketika pengguna memilih salah satu penanda (*marker*) peta di aplikasi mobile, parameter `mapping_id` dikirimkan ke backend. Sistem langsung memvalidasi struktur string tersebut menggunakan `ObjectId.is_valid()`, di mana jika formatnya rusak atau tidak sesuai standar MongoDB, proses dihentikan dengan status HTTP 400. Jika formatnya valid, kueri pencarian tunggal dijalankan; apabila toko ditemukan, datanya diserialisasi dan dikirimkan dengan status HTTP 200 untuk menampilkan profil lengkap toko beserta koordinatnya.

```

1 @mapping_model.route('/mappings/<mapping_id>', methods=['GET'])
2 def get_detail_location(mapping_id):
3     try:
4         # Validasi format ObjectId
5         if not ObjectId.is_valid(mapping_id):
6             return jsonify({"status": "error", "message": "ID lokasi tidak valid"}),
7             400
8
9         mapping = mapping_model.get_by_id(mapping_id)
10        if not mapping:
11            return jsonify({"status": "error", "message": "Data lokasi tidak
12            ditemukan"}), 404
13
14        return jsonify({
15            "status": "success",
16            "message": "Berhasil mengambil detail lokasi botih",
17            "data": serialize_obj(mapping)
18        }), 200
19    except Exception as e:
20        return jsonify({"status": "error", "message": str(e)}, 500)

```

c. Endpoint: `post_review(mapping_id)`

Endpoint ini membutuhkan verifikasi token JWT untuk mencatat ulasan baru dari pengguna aplikasi. Setelah input angka rating divalidasi harus berada pada rentang 1 sampai 5, backend menerapkan proteksi anti-spam ketat dengan memanggil fungsi `check_existing_review()`. Jika database mendeteksi ID pengguna tersebut sudah pernah mengulas toko yang sama, permintaan langsung ditolak dengan status HTTP 400. Sebaliknya, jika belum pernah, sistem merangkai struktur sub-dokumen review baru berisi data akun beserta waktu ulasan, lalu mendorongnya masuk ke dalam array ulasan MongoDB melalui perintah `mapping_model.add_review()`.

```
from flask import Flask, jsonify, request, abort, g
from flask_jwt import JWT
from flask_jwt_extended import jwt_required, get_jwt_identity
from flask_cors import CORS
from datetime import datetime
from bson import ObjectId
from pymongo import MongoClient
from flask import jsonify

app = Flask(__name__)
CORS(app)

def jwt_authenticate(username, password):
    # Authentication logic here
    return True

jwt = JWT(app, jwt_authenticate, 'SECRET_KEY')

@app.route('/api/review', methods=['POST'])
@jwt_required()
def post_review(mapping_id):
    # Validation logic here
    # Check if user has already reviewed this shop
    # Add new review if not
    # Return response
    return jsonify({'status': 'success', 'message': 'Ulasan berhasil ditambahkan'})
```

d. Endpoint: `update_review(mapping_id)`

Sebagai alternatif dari penolakan ganda ulasan baru, fungsi ini memfasilitasi pengguna untuk memperbarui ulasan lama yang telah tersimpan. Sistem memanfaatkan data `current_user_id` dari token JWT untuk melacak letak ulasan spesifik miliknya di dalam array dokumen lokasi. Fungsi `mapping_model.update_user_review()` bekerja memperbarui nilai rating dan deskripsi komentar secara instan; jika status pembaruan bernilai gagal karena pengguna memang belum pernah mengulas tempat tersebut, sistem mengirimkan kode status HTTP 400, dan mengirimkan HTTP 200 jika data berhasil diperbarui.

```
1 mapping_api.route('/mappings/<string:mapping_id>/reviews', methods=['PUT'])
2 @jwt.requires_auth
3 def update_review(mapping_id):
4     try:
5         if not ObjectId.is_valid(mapping_id):
6             return jsonify({"status": "error", "message": "ID Lokasi tidak valid"}), 400
7
8         data = request.get_json()
9         rating = data.get('rating')
10        comment = data.get('comment', '')
11
12        if not rating or not (1 <= int(rating) <= 5):
13            return jsonify({"status": "error", "message": "Rating wajib diisi angka 1-5"}), 400
14
15        current_user_id = get_jwt_identity()
16
17        # Lakukan update ke sub-dokumen array MongoDB
18        success = mapping_model.update_user_review(mapping_id, current_user_id,
19            rating, comment)
20
21        if not success:
22            return jsonify({
23                "status": "error",
24                "message": "Gagal memperbarui ulasan. Kamu belum pernah mengulas tempat ini atau lokasi tidak ditemukan."
25            }), 400
26
27        return jsonify({"status": "success", "message": "Ulasan berhasil diperbarui"}), 200
28
29    except Exception as e:
30        return jsonify({"status": "error", "message": str(e)}), 500
```

5. Informasi Service

Layanan ini bertanggung jawab untuk mengelola dan mendistribusikan artikel, berita, atau konten edukatif seputar perkembangan Batik Tegalana kepada pengguna. Selain mendukung pencarian dan pembagian halaman (*pagination*), layanan ini mengintegrasikan data relasional untuk menampilkan identitas admin selaku penulis artikel melalui mekanisme penggabungan data database.

a. Endpoint: `get_all_informasi()`

[illegible]

Sistem menerima parameter halaman aktif dan kata kunci query pencarian via URL, lalu mengambil seluruh dokumen artikel yang relevan dari MongoDB melalui fungsi `info_model.get_all()`. Untuk meringankan beban transfer data pada aplikasi Flutter, baris array data dipotong berdasarkan batas maksimal 50 item per halaman (`per page = 50`). Dalam proses

perulangan, objek mentah hasil agregasi \$lookup database (admin_data) diekstrak untuk memetakan nama akun admin ke field author_name sebelum dihapus dari memori respons, diikuti dengan konversi data ID, standarisasi ISO untuk tanggal pembuatan, serta pembentukan tautan URL absolut gambar artikel secara lengkap.

b. Endpoint: `get_categories_api()`

```
1 @informasi_api_bp.route('/informasi/categories', methods=['GET'])
2 def get_categories_api():
3     try:
4         categories = info_model.get_distinct_categories()
5         # Bersihkan data jika ada string kosong atau null
6         cleaned_categories = [cat for cat in categories if cat]
7
8         return jsonify({
9             "status": "success",
10            "data": cleaned_categories
11        }), 200
12    except Exception as e:
13        return jsonify({"status": "error", "message": str(e)}), 500
```

Endpoint ini bertugas untuk menyajikan daftar kategori unik dari seluruh artikel edukasi yang tersedia guna mempermudah fitur penyaringan (*filtering*) di antarmuka klien. Backend memanggil fungsi model `info_model.get_distinct_categories()` yang mengeksekusi kueri pencarian nilai unik langsung pada database MongoDB. Sebelum data dikembalikan, sistem menerapkan operasi pembersihan array (*list comprehension*) untuk membuang entri data kosong atau bernilai kosong (*null*) agar parameter respons JSON yang dikirimkan tetap bersih dan valid dengan kode status HTTP 200.

c. Endpoint: `get_detail_informasi(info_id)`

Fungsi ini berjalan untuk memuat konten lengkap dari satu artikel pilihan pengguna berdasarkan parameter `info_id` yang disematkan pada segmen URL. Jika kueri database mengembalikan nilai kosong, proses seketika dihentikan dengan memuntahkan respons *error* berstatus HTTP 404. Sebaliknya, apabila dokumen ditemukan, sistem melakukan penyesuaian

data yang serupa dengan fungsi galeri utama, mencakup transformasi string ID, pemformatan zona waktu, konstruksi path berkas gambar statis, serta ekstraksi nama admin pembuat konten dari sub-dokumen agregasi demi menyajikan kredibilitas penulis pada halaman detail aplikasi Flutter.

6. Studi Service

a. Endpoint: `get_canvas_sketches()`

is_sketsa_available bernilai benar serta memiliki tautan gambar sketsa. Dalam proses iterasi, string URL sketsa diubah jalurnya secara dinamis mengarah langsung ke endpoint API internal berbasis ID unik batik guna memperketat manajemen berkas, lalu dikemas dalam respons JSON sukses berstatus HTTP 200 untuk merendering daftar opsi sketsa kanvas di Flutter.

```
@staticmethod
def get_sketch_image_by_id(batik_id):
    """
    Get sketch image by batik id
    """
    all_batik = batik_model.get_all(include_deleted=False)
    canvas_ready_batik = []
    for b in all_batik:
        if b.get('is_sketch_available') == True and b.get('sketch_image_url'):
            b['id'] = batik_id
            b['image_url'] = ''
            request_host_url = static_url + f'/b/{b.get('image_url')}/sketch_image.png/'
            request_host_url = request_host_url + f'/b/{b.get('id')}/'
            if 'created_at' in b and b['created_at']:
                b['created_at'] = b['created_at'].strftime('%Y-%m-%d %H:%M:%S')
            canvas_ready_batik.append(b)

    return jsonify({
        'status': 'success',
        'message': 'Data sketsa studio sukses berhasil dimuat',
        'total_canvas': len(canvas_ready_batik),
        'data': canvas_ready_batik
    })

except Exception as e:
    return jsonify({'status': 'error', 'message': str(e)}), 500
```

b. Endpoint: `get_sketch_image_by_id(batik_id)`

```
@staticmethod
def get_sketch_image_by_id(batik_id):
    """
    Get sketch image by batik id
    """
    batik = batik_model.get_by_id(batik_id)
    if not batik or not batik.get('sketch_image_url'):
        return jsonify({'status': 'error', 'message': 'File sketsa tidak ditemukan di database'}), 404

    sketch_filename = batik['sketch_image_url']
    if not sketch_filename.endswith('.png'):
        sketch_filename = f'{sketch_filename}.png'

    SKETCH_DIRECTORY = os.path.join('static', 'img', 'sketch')
    if not os.path.exists(os.path.join(SKETCH_DIRECTORY, sketch_filename)):
        return jsonify({'status': 'error', 'message': 'File tidak ada di direktori server'}), 404

    return send_from_directory(SKETCH_DIRECTORY, sketch_filename)

except Exception as e:
    return jsonify({'status': 'error', 'message': str(e)}), 500
```

Endpoint ini bertugas menyajikan berkas fisik citra sketsa batik hitam-putih transparan kepada aplikasi klien untuk dijadikan sebagai latar belakang (*background layer*) kanvas mewarnai. Backend memburu dokumen batik

terkait berdasarkan parameter batik_id di URL, melakukan standarisasi ekstensi penamaan berkas ke format .png, serta memverifikasi keberadaan fisik file di dalam direktori internal server. Jika pemeriksaan fisik file terbukti sah, server langsung mengalirkan berkas biner gambar mentah tersebut menggunakan fungsi bawaan Flask send_from_directory().

c. Endpoint: save_draft()

```

1 # Batik API RESTful /studio/save-draft', methods=['POST'])
2 @jwt_required()
3 def save_draft():
4     try:
5         current_user_id = get_jwt_identity()
6         data = request.get_json()
7         batik_id = data.get('batik_id')
8         canvas_id = data.get('canvas_id')
9
10        if not batik_id or not canvas_id:
11            return jsonify({'status': 'error', 'message': 'Parameter batik_id atau
12            canvas_id tidak boleh kosong'}), 400
13
14        success = draft_model.save_or_update_draft(current_user_id, batik_id,
15            canvas_id)
16        if success:
17            return jsonify({'status': 'success', 'message': 'Progres mendefinisikan
18            teruskan'}), 200
19        else:
20            return jsonify({'status': 'error', 'message': 'Gagal menyimpan draf ke
21            database'}), 500
22    except Exception as e:
23        return jsonify({'status': 'error', 'message': str(e)}), 500

```

Fungsi ini dipicu secara berkala atau ketika pengguna menekan tombol simpan progres mewarnai pada aplikasi mobile. Sistem membongkar token akses JWT klien guna mengekstrak parameter current_user_id lalu menangkap parameter payload data JSON koordinat garis atau pewarnaan kanvas dari tubuh request. Paket data terisolasi tersebut dikirim menuju fungsi pintar draft_model.save_or_update_draft(), yang secara otomatis akan membuat dokumen draf baru jika belum pernah ada, atau menimpa progres draf lama di MongoDB jika user mengedit motif batik yang sama.

d. Endpoint: get_draft()

Ketika halaman studio kanvas memuat suatu motif batik, endpoint terproteksi JWT ini dipanggil untuk memulihkan progres mewarnai terakhir milik pengguna bersangkutan. Backend memanggil fungsi draft_model.get_draft() dengan mengirimkan relasi data berupa kombinasi

ID user dan ID batik. Jika database mendeteksi bahwa pengguna baru pertama kali membuka canvas motif tersebut (belum ada draf), sistem secara cerdas mengembalikan nilai string kosong "" agar canvas Flutter bersih, dan mengirimkan objek isi matriks koordinat canvas_json jika rekam jejak draf lama ditemukan.

```

@staticmethod.route('/studio/get-draft', methods=['GET'])
def get_draft():
    try:
        current_user_id = get_jwt_identity()
        batik_id = request.args.get('batik_id')

        if not batik_id:
            return jsonify({"status": "error", "message": "Parameter batik_id wajib disertakan"}), 400

        draft = draft_model.get_draft(current_user_id, batik_id)
        if not draft:
            return jsonify({"status": "success", "canvas_json": ""}), 200

        return jsonify({
            "status": "success",
            "canvas_json": draft['canvas_json']
        }), 200
    except Exception as e:
        return jsonify({"status": "error", "message": str(e)}), 400

```

e. dpoint: get_my_drafts()

```

@staticmethod.route('/studio/my-drafts', methods=['GET'])
def get_my_drafts():
    try:
        # Ambil data identitas (user) identitas
        current_user_id = get_jwt_identity()

        my_my_drafts = SQLAlchemy.db.session.query(Draft).filter(
            Draft.user_id == current_user_id).all()

        # Konversi draft_list = []

        for draft in my_my_drafts:
            batik_tema = batik_model.get_by_id(draft.batik_id)
            if batik_tema:
                # Format data ke json
                data = {
                    "id": draft.id,
                    "nama": batik_tema.get('nama'),
                    "image_url": "",
                    "thumbnail_url": batik_tema.get('image_url'),
                    "thumbnail_path": "",
                    "thumbnail_url": batik_tema.get('image_url')
                }
                # Simpan ke list
                draft_list.append(data)

        return jsonify({
            "status": "success",
            "message": "Daftar draft berhasil di ambil. Berhasil diambil",
            "data": draft_list
        }), 200
    except Exception as e:
        return jsonify({"status": "error", "message": str(e)}), 400

```

Endpoint ini melayani pembuatan daftar portofolio draf membatik pribadi pengguna pada menu galeri karya saya. Server melakukan kueri pencarian

array dokumen ke koleksi database studio_drafts yang dicocokkan berdasarkan kepemilikan data user_id terenkripsi JWT. Melalui proses perulangan, sistem melakukan jembatan relasional data dengan memanggil batik_model.get_by_id() untuk menjemput informasi pelengkap berupa nama motif asli dan gambar visual utama dari tabel batik induk, merapikan format waktu perubahan terakhir (updated_at), lalu mengirimkannya kembali sebagai respons HTTP 200.

7. Video Service

Layanan ini bertanggung jawab untuk mengelola dan mendistribusikan konten edukasi berbasis multimedia (video tutorial atau dokumentasi budaya) mengenai Batik Tegal. Modul ini mendukung pencarian konten serta mengimplementasikan pembagian halaman (*pagination*) agar proses pemuatan data video pada aplikasi mobile tetap ringan dan hemat kuota data.

a. Endpoint: get_all_videos()

```

1 def get_all_videos():
2     try:
3         page = request.args.get('page', 1, type=int)
4         search_query = request.args.get('q', '')
5         per_page = 10
6
7         all_data = video_model.get_all(search_query)
8         total_items = len(all_data)
9         total_pages = math.ceil(total_items / per_page)
10
11         start = (page - 1) * per_page
12         end = start + per_page
13         data_list = all_data[start:end]
14
15         formatted_videos = []
16         for v in data_list:
17             v['id'] = str(v['id'])
18             if 'created_at' in v and v['created_at']:
19                 v['created_at'] = v['created_at'].strftime('%d-%m-%Y %H:%M:%S')
20             formatted_videos.append(v)
21
22         return jsonify({
23             'status': 'success',
24             'message': 'Data video berhasil diambil',
25             'data': formatted_videos,
26             'page': 1,
27             'current_page': page,
28             'total_page': total_pages,
29             'total_items': total_items
30         })
31     except Exception as e:
32         return jsonify({'status': 'error', 'message': str(e), 'code': 500})

```

Sistem membaca parameter eksternal dari URL berupa halaman aktif page (default halaman 1) dan kata kunci pencarian q. Data mentah koleksi video

ayanan ini berfungsi untuk menampilkan agenda kegiatan, pameran, atau karya Batik Tegal ke aplikasi mobile.

- a. Endpoint: `get_all_events()`

a. Endpoint: `get_all_events()`

000

E - 25

Sistem mengambil seluruh data agenda dari MongoDB, lalu memotongnya maksimal 10 data per halaman (`per_page = 10`) agar aplikasi Flutter tetap ringan. Untuk mencegah *Error 500*, objek waktu Python pada field `event_date` dikonversi menjadi string ISO standar via `isoformat()` dan field `created_at` diubah menjadi format teks tanggal standar agar bisa dikirim via JSON dengan aman.

b. Endpoint: `get_event_detail(event_id)`

```
@event_api.route("/events/<string:event_id>", methods=["GET"])
def get_event_detail(event_id):
    try:
        event = event_model.get_by_id(event_id)
    except:
        return jsonify({"status": "error", "message": "Event tidak ditemukan"}), 404

    event["_id"] = str(event["_id"])
    event["banner_image_url"] = (
        f"{request.host_url}static/img/events/{event.get('banner_image_url', 'default_event.png')}"
    )

    if "created_at" in event and event["created_at"]:
        event["created_at"] = event["created_at"].strftime("%Y-%m-%d %H:%M:%S")

    return jsonify({"status": "success", "data": event}), 200
except Exception as e:
    return jsonify({"status": "error", "message": str(e)}), 500
```

Fungsi ini berjalan di layer backend untuk menyajikan rincian deskripsi, tanggal pelaksanaan, dan lokasi lengkap dari satu agenda pameran pilihan pengguna. Sistem menerima parameter kunci `event_id` dari URL dan langsung melempar kueri pencarian tunggal ke database MongoDB via fungsi model `event_model.get_by_id()`. Jika rekam jejak agenda tidak ditemukan di database, server seketika membatalkan proses dan memuntahkan respons galat dengan kode HTTP 404 Not Found. Sebaliknya, jika data tersedia, sistem menyelaraskan parameter tipe data ID dokumen menjadi bentuk string, merangkai URL statis lengkap untuk spanduk pameran, serta memformat string tanggal pembuatan artikel sebelum dikirimkan sebagai objek JSON sukses dengan kode status HTTP 200.

Lampiran 6. Sertifikasi HKI



REPUBLIK INDONESIA
KEMENTERIAN HUKUM

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	EC002026099690, 29 Juni 2026
Pencipta	
Nama	Muchamad Rizqi Pratama, Muhammad Fikri Hidayattullah, S.T., M.Kom. dkk
Alamat	Jl. Perkitut RT 03/RW01, Desa Slawi Kulon, Slawi, Kab. Tegal, Jawa Tengah, 52419
Kewarganegaraan	Indonesia
Pemegang Hak Cipta	
Nama	Unit Penelitian Abdimas Universitas Harkat Negeri
Alamat	Jalan Mataram No. 9, Pesutungan Lor, Kecamatan Margadana 52142, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	Indonesia
Jenis Ciptaan	Program Komputer
Judul Ciptaan	Sibatikgal: Sistem Informasi Mobile Berbasis Kecerdasan Buatan untuk Identifikasi Motif Batik Tegalan
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	28 Juni 2026, di Kota Tegal
Jangka waktu perlindungan	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman
Nomor Pencatatan	001307540

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri



Agung Dumasasongko.SH.,MH.
NIP. 196912261994031001



Disclaimer:

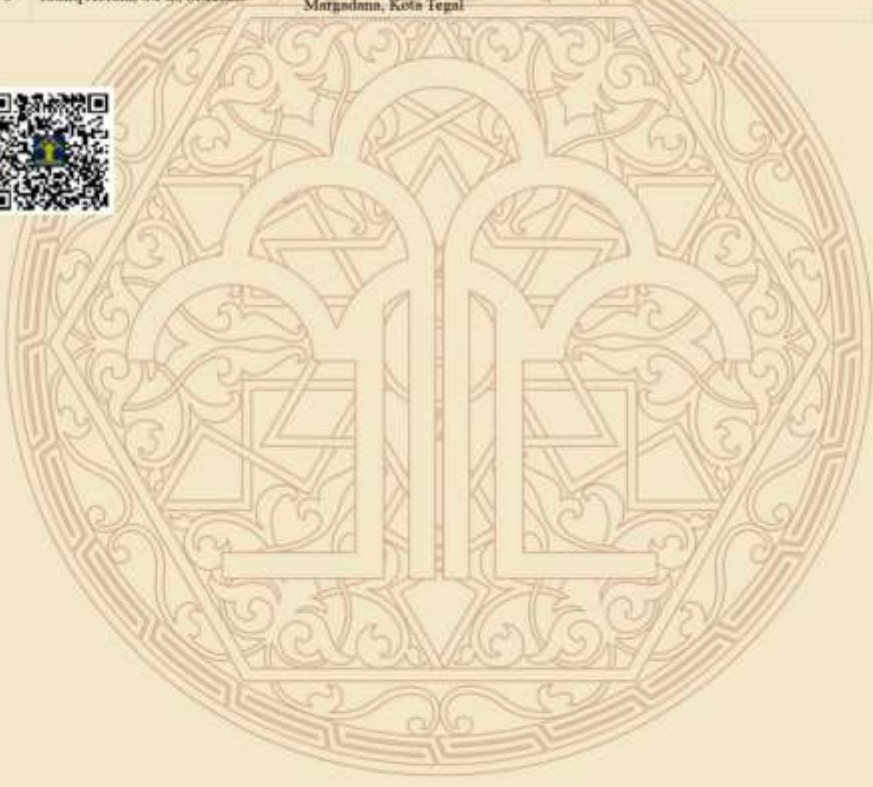
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah diunggah secara elektronik menggunakan segel elektronik yang diterbitkan oleh Badan Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keabsahannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.



LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Mochammad Rizqi Pratama	Jl. Perkatut RT 03/RW01, Desa Slawi Kulon Slawi, Kab. Tegal
2	Muhammad Fikri Hidayatullah, S.T., M.Kom.	Jalan Glatik No. 68, Kelurahan Randugunting Tegal Selatan, Kota Tegal
3	Taufiq Abidin, S.Pd., M.Kom.	Jl. Kaligangsa Asri Timur VIII no 33 RT 3 RW 7 Kel. Kaligangsa Margadana, Kota Tegal







Balai Besar
Sertifikasi
Elektronik

Disclaimer:

1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat penciptaan permohonan;
2. Surat Penciptaan ini telah diunggah secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Balai Seder dan Seder Negeri;
3. Surat Penciptaan ini dapat dibuktikan keabsahannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser;

Lampiran 7. Lembar Bimbingan Skripsi

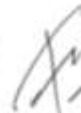


PROGRAM STUDI SARJANA TERAPAN TEKNIK
INFORMATIKA
FAKULTAS SEKOLAH VOKASI
UNIVERSITAS HARKAT NEGERI

LEMBAR BIMBINGAN SKRIPSI

Nama : Muchamad Rizqi Pratama
NIM : 22090104
No. Ponsel : 081391497365
Judul Skripsi : PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI
MOTIF BATIK TEGALAN BERBASIS COMPUTER VISION
UNTUK Mendukung PELESTARIAN BUDAYA LOKAL
Dosen Pembimbing 1 : Muhammad Fikri Hidayattullah, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	15/4 2026	Dasar Set	- Desain utaman diambil dari bun Nur Anisa Anini - Laksanakan negosiasi	
2.	6/5	Desain	Selalu digar dgn dan Anisa!	
3.	13/5	Komponen		
4.	2/6	Detail	- Canvas batik - Validasi dan Bulan tidak a. f. 6. 1!	

5	2/6	Hbl	Agnes	
6	2/6	Leipen	1. Schreiben des Kaiden ilenial 2. Absorption	
7	1/7	Leipen	Schreiben des Kaiden ilenial	
8	2/7	Leipen	Ace	

--	--	--	--	--

Tegal, 2 Juli 2022
Dosen Pembimbing I


Muhammad Fikri Hidayattullah, S.T., M.Kom.
NIPY. 09.016.307



PROGRAM STUDI SARJANA TERAPAN TEKNIK
INFORMATIKA
FAKULTAS SEKOLAH VOKASI
UNIVERSITAS HARKAT NEGERI

LEMBAR BIMBINGAN SKRIPSI

Nama : Muchamad Rizqi Pratama
NIM : 22090104
No. Ponsel : 081391497365
Judul Skripsi : PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI
MOTIF BATIK TEGALAN BERBASIS COMPUTER VISION
UNTUK MENDUKUNG PELESTARIAN BUDAYA LOKAL
Dosen Pembimbing II : Taufiq Abidin, S.Pd., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	15 April 2026	Konsep	Lengkapi dataset	
2	30 April 2026	ERD	Sistem mampu mendeteksi motif yang bukan batik Tegal, siapkan database sesuai ERD	
3	13 Mei 2026	produk	diselesaikan sekitar 80%, sembari meningkatkan optimasi	
4	4 Juni 2026	Produk	dihosting, lanjut ke laporan	
5	18 Juni 2026	Laporan Bab I	perbaiki margin sesuai panduan	
6	23 Juni 2026	Laporan	Bab I, II Oke	
7	25 Juni 2026	Laporan	Bab III Oke	
8	30 Juni 2026	Laporan	Oke siap ujian	

Tegal, 30 Juni 2026
Dosen Pembimbing II



Taufiq Abidin, S.Pd., M.Kom.
NIPY. 06.014.184